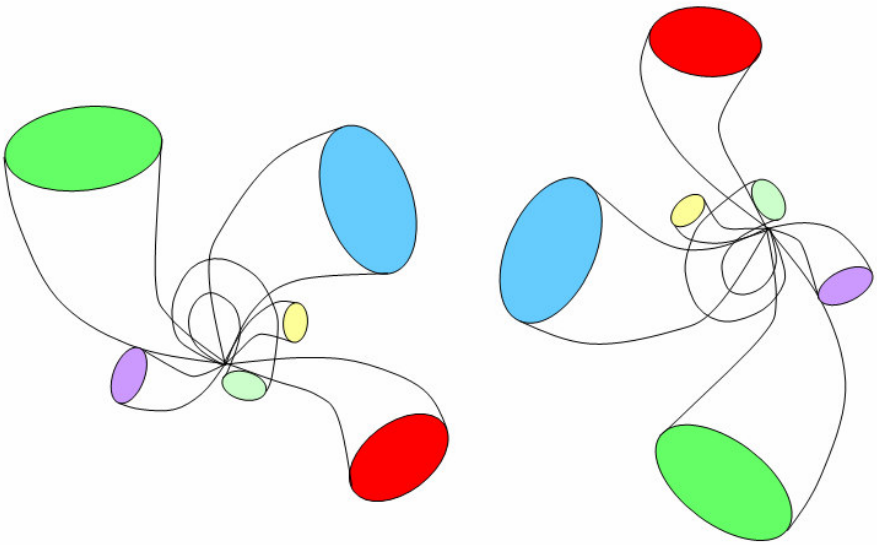
部分空間法研究会 Subspace2006

MIRU2006 サテライト研究会



主催　部分空間法研究会 実行委員会

共催　MIRU2006

序文

本研究会は，部分空間法の歴史的意義を確認するとともに，今後の理論的，実用的な可能性を探るための議論の場を提供することを目標として企画されました．部分空間法は，渡辺，飯島により提案されてから約 40 年が経っており，当初，このような研究会を開催して果たしてどの程度の参加や発表があるのだろか危惧していました．しかし最終的には会場変更やプログラム編成に頭を抱える程の申込を頂きました．委員一同，この大きな反響に，普遍な基盤理論に対する皆様の強い思いを感じずにはいられません．

部分空間法はアルゴリズムの簡潔性と同時に大きなポテンシャルを持つ方法です．言うまでもなく，部分空間法の基礎である部分空間の概念は，科学全般における共通的な概念であります．したがって，部分空間法の背後に，多変量解析，統計学などの様々な分野の拡がりを鮮明に見ている方も多いのではないでしょうか．もちろん，部分空間法はパターン認識のための極めて重要な理論であるばかりではなく，この 30 年以上，文字認識や顔画像認識など様々な現場でのテストに耐えてきた実用的な技術でもあります．さらに，部分空間法は，日本発の技術であるという意味で国内での議論が重要な技術です．

この主旨のもと，論文そのものの新規性は必ずしも問わず，新しい可能性を探るための論文を幅広く募集しました．その結果，新規提案も含めて，部分空間法の有効性を再認識する研究や，今後の理論拡張に可能性を見出せる研究など，充実した研究論文が揃いました．さらに，本研究会では，これらの一般講演に加えて，部分空間法の有効性やそのポテンシャルをより理解してもらう目的で，名古屋大の村瀬洋教授による特別講演，および東芝の黒沢由明氏による「部分空間法入門」と題したチュートリアルも企画しました．これらの講演を通して，部分空間法の奥深さを皆様に改めて感じて頂ければ，委員一同の喜びです．

本研究会は MIRU として初の試みである MIRU2006 サテライトワークショップとして開催されます．最後になりましたが，サテライトとして開催することをご承認頂いた村瀬、谷口両委員長をはじめとした実行委員各位，協賛学会各位，また本研究会の開催に当たり，会場をご提供頂いた東北文化学園大学をはじめとして，様々な準備にご協力頂いた多くの方々に深く感謝いたします．本研究会が，部分空間法の有効性と可能性を再認識する機会を提供すると共に，参加者皆様の研究の進展の一助になることを祈念しております．

部分空間法研究会 Subspace2006　実行委員長　福井和広

実行委員　内田誠一, 大町真一郎, 坂野 鋭

佐藤 敦，佐藤真一，佐藤洋一, 牧 淳人

顧問　前田賢一

プログラム

7 月 18 日(火曜日)

09:50 開会の挨拶 A 会場

福井 和広 委員長

10:00-11:40 A 会場: 応用

1-1 市野 将嗣(早大), 坂野 鋭(NTT データ), 小松 尚久(早大), 話者認識における核非線形相互部分空間法の適用に関する検討

1-2 玉木 徹(広大), 天野 敏之(名工大), マルチポート固有空間法

1-3 Naoya OHNISHI, Atsushi IMIYA(IMIT, Chiba Univ.), Independent Component Analysis for Obstacle Detection

1-4 榎田 修一 江島 俊朗(九工大), 部分空間法に基づく階層的自己位置推定

1-5 Tsuyoshi Kato, Kiyoshi Asai, Paul B. Horton, Koji Tsuda, Wataru Fujibuchi(AIST), Network-based Noise Reduction for Microarray Data

10:00-11:40 B 会場: 基礎

2-1 Seiichi Uchida(Kyushu Univ.), Subspace of deformations

2-2 岩村 雅一(阪府大), 大町 真一郎, 阿曽 弘具(東北大), 部分空間法における原点の位置と認識性能に関する考察

2-3 平山 淳一, 中山 英久, 加藤 寧(東北大), 差分部分空間を用いた混合マハラノビス関数に関する一考察

2-4 河原 智一, 西山 正志, 山口 修(東芝), 多重直交相互部分空間法を用いた顔認識

2-5 福井 和広(筑波大), 山口 修(東芝), 制約相互部分空間法と直交相互部分空間法の比較 ー 各クラス部分空間の直交化の観点から ー

11:40-11:50 休憩

11:50-12:50 A 会場: チュートリアル

黒沢 由明(東芝ソリューション), Subspace2006 部分空間法入門

12:50-14:20 昼休み A 会場 ポスターセッション

14:20-15:00 A 会場: 特別講演

村瀬 洋(名大), 体験談:部分空間法による画像認識

15:00-15:10 休憩

15:10-16:10 A会場: Novel Mathematical Feature Extraction

3-1 Jun Fujiki, Shotaro Akaho(AIST), Approximation of Spherical PCA by Euclideanization

3-2 Masashi Sugiyama(Tokyo Institute of Technology), Local Fisher Discriminant Analysis

3-3 Kohei Inoue, Kenji Hara, Kiichi Urahama(Kyushu Univ.), Equivalence of Non-Iterative Algorithms for Simultaneous Low Rank Approximations of Matrices

16:10-16:20 休憩

16:20-17:40 A会場: Nonlinear Subspace Method

4-1 Atsushi Imiya, Yoshihiko Yamagishi, Tomoya Sakai(IMIT, Chiba Univ.), Principal Curve Analysis on Manifold

4-2 堀田 政二, 喜安 千弥, 宮原 末治(長崎大), 局所部分空間法と変形不変性を用いた画像パターン分類

4-3 Ryo Saegusa, Shuji Hashimoto(Waseda Univ.), A Nonlinear Subspace Method for Pattern Recognition using Multi-Layered Perceptrons

4-4 Yoshikazu Washizawa(RIKEN), Yukihiko Yamashita(Tokyo Institute of Technology), A Family of Kernel Subspace Classifiers

17:40-18:30 A会場: ポスターセッション

18:30- A会場: フリーディスカッション 座長 前田 賢一(東芝)

目次

序文 福井 和広 委員長
プログラム

・市野 将嗣(早大), 坂野 鋭(NTT データ), 小松 尚久(早大), 話者認識における核非線形相互部分空間法の適用に関する検討 ---------- 1
・玉木 徹(広大), 天野 敏之(名工大), マルチポート固有空間法 ---------- 7
・Naoya OHNISHI, Atsushi IMIYA(IMIT, Chiba Univ.), Independent Component Analysis for Obstacle Detection ---------- 16
・榎田 修一, 江島 俊朗(九工大), 部分空間法に基づく階層的自己位置推定 ---------- 22
・Tsuyoshi Kato, Kiyoshi Asai, Paul B. Horton, Koji Tsuda, Wataru Fujibuchi(AIST), Network-based Noise Reduction for Microarray Data ---------- 28

・Seiichi Uchida(Kyushu Univ.), Subspace of deformations ---------- 34
・岩村 雅一(阪府大), 大町 真一郎, 阿曽 弘具(東北大), 部分空間法における原点の位置と認識性能に関する考察 ---------- 42
・平山 淳一, 中山 英久, 加藤 寧(東北大), 差分部分空間を用いた混合マハラノビス関数に関する一考察 ---------- 52
・河原 智一, 西山 正志, 山口 修(東芝), 多重直交相互部分空間法を用いた顔認識 ---- 57
・福井 和広(筑波大), 山口 修(東芝), 制約相互部分空間法と直交相互部分空間法の比較
－ 各クラス部分空間の直交化の観点から － ---------- 63

・Jun Fujiki, Shotaro Akaho(AIST), Approximation of Spherical PCA by Euclideanization ---------- 72
・Masashi Sugiyama (Tokyo Institute of Technology), Local Fisher Discriminant Analysis ---------- 85
・Kohei Inoue, Kenji Hara, Kiichi Urahama(Kyushu Univ.), Equivalence of Non-Iterative Algorithms for Simultaneous Low Rank Approximations of Matrices ---------- 101

・Atsushi Imiya, Yoshihiko Yamagishi, Tomoya Sakai, (IMIT, Chiba Univ.), Principal Curve Analysis on Manifold ---------- 107
・堀田 政二, 喜安 千弥, 宮原 末治(長崎大), 局所部分空間法と変形不変性を用いた画像パターン分類 ---------- 111

・Ryo Saegusa, Shuji Hashimoto, (Waseda Univ.), A Nonlinear Subspace Method for Pattern Recognition using Multi-Layered Perceptrons ---------- 119
・Yoshikazu Washizawa(RIKEN), Yukihiko Yamashita (Tokyo Institute of Technology), A Family of Kernel Subspace Classifiers ---------- 125

・黒沢 由明(東芝ソリューション), Subspace2006 部分空間法入門 ---------- 136
・村瀬 洋(名大), 体験談:部分空間法による画像認識 ---------- 144

付録
黒沢 由明(東芝ソリューション), Subspace2006 部分空間法入門 講演スライド ---------- 150

話者認識における核非線形相互部分空間法の適用に関する検討

市野　将嗣† 　坂野　鋭†† 　小松　尚久†

† 早稲田大学理工学部　〒 169–8555 東京都新宿区大久保 3–4–1
†† （株）NTT データ　〒 104–0033 東京都中央区新川 1–21–2 茅場町タワービル 7F
E-mail: †{ichino,komatsu}@kom.comm.waseda.ac.jp, ††sakanoh@nttdata.co.jp

あらまし　本稿では核非線形相互部分空間法を用いた音声に基づく話者認識のアルゴリズムを提案し, 実験的に有効性を示す. 従来より音声による話者認識において，混合ガウス分布モデルが用いられている．しかしながら，音声データの分布によるアルゴリズムの妥当性の検証について十分には行われていない．分布に湾曲構造のような非線形構造が認められる場合には，カーネルマシンに代表される非線形アルゴリズムを用いることが有効である可能性がある．そこで我々は，扱う対象の分布について調べ，非線形性を示すことを実験的に確認した．それを踏まえ，非線形アルゴリズムのひとつである，核非線形相互部分空間法を適用することで，従来より用いられている混合ガウス分布モデルに比較して高い識別率が得られることを示す.

キーワード　音声，テキスト提示型話者認識，LPC ケプストラム，核非線形相互部分空間法

A study on using the Kernel Mutual Subspace Method in speaker recognition

Masatsugu ICHINO†, Hitoshi SAKANO††, and Naohisa KOMATSU†

† Department of Science and Engineering,Waseda University　Okubo 3–4–1, Shinjuku-ku, Tokyo,169–8555 Japan
†† NTT Data Corporation　Kayabacho Tower, 1–21–2 Shinkawa,Chuo-ku, Tokyo,104–0033 Japan
E-mail: †{ichino,komatsu}@kom.comm.waseda.ac.jp, ††sakanoh@nttdata.co.jp

Abstract　We propose a method of speaker recognition based on voice by using the kernel mutual subspace method. The Gaussian Mixture Model has already been developed as an algorithm for person authentication using voice. However, it is not sufficiently discussed the verification of the algorithm using the distribution of voice data. When the distributions are nonlinear properties like curved structures, it is possible that using the nonlinear algorithms like the kernel machines is effectiveness. So, we have investigated the distributions of voice data in feature space and experimentally obtained their nonlinear properties. We experimentally demonstrate the proposed method's effectiveness with simulation results and show that the method achieved higher accuracy than that of using the Gaussian Mixture Model.

Key words　voice, text prompted speaker recognition, LPC cepstrum, kernel mutual subspace method

1. ま え が き

本稿では核非線形相互部分空間法を用いた音声に基づく話者認識のアルゴリズムを提案し, 実験的に有効性を示す.

音声による個人認証はマイクロフォンなどで実装できるため，携帯電話や PC に備え付けのマイクなど，身近なところで利用できる．また，発話は人間が普段から行っている自然な動作であるため，指紋などによる個人認証に比べ，ユーザの心理的な負担が少ない．そのため，音声による個人認証は高レベルのセキュリティを必要としない場面での簡便な個人認証方式として注目されている.

音声による個人認証のためには特定のパスワードを用いることを前提としたテキスト固定型，何を話してもよいテキスト自由型，認証システムが話す言葉を指定するテキスト提示型の 3 つの方法が提案されている.

本研究では，個人認証方式としてテキスト提示型を採用する．認証を行う際，本人の特徴の現れやすいテキストを提示することによって認証精度の向上が期待できる．また，事前に録音を

しておくことが困難であるため，なりすましの問題が軽減される．それに対してテキスト固定型，テキスト自由型は録音装置から出力された音声を用いるなりすましに弱いことが指摘されている．

従来，音声による個人認証の研究は記憶した音声情報との照合の問題として捉えられ，音素の類似性を比較するアプローチが行われてきた．

その中でも，隠れマルコフモデル（Hidden Markov Model, 以下 HMM）や動的計画法 (dynamic programing，以下 DP) による認識が試みられてきた [1] [2]．これらは特定の情報を学習・認識するものであり，テキスト固定型の音声認証アルゴリズムとの親和性が高い．しかし，HMM や DP などに基づく音声認識手法では，時系列情報を利用するため，明らかにテキスト固定型以外への応用が困難である．

また，同様のアプローチとして，混合ガウス分布モデル (Gaussian Mixture Model, 以下 GMM) が用いられている [3]．GMM は時系列情報を利用しないためテキスト自由型，テキスト提示型の音声認証アルゴリズムとも共用することが可能である．

GMM は話者認識のアルゴリズムとして広く用いられているにもかかわらず，音声データの分布によるアルゴリズムの妥当性の検証は十分には行われていない．音声データの分布の様子を考慮して話者認識のアルゴリズムを選択することにより，さらに高性能な認識系が構成できる可能性がある．

著者らは既に音声において非線形性が存在することを確認している [4]．音声データの分布に非線形構造が認められる場合，線形性を仮定したアルゴリズムでは分布を十分正確に近似することが出来ず，認識精度の低下を引き起こすことになる．

また，音声データの分布の非線形性を考慮した話者認識の研究として，核非線形部分空間法 [5] [6] を適用した例がある [7]．そこでは，核非線形部分空間法を適用した際の識別性能は，GMM を用いた際の識別性能に匹敵することが示されている．

音声による個人認証は，連続的な音声入力を仮定している．しかし，GMM あるいは核非線形部分空間法では，これを単一の音声認識問題の連続したものとして扱っている．

本研究では，音素の連続する軌跡の形状を比較するアプローチにより連続的な入力音声を積極的に利用し，さらに高性能な認識系を構成することを目指す．

こうした話者認識装置において重要なのは音素の連続する軌跡がどのような形状を取るかである．

本研究では，特徴抽出系として LPC ケプストラムを用いる．LPC ケプストラムは本来，音声認識のために開発された特徴抽出系であり，個人性よりも音声信号の特徴を残している可能性がある．各個人の同一音声が個人の別の音素より離れて分布している可能性がある．つまり，各個人の音素の連続する軌跡は非線形に絡み合って存在する可能性がある．軌跡の形状を正確にモデル化できない場合には，他のカテゴリに誤認識してしまう．

以上を踏まえて，本稿においては，非線形分布が存在し，連続的な入力が仮定できる場合に強力な識別アルゴリズムとして知られている核非線形相互部分空間法を用いることを提案する．以下，**2.** では，非線形, 連続分布条件下での話者認識アルゴリズムを提案する．また，**3.** では XM2VTS データベースを用いた実験により提案手法の有効性を示す．さらに，**4.** ではまとめと今後の課題である．

2. 核非線形相互部分空間法

2.1 相互部分空間法

相互部分空間法（Mutual Subspace Method, 以下 MSM) [8] [9] は，認識対象の入力として複数データが利用できる場合に適用され，入力ベクトル集合も主成分分析を用いて部分空間で表現し，テンプレートの部分空間との間の角度を類似度として識別を行う．この角度に基づいた識別は正準角の概念を用いる．

2 つの部分空間 V, W のなす正準角 θ の余弦は，次のように計算される [8]．

テンプレートの部分空間を V，入力された時系列データに対する部分空間を W とする．V の部分空間の次元を M，W の部分空間の次元を N とし，$\phi_m (m = 1, \cdots, M), \psi_i (i = 1, \cdots, N)$ を各部分空間 V, W における正規直交基底ベクトルとする．次いで，式 (1) で表される行列 X の固有値問題を解き，その最大固有値を第 1 正準角に対応する類似度 S_{mutual} とする．また，$N \leqq M (1 \leqq i, j \leqq N)$ とする．

ここで，

$$X = (x_{ij}) \tag{1}$$

$$x_{ij} = \sum_{m=1}^{M} (\psi_i \cdot \phi_m)(\phi_m \cdot \psi_j) \tag{2}$$

とおくと，

$$XU = \Lambda U \tag{3}$$

となる．ここで U は X の固有ベクトル，λ_{max} は固有値 Λ の最大値である．故に，第 1 正準角に対応する類似度 S_{mutual} は，

$$S_{mutual}(V, W) = \lambda_{max} = \cos^2 \theta \tag{4}$$

となる．

さらに，式 (1) の固有値問題の第 j 固有値を第 j 正準角に対応する類似度として扱うことができる [10]．

MSM では，学習データと入力データの変動の少ない統計量同士を比較することになり，扱う対象に非線形性がない場合には強力な物体認識手法になる．

2.2 核非線形相互部分空間法

前節で導入した MSM は，扱う対象に非線形性がある場合には，十分な精度を達成できないという問題がある．このような非線形性の問題を解決するために，坂野らによって相互部分空間法と核非線形主成分分析 (Kernel Principal Component Analysis，以下 KPCA) [11] を融合した核非線形相互部分空間法 [12](Kernel Mutual Subspace Method, 以下 KMS) が提案されている．

Schölkopf により提案された強力な非線形 PCA である KPCA

では，非線形な関数表現を考えるために関数空間(注1)への非線形写像

$$\Psi : \mathcal{R}^n \to \mathcal{F}, \vec{x} \to \vec{X} \tag{5}$$

を考える．ただし，$\mathcal{F}$ は極めて高次元もしくは無限次元の関数空間である．

次の $m \times m$ 行列

$$K_{ij} = (\Psi(\vec{x}_i) \cdot \Psi(\vec{x}_j)) \tag{6}$$

を定義し，

$$m\lambda\alpha = \alpha K \tag{7}$$

なる固有値問題を解き，特異値分解の公式に従い，

$$V = \frac{1}{\lambda}\alpha\Psi(\vec{x}) \tag{8}$$

の形で基底ベクトルを計算する．$\Psi(\cdot)$ の選択のためには，Mercer の条件

$$k(\vec{x}, \vec{y}) = (\Psi(\vec{x}) \cdot \Psi(\vec{y})) \tag{9}$$

を満たすような写像を選ぶ．このような写像が選択できた場合には，$\Psi(\vec{x}) \cdot \Psi(\vec{y})$ は単に関数 $k(\vec{x}, \vec{y})$ を計算することに帰着される．これより，

$$V \cdot \Psi(\vec{x}) = \frac{1}{\lambda}\sum_{i=1}^{m} \alpha_i k(\vec{x}_i, \vec{x}) \tag{10}$$

のようになり，高次元の固有ベクトル V をあらわに求めないでも，その写像を計算できるようになる．

KMS では，辞書側と入力側の部分空間の角度を類似度として扱う．ここで，V を辞書側部分空間の基底ベクトル，W を入力側部分空間の基底ベクトルとする．V，W はそれぞれ辞書側，入力側データから KPCA によって計算された部分空間の基底ベクトルであり，ノルムが正規化されていると仮定すると，辞書登録された m 個の音声群と m' 個の入力された音声系列の類似度は，$(V \cdot W)$ の大きさで評価される．$(V \cdot W)$ は $\mathcal{F}$ 上の内積であるから，この値をあらわに有限の時間で計算することができない．しかし，V, W を

$$V = \sum_{i=1}^{m} \alpha_i \Psi(\vec{x}_i) \tag{11}$$

$$W = \sum_{j=1}^{m'} \alpha'_j \Psi(\vec{x}'_j) \tag{12}$$

と表現することにより，$(V \cdot W)$ の表式は

$$V \cdot W = \sum_{i=1}^{m} \alpha_i \Psi(\vec{x}_i) \cdot \sum_{j=1}^{m'} \alpha'_j \Psi(\vec{x}'_j) \tag{13}$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{m'} \alpha_i \alpha'_j (\Psi(\vec{x}_i) \cdot \Psi(\vec{x}'_j)) \tag{14}$$

(注1)：関数空間のベクトルについても有限次元のベクトル空間の記号・用語(「行列」，転置の記号) を用いることとする．

となる．

ここで，式 (14) に式 (9) を代入すると，

$$V \cdot W = \sum_{i=1}^{m}\sum_{j=1}^{m'} \alpha_i \alpha'_j k(\vec{x}_i, \vec{x}'_j) \tag{15}$$

となり，有限の時間で類似性が評価できることがわかる．実際に類似度を評価するには式 (1)，(2)，(3) に式 (15) を代入すればよい．

3. 認 識 実 験

本節では, 音声による話者認識に KMS を用いた際の実験結果を示す．最初に実験系の概要を示し, 音声データの分布の様子を調べ, 認識実験結果を報告する．

3.1 実験系の概要

3.1.1 特 徴 抽 出

本研究で用いる話者認識装置では音声のうち有音のみを対象とした．発話動作による連続的な音声情報を解析するためである．

KMS は，データを部分空間で表現する際，サンプル数の 3 乗に比例する処理量を有する．また，認識時の類似度を計算する際には，すべての学習データ，認識対象データの間での核関数を計算することになり，処理量が大きい．本研究では，計算時間の発散を防ぐために, 学習データのサンプル数を削減することにより処理量削減を行った．具体的には，学習データに対して k-平均法を行った際に求められるクラスタ中心を学習データとして用いた [13].

また，特徴抽出系として LPC ケプストラムを用いた．LPC ケプストラムは話者認識のための特徴抽出系として最も多く用いられていることから話者認識装置で採用することとした．

この特徴抽出系と KMS を用いた個人認証方式の概要を図 1 に示す．

3.1.2 実験データ

実験データとして，大規模データベース XM2VTS [14] を使用した．XM2VTS は 295 人の被験者を一ヶ月間隔で 4 回撮影した (1 回ごとに 1 時期とし，合計 4 時期あるとする) データベースである．発話した際の動画像 (顔画像と音声) が収録されている．

今回の実験データとして，40 人分 (男性 26 人，女性 14 人，4 時期分) を用いた．本研究では，あらかじめ「0,1,2,3,4,5,6,7,8,9」(英語) の 10 個の数字を発話した際の音声のデータをシステムに登録し，認証時に毎回数字の並びを変化させた数字列を提示し，その数字を連続で読み上げることにより認証するというテキスト提示型話者認識を想定している．そのため，学習データは「0,1,2,3,4,5,6,7,8,9」(英語) の 10 個の数字を連続で読み上げたデータを 1 つのデータとし，テストデータは「5,0,6,9,2,8,1,3,7,4」(英語) の 10 個の数字を連続で読み上げたデータを 1 つのデータとして評価した．そして，4 時期中の 2 時期分を学習データ，テストデータとして用いた．また，残りの 2 時期分の学習データとテストデータに関しても実験を行った．ただし，学習データとテストデータの選択に関して，同一

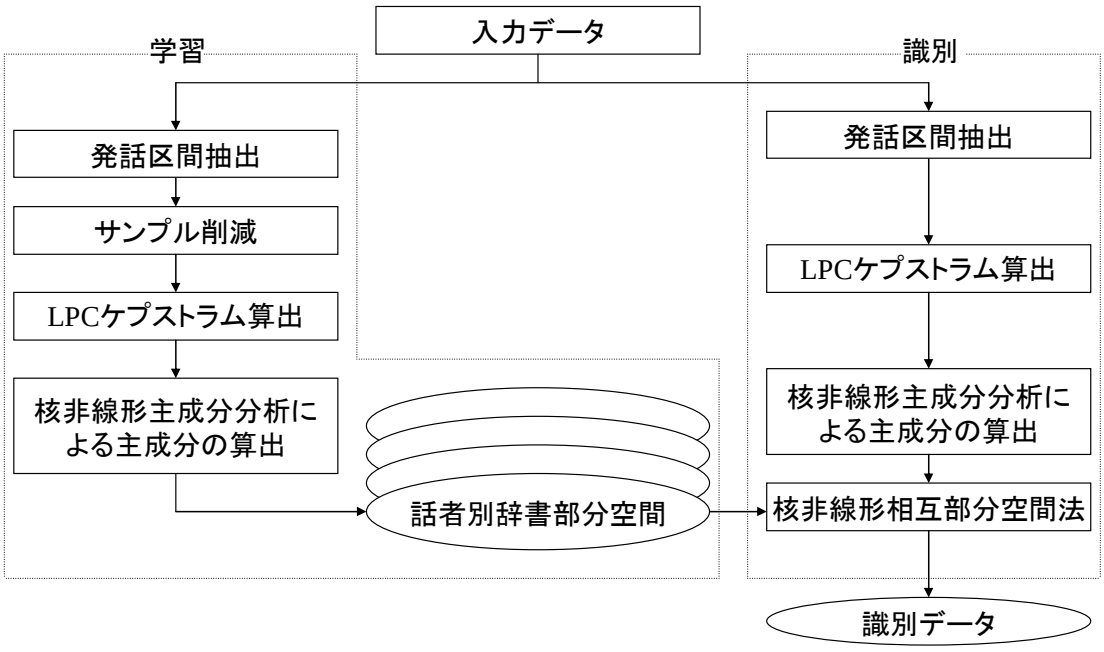


図 1　実験系の概要

時期のデータを選んでいない．2 回実験を行い，2 回分をあわせて実験結果としている．1 時期につき同じテキストを 2 回発話しているので 1 回の実験につき学習データは 1 人あたり 4 データ，テストデータは 1 人あたり 4 データを用いた．結局 2 回の実験でテストデータとして 40×4×2=320 データを用いた．

1 人あたり 4 データ (あわせて 16 秒前後) を用いて辞書部分空間を表現し，1 人あたり 1 データずつ (4 秒前後) を用いて入力部分空間を表現した．

発話区間抽出では，実験で使用するフレームを選択する．実験で使用するフレームとして，音声波形の有音に対応するフレームのみを手動で切り出した結果を用いた．その後，線形予測分析を行い LPC ケプストラムを算出し，特徴ベクトルとして用いた．

実験諸元を表 1 に示す．

表 1　実 験 諸 元

サンプリング周波数	32[kHz]
フレーム長	32[ms]
フレーム周期	8[ms]
特徴抽出	LPC ケプストラム
LPC ケプストラム次元数	24

3.2　実 験 結 果

3.2.1　音声データの分布

線形主成分分析により特徴ベクトルを 2 次元に写像することで音声データの分布の様子について調べた．

図 2 は「3」(three) を発話した音声データ 1 人分を線形主成分分析により 2 次元に写像した結果である．散布図の x 軸，y 軸がそれぞれ第 1，第 2 主成分に対応している．テストデータを+印，全テストデータの平均値を黒四角で示した．発話することによる軌跡の概形を実線で示した．

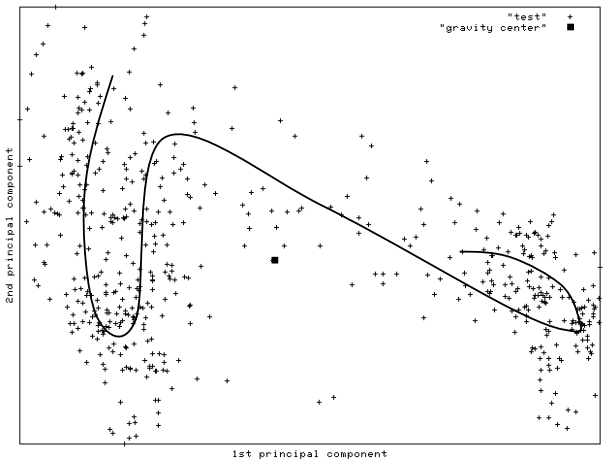


図 2　線形主成分分析による音声データの分布の散布図

この散布図において，音声データは，混合ガウス分布とも湾曲構造とも認識できる分布をしていることがわかる．「3」を発話する際の軌跡に着目すると，一本の湾曲した軸上を移動する

ことがわかる．

さらに，データの重心の周りにはサンプルが少ないことより，この分布は非線形構造を持つと考えられる．

LPC ケプストラムは本来, 音声認識のために開発された特徴抽出系であり, 個人性よりも音声信号の特徴を残している可能性がある．つまり, 各個人の同一音声が個人の別の音素より離れて分布している可能性がある．そのため，他人の同じ数字の分布についても同様の非線形構造を取ると考えられる．

3.2.2 識 別 結 果

音声に関して，KMS を適用することの有効性を確認するために，話者認識において広く用いられている GMM との比較を行った．

KMS については大幅な処理時間の削減と部分空間次元数によらず安定した識別率を得ることを考慮して，学習サンプルを 80%削減した．GMM についてはサンプル削減を行わない．GMM は複数の入力を仮定した方法ではないため，複数回の認識処理の類似度の平均を類似度として用いた．KMS は第 1 正準角から第 10 正準角まで考慮した類似度の平均を類似度とした．

核関数として，ガウス型動径基底関数

$$k(\vec{x},\vec{y}) = \exp\left(\frac{-||\vec{x}-\vec{y}||^2}{2\sigma^2}\right) \tag{16}$$

を用い，予備実験により適切と思われる σ を設定した．

識別結果を表 2 に示す．テストデータ数を考慮すると，KMS が GMM より高い識別性能を示すことがわかる．また，図 3 に照合精度特性 (Receiver Operating Characteristic Curve, 以下 ROC 曲線) を示す．図において横軸が FRR(False Reject Rate)，縦軸が FAR(False Accept Rate) を表す．図 3 より KMS が GMM と比較して高い識別性能を示すことが確認できた．

図 4 に累積識別精度特性 (Cumulative Match Characteristic Curve) を示す．図において，縦軸が累積識別率 (Cumulative Match Rate, 以下 CMR)，横軸が順位を表す．この図より，KMS は GMM と比較して高い CMR を示すことがわかる．また，KMS を用いて誤識別した場合においても識別候補の上位に本人が判定される．つまり，唇動作などの他のモダリティと統合した場合にも KMS は GMM より高精度化できる可能性があると考えられる．

表 2 識 別 結 果

識別手法	KMS	GMM
識別率（%）	83.8	80.0
σ	0.7	-
辞書次元数	13	-
入力次元数	10	-

3.2.3 考　　察

今回の実験において，KMS が GMM より高い識別性能を示した理由について考察する．

KPCA により特徴ベクトルを 2 次元に写像することで音声データの分布の様子について調べた．

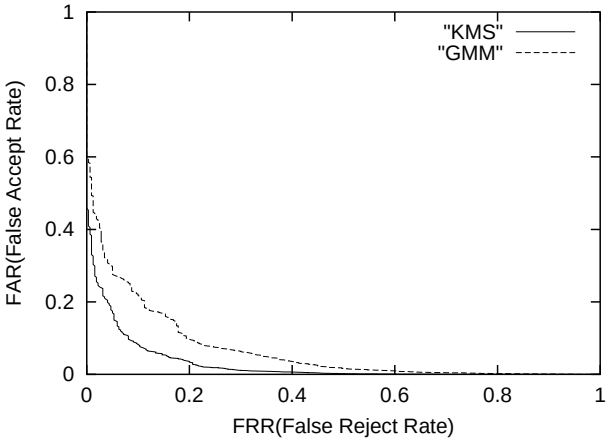


図 3 ROC 曲線

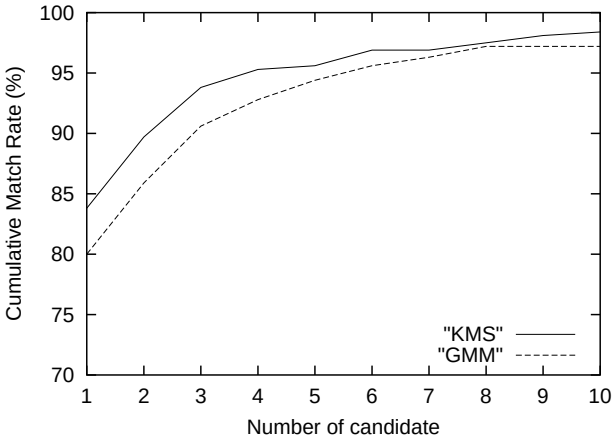


図 4 累積識別精度特性

図 5 は図 2 と同一の音声データ 1 人分 (「3」) を KPCA により 2 次元に写像した結果である．散布図の x 軸，y 軸がそれぞれ第 1，第 2 主成分に対応している．テストデータを+印で示した．核関数として，ガウス型動径基底関数を用い，図は，$\sigma = 0.1$ の結果である．

核関数の式 (16) を見てもわかるように σ が小さいときには関数が鋭くなり，複雑な分布を近似できる．図 5 より，今回は σ が非常に小さい値で線形性を示すことがわかる．つまり音声データは，非常に強い非線形性を示していることがわかる．この結果は，英語の音素は 44 個あるため，音声データとして取りえる状態が多く，複雑な分布になると考えられることとも一致する．

GMM を用いる場合，複雑な分布を表現するためには，混合数を大きくする必要がある．混合数が大きい場合，多くの学習データが必要となる．学習データが十分でない場合は，過学習を起こしてしまい，精度低下につながる．

それに対して，KMS を用いる場合，音素の連続する軌跡の形状を核関数のパラメータを調整して扱うため，多少学習データが少なくても複雑な分布を表現できると考えられる．

そこで，KMS に関してサンプル削減量と識別率の関係について調べた．結果を図 6 に示す．縦軸が識別率，横軸がもとのサンプル数に対するサンプル削減数の割合を表す．90%のサン

プルを削減した場合においてもほとんど識別率が低下していないことがわかる．

多少学習データが少なくても複雑な分布を表現できることは，図 6 および 80%のサンプルを削減し残り 20%のサンプルのみを用いて実験した KMS の識別性能がサンプルを削減しないで実験した GMM の識別性能より高いことからわかる．

以上より，今回の実験においては，KMS が GMM より高い識別性能を示したと考えられる．

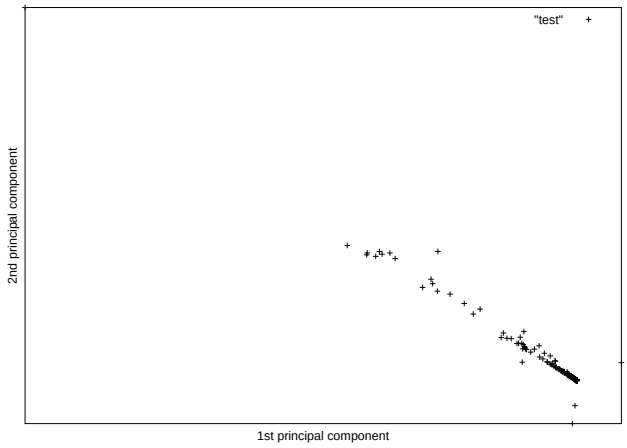


図 5　核非線形主成分分析による音声データの分布の散布図

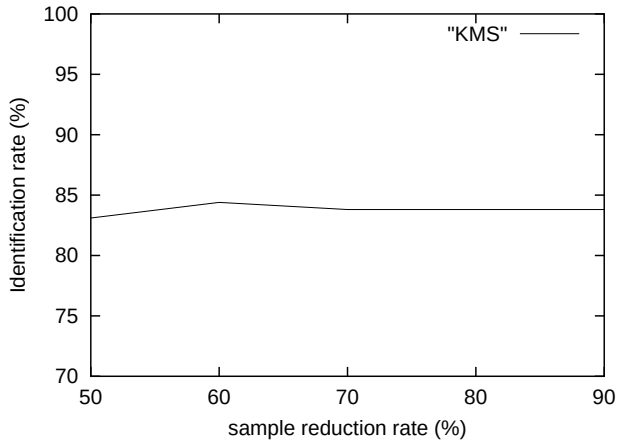


図 6　サンプル削減量と識別率の関係

4. むすび

テキスト提示型話者認識に適用可能な音声による話者識別のアルゴリズムを提案した．

発話時における音声が，音声空間において非線形性を示すことを実験的に確認した．この結果を踏まえて音声による個人認証に KMS を適用した．識別実験の結果より，提案手法が有効である可能性を示した．

さらに，より大規模なデータで再検証を行い，KMS を用いることの有効性を確認していく予定である．

我々は，唇動作に関しても非線形性の存在を実験的に確認している．それを踏まえて，唇動作による話者認識のアルゴリズムとして KMS を適用することの有効性を確認している [15]．今後は，音声，唇動作の 2 つのモダリティを用いた多重バイオメトリックシステムの構築および精度向上に関する検討を進めていく予定である．MIRU2006 において，音声，唇動作の 2 つのモダリティを用いた多重バイオメトリックシステムについて発表する予定である [16]．

文　献

[1] J.P. Campbell,“Speaker recognition: A tutorial,” Proc.IEEE, vol.85, no.9, pp.1437-1462, 1997.

[2] K.Yu, J.Mason and J.Oglesby,“Speaker recognition using hidden Markov models, dynamic time warping and vector quantisation,” Vision, Image and Signal Processing, IEE Proceedings- vol.142, issue 5, pp.313 - 318, October 1995.

[3] R.A. Reynolds and R.C. Rose,“Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models,” Speech and Audio Processing, IEEE Transactions on , Volume:3, Issue:1, pp.72-83, January 1995.

[4] 市野将嗣，高倉大樹，坂野　鋭，小松尚久，“母音音素分布の非線形性について，”信学総大，D-14-14，March 2004.

[5] 津田宏治，“ヒルベルト空間における部分空間，”信学論 (D-II)，vol.J82-D-II，no.4，pp.592-599，April 1999.

[6] 前田英作，村瀬　洋，，“カーネル非線形部分空間法によるパターン認識．”信学論 (D-II)，vol.J82-D-II，no.4，pp.600-612，April 1999.

[7] 浜崎　武，野田秀樹，河口英二，“ヒルベルト空間で部分空間法を用いた話者識別，”信学技報，PRMU2000-127,2000 年．

[8] 前田賢一，渡辺貞一，“局所構造を導入したパターン・マッチング法，”信学論（D），vol.J68-D，no.3，pp.345-352，March 1985.

[9] 山口　修，福井和広，前田賢一，“動画像を用いた顔認識システム，”信学技報，PRMU97-50，June 1998.

[10] 福井和広，山口　修，鈴木　薫，前田賢一，“制約相互部分空間法を用いた環境変動にロバストな顔画像認識—照明変動の影響を抑える制約部分空間の学習—，”信学論（D-II），vol.J82-D-II，no.4，pp.613-620，April 1999.

[11] B.Schölkopf *et al.*,“Nonlinear component analysis as a Kernel eigenvalue problem,” Neural Computation, vol.10, pp.1299-1319, 1998

[12] 坂野　鋭，武川直樹，中村太一，“核非線形相互部分空間法による物体認識，”信学論（D-II），vol.J84-D-II，no.8，pp.1549-1556，August 2001.

[13] 市野将嗣，坂野　鋭，小松尚久，“核非線形相互部分空間法の処理量削減に関する検討，”画像の認識・理解シンポジウム (MIRU2005)，pp.1035-1042，July 2005.

[14] K.Messer, J.Matas, J. Kittler, J. Luettin and G. Maitre, “XM2VTSDB: The extended M2VTS database,” in Proc. of Int. Conf. on Audio and Video based Biometric Person Authentication, Washington, USA,1999.

[15] 市野将嗣，坂野　鋭，小松尚久，“核非線形相互部分空間法による話者認識，”信学論 (D-II)，vol.J88-D-II，no.8，pp.1331-1338,August 2005.

[16] 市野将嗣，坂野　鋭，小松尚久，“唇動作と音声を用いた多重バイオメトリクスの認証器設計に関する一考察，”画像の認識・理解シンポジウム (MIRU2006)，July 2006 (発表予定).

マルチポート固有空間法

玉木　徹†　天野　敏之††

† 広島大学大学院工学研究科情報工学専攻　〒 739–8527 広島県東広島市鏡山 1–4–1
†† 名古屋工業大学大学院おもひ領域　〒 466–8555 名古屋市昭和区御器所町
E-mail: †tamaki@hiroshima-u.ac.jp, ††amano@nitech.ac.jp

あらまし　本稿では、多様体の教師付き問題としての姿勢パラメータ推定手法である「マルチポート固有空間法」[1], [2] について議論する。これは欠損画素の輝度値を推定して画像を補間する BPLP [3], [4] を基にしており、群の回帰問題を空間への投影という線形演算で行うものである。まずマルチポート固有空間法の内容を説明し、その主要部分は連立方程式による最小ノルム推定であることを示す。またその方程式による空間への投影がどのようなものかを説明し、学習とサンプル数の影響について述べる。

キーワード　パラメトリック固有空間法, マルチポート固有空間法, 多様体学習, 教師付き学習, 回帰, 固有空間, 線形写像, EbC

Multi-port Eigenspace Method

Toru TAMAKI† and Toshiyuki AMANO††

† Graduate School of Engineering, Hiroshima University
1–4–1 Kagamiyama, Higashi-Hiroshima-shi, Hiroshima, 739–8527, Japan
†† Omohi College, Graduate School of Engineering Nagoya Institute of Technology
Gokiso-cho, Showa-ku, Nagoya 466-8555 Japan
E-mail: †tamaki@hiroshima-u.ac.jp, ††amano@nitech.ac.jp

Abstract　In this paper, we discuss on Multi-port Eigenspace Method [1], [2], an supervised manifold learning of pose parameters. This method is based on BPLP [3], [4], a method of intensity interpolation, and operates a linear mapping of projection to a subspace as a regression to a group. First we describe the method, and show that the important part of it is a least norm solution of a system of equations. Then we illustrate the projection by the system, and the effect of the number of learning samples.

Key words　Parametric Eigenspace Method, Multiport Eigenspace Method, Manifold learning, supervised learning, regression, Eigenspace, linear mapping, EbC

1. はじめに

画像に写る物体の姿勢パラメータ推定はコンピュータビジョンの重要な問題として多くの研究がなされている。その多くは既知形状の剛体を仮定し、運動や投影の解析的モデルを採用して最適化問題を解くものである。それに対して村瀬ら [5], [6] のパラメトリック固有空間法は、形状や投影を仮定せず、連続的に変化するパラメータ列とそれに付随する画像系列を学習セットとし、その画像列が固有空間中に描く軌跡（多様体）を認識に用いる、いわばパターン認識的姿勢推定問題を提起した。

多様体を描く非線形性の強い画像列のようなデータを扱う手法は、現在二つある。一つは非線形主成分分析 (kernel PCA, kPCA) [7], [8] などのカーネルトリックを利用した手法である。もう一つは、多様体を低次元に展開する多様体学習 (manifold learning) [9] である。前者は直感的な主成分方向が得られないのに対して、後者は主観的に意味のありそうな低次元の軸を見つけることができるとされている。どちらも主な目的は教師なしのデータ解析である。

パラメトリック固有空間法が扱うのは教師付き問題や回帰である。しかし、カーネルトリックを用いたものには SVM 回帰 [10] などの回帰手法が提案されているのに対し、多様体学習で教師付き問題を扱っているものはほとんどない。多様体学習の火付け役となった LLE(Locally Linear Embedding) [11] や Isomap [12] は、もともと主成分分析 (PCA, Principal Component Analysis) や多次元尺度構成法 (MDS, Multidimensional Scaling) が非線形データに対処できない問題に対して提案さ

れた教師なし学習である。その後も Hessian Eigenmap [13], Laplacian eigenmap [14] などが出現しているが、やはり非線形データである多様体を扱う教師なし学習法である。

本稿で議論する内容は、パラメトリック固有空間法と同様に、多様体を扱う教師付き問題としての姿勢パラメータ推定手法「マルチポート固有空間法」である。これは欠損画素の輝度値を推定して画像を補間する BPLP(Back Projection for Lost Pixels) [3], [4] を基に、2000 年に天野が提案 [1] し、その後 EbC(Estimation-by-Completion) 法 [2] として整理された。この手法（以下本手法）は次のような性質を持つ。

まず、数直線への回帰ではなく、群の元への回帰問題を扱うことである。通常の回帰問題は

$$y = f(x), \boldsymbol{x} \in R^N, y \in R$$

という形式である。しかし姿勢パラメータは、1 軸回転なら S^1（3 次元の回転であれば $SO(3)$）で表される回転群への回帰であり、

$$y = f(x), \boldsymbol{x} \in C \hookrightarrow R^N, y \in S^1$$

という形式になる。ここで C は R^N に埋め込まれた多様体であり、$\hookrightarrow$ は埋め込みを表す。多様体と群の対応を見つけ出す手法には、Gong ら [15] によって提案された APD (Anchor Points Diffusion) がある。しかし APD は semi-supervised の多様体学習であり、推定された対応関係では位相が保存されないため、回帰問題を扱うには不向きである。

次に、空間への投影という線形計算で推定できることである。一般に部分空間を用いた認識では、未知サンプルを部分空間に投影するため、線形計算というメリットがある。しかし、パラメトリック固有空間法のパラメータ推定にはスプライン曲線との最近傍探索が必要になってしまう。また多様体学習の多くは、学習サンプルによる多様体を表現するための手法であり、未知サンプルが与えられた場合には基本的には再計算を行わなければならない。最近になって、再計算をせず逐次的に学習した多様体を更新する手法 [16], [17] も提案されてはいるが、更新の必要がない未知サンプル識別ということは考慮されていない。未知サンプルと学習された多様体の関係を指摘した He ら [18] は、線形計算を用いた教師なし多様体学習である NPE (Neighborbood Preserving Embedding) を提案しているが、回帰問題を扱うものではない。本手法は部分空間への投影を用いた線形計算で回帰問題を扱うものである。

図 1 に、1 軸回転の場合の本手法の概念図を示す。この場合推定パラメータは回転角度 φ の 1 パラメータだけであり、これはある 2 次元平面上の単位円周 S^1 上の点で表されるとみなすことができる。本手法は、画像列がなす多様体 C から S^1 への線形写像を求めるものである。

本稿ではマルチポート固有空間法について議論するものであり、[2] の補遺にあたるものである。以下 2 節ではマルチポート固有空間法を説明し、3 節では本手法の主要部分は連立方程式による最小ノルム推定であることを示す。4 節では図 1 で示したような平面への投影がどのようなものかを説明し、5 節では学習のサンプル数の影響について述べる。

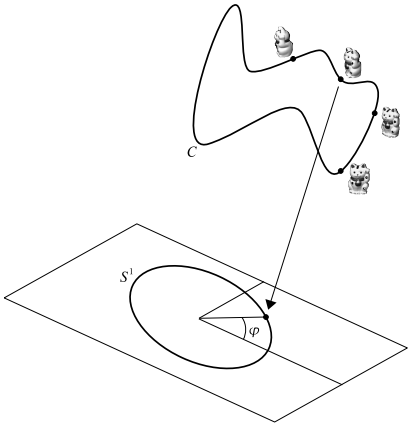


図 1　1 パラメータの場合のマルチポート固有空間法の概念図

2. 手法の概要

ここではマルチポート固有空間法の概要について述べることにする。

2.1 問題設定

ある n^{in} 次元入力ベクトル $\boldsymbol{x}^{\text{in}}$ が、それに付随するパラメータ φ とともに与えられたとき、そのパラメータを表す n^{out} 次元出力ベクトル $\boldsymbol{x}^{\text{out}}$ を考える。ここで $\boldsymbol{x}^{\text{in}}$ は角度 $\varphi \in S^1 = [0, 2\pi)$ の方向から撮影された学習画像であり、座標 $i (i = 1, \ldots, n^{\text{in}})$ の輝度値を $\boldsymbol{x}^{\text{in}}[i]$ とするベクトルである。また $\boldsymbol{x}^{\text{out}}$ は角度 φ の位相を持つ正弦波(注1) であり、以下の式で表される。

$$\boldsymbol{x}^{\text{out}}[i] = K \sin(\omega i - \varphi) + \text{offset} \tag{1}$$

ここで $i = 1, \ldots, n^{\text{out}}$ である。

この $\boldsymbol{x}^{\text{out}}$ を $\boldsymbol{x}^{\text{in}}$ に対するトラック情報と呼ぶ。$\boldsymbol{x}^{\text{out}}$ と $\boldsymbol{x}^{\text{in}}$ を対にして固有空間を生成し、$\boldsymbol{x}^{\text{in}}$ のみからトラック情報である $\boldsymbol{x}^{\text{out}}$ を BPLP により推定（復元）することがマルチポート固有空間法である。

2.2 固有空間の学習

次に学習について説明する。入力ベクトル $\boldsymbol{x}^{\text{in}}$ とトラック情報 $\boldsymbol{x}^{\text{out}}$ を対にしたベクトル

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}^{\text{in}} \\ \boldsymbol{x}^{\text{out}} \end{pmatrix} \tag{2}$$

を一つの n 次元学習ベクトル $(n = n^{\text{in}} + n^{\text{out}})$ として考え、M 個の学習ベクトル $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_M\}$ から固有空間 E を生成する（詳細は省略）。

この固有空間 E の第 i 固有ベクトルを $\boldsymbol{e}_i (i = 1, 2, \ldots, M)$ とすると、学習ベクトルと同様に

$$\boldsymbol{e}_i = \begin{pmatrix} {\boldsymbol{e}^{\text{in}}}_i \\ {\boldsymbol{e}^{\text{out}}}_i \end{pmatrix} \tag{3}$$

と分解できる。

(注1)：周波数は $\omega = \frac{k}{n^{\text{out}}}\pi$ とし、通常は $k = 2$ つまり出力ベクトルの長さが波長になるものを用いるが、$k = 1$ などでもよい。

ここで固有空間 E を表す以下の $n \times M'$ 行列 E （ここで $M' \leqq M$）を考える。

$$E = (\boldsymbol{e}_1 \ \boldsymbol{e}_2 \ \ldots \ \boldsymbol{e}_{M'}) \tag{4}$$

すると、同様に

$$E = \begin{pmatrix} E^{\mathrm{in}} \\ E^{\mathrm{out}} \end{pmatrix} \tag{5}$$

と分解できる。ここで、E^{in} は $n^{\mathrm{in}} \times M'$ 行列、E^{out} は $n^{\mathrm{out}} \times M'$ 行列である。

2.3 パラメータの推定

新たな未知データ $\boldsymbol{x}^{\mathrm{in}}$ が与えられると、BPLP より

$$\boldsymbol{x}^* = E(E^T\Sigma^{\mathrm{in}}E)^{-1}E^T\Sigma^{\mathrm{in}} \begin{pmatrix} \boldsymbol{x}^{\mathrm{in}} \\ 0 \end{pmatrix} \tag{6}$$

によって復元データ $\boldsymbol{x}^*$ が得られる。ここで

$$\Sigma^{\mathrm{in}} = \begin{pmatrix} I_{n^{\mathrm{in}}} & 0 \\ 0 & 0 \end{pmatrix} \tag{7}$$

は $n \times n$ 行列であり、$I_{n^{\mathrm{in}}}$ は n^{in} 次元の単位行列である。

ここで式 (6) を以下のように変形する。

$$\begin{aligned}
\boldsymbol{x}^* &= E(E^T\Sigma^{\mathrm{in}}E)^{-1}E^T\Sigma^{\mathrm{in}}(\boldsymbol{x}^{\mathrm{in}}0) && (8)\\
&= E(E^T\Sigma^{\mathrm{in}}\Sigma^{\mathrm{in}}E)^{-1}{E^{\mathrm{in}}}^T\boldsymbol{x}^{\mathrm{in}} && (9)\\
&= E(E^T{\Sigma^{\mathrm{in}}}^T\Sigma^{\mathrm{in}}E)^{-1}{E^{\mathrm{in}}}^T\boldsymbol{x}^{\mathrm{in}} && (10)\\
&= E((\Sigma^{\mathrm{in}}E)^T(\Sigma^{\mathrm{in}}E))^{-1}{E^{\mathrm{in}}}^T\boldsymbol{x}^{\mathrm{in}} && (11)\\
&= E\begin{pmatrix} E^{\mathrm{in}} \\ 0 \end{pmatrix}^T \begin{pmatrix} E^{\mathrm{in}} \\ 0 \end{pmatrix}^{-1} {E^{\mathrm{in}}}^T\boldsymbol{x}^{\mathrm{in}} && (12)\\
&= E({E^{\mathrm{in}}}^TE^{\mathrm{in}})^{-1}{E^{\mathrm{in}}}^T\boldsymbol{x}^{\mathrm{in}} && (13)
\end{aligned}$$

さらに、$\boldsymbol{x}^*$ のうちトラック情報部分を抽出する $n \times n$ 行列

$$\Sigma^{\mathrm{out}} = \begin{pmatrix} 0 & 0 \\ 0 & I_{n^{\mathrm{out}}} \end{pmatrix} \tag{14}$$

を定義する。すると、

$$\boldsymbol{x}^* = \begin{pmatrix} \boldsymbol{x}^{\mathrm{in}*} \\ \boldsymbol{x}^{\mathrm{out}*} \end{pmatrix} \tag{15}$$

と分解できるので、

$$\begin{aligned}
\boldsymbol{x}^{\mathrm{out}*} &= \Sigma^{\mathrm{out}}\boldsymbol{x}^* && (16)\\
&= \Sigma^{\mathrm{out}}E({E^{\mathrm{in}}}^TE^{\mathrm{in}})^{-1}{E^{\mathrm{in}}}^T\boldsymbol{x}^{\mathrm{in}} && (17)\\
&= (\Sigma^{\mathrm{out}}E)({E^{\mathrm{in}}}^TE^{\mathrm{in}})^{-1}{E^{\mathrm{in}}}^T\boldsymbol{x}^{\mathrm{in}} && (18)\\
&= E^{\mathrm{out}}({E^{\mathrm{in}}}^TE^{\mathrm{in}})^{-1}{E^{\mathrm{in}}}^T\boldsymbol{x}^{\mathrm{in}} && (19)
\end{aligned}$$

とトラック情報の推定値 $\boldsymbol{x}^{\mathrm{out}*}$ が得られる。

2.4 位相の推定

トラック情報の推定値ベクトル

$$\boldsymbol{x}^{\mathrm{out}*} = (\boldsymbol{x}^{\mathrm{out}*}[0], \boldsymbol{x}^{\mathrm{out}*}[1], \ldots, \boldsymbol{x}^{\mathrm{out}*}[n^{\mathrm{out}}-1])^T \tag{20}$$

が得られたとき、次のスカラー量 S, C を定義する。

$$S = \sum_{i=0}^{n^{\mathrm{out}}-1} \boldsymbol{x}^{\mathrm{out}*}[i]\sin i\omega \tag{21}$$

$$C = \sum_{i=0}^{n^{\mathrm{out}}-1} \boldsymbol{x}^{\mathrm{out}*}[i]\cos i\omega \tag{22}$$

すると、位相 φ は次式で求めることができる。

$$\varphi = \tan^{-1}\left(\frac{C}{S}\right) \tag{23}$$

詳細は付録 1. 節を参照。

2.5 内積による位相推定

前項で示した方法は、一旦 $\boldsymbol{x}^{\mathrm{out}}$ を推定してから A, B を計算するものであった。ここでは、$\boldsymbol{x}^{\mathrm{in}}$ から直接 A, B を計算する方法を示す。

まずベクトル

$$\boldsymbol{s} = (\sin 0, \sin\omega, \sin 2\omega, \ldots, \sin((n^{\mathrm{out}}-1)\omega))^T \tag{24}$$

$$\boldsymbol{c} = (\cos 0, \cos\omega, \cos 2\omega, \ldots, \cos((n^{\mathrm{out}}-1)\omega))^T \tag{25}$$

を定義する。すると、式 (21),(22) を変形して

$$\begin{aligned}
S &= \sum_{i=0}^{n^{\mathrm{out}}-1} \boldsymbol{x}^{\mathrm{out}*}[i]\sin i\omega && (26)\\
&= \boldsymbol{s}^T\boldsymbol{x}^{\mathrm{out}*} && (27)\\
&= \boldsymbol{s}^T E^{\mathrm{out}}({E^{\mathrm{in}}}^TE^{\mathrm{in}})^{-1}{E^{\mathrm{in}}}^T\boldsymbol{x}^{\mathrm{in}} && (28)\\
&= \boldsymbol{s}'^T\boldsymbol{x}^{\mathrm{in}} && (29)
\end{aligned}$$

ここで $\boldsymbol{s}'^T = \boldsymbol{s}^T E^{\mathrm{out}}({E^{\mathrm{in}}}^TE^{\mathrm{in}})^{-1}{E^{\mathrm{in}}}^T$ とする。同様に

$$\begin{aligned}
C &= \sum_{i=0}^{n^{\mathrm{out}}-1} \boldsymbol{x}^{\mathrm{out}*}[i]\cos i\omega && (30)\\
&= \boldsymbol{c}^T\boldsymbol{x}^{\mathrm{out}*} && (31)\\
&= \boldsymbol{c}^T E^{\mathrm{out}}({E^{\mathrm{in}}}^TE^{\mathrm{in}})^{-1}{E^{\mathrm{in}}}^T\boldsymbol{x}^{\mathrm{in}} && (32)\\
&= \boldsymbol{c}'^T\boldsymbol{x}^{\mathrm{in}} && (33)
\end{aligned}$$

ここで $\boldsymbol{c}'^T = \boldsymbol{c}^T E^{\mathrm{out}}({E^{\mathrm{in}}}^TE^{\mathrm{in}})^{-1}{E^{\mathrm{in}}}^T$ とする。

$\boldsymbol{s}', \boldsymbol{c}'$ はともに n^{in} 次元のベクトルであり、入力された未知データ $\boldsymbol{x}^{\mathrm{in}}$ と $\boldsymbol{s}', \boldsymbol{c}'$ の内積をとることで S, C が計算できることになる。$\boldsymbol{s}', \boldsymbol{c}'$ は未知データに依存せず、固有空間を求めた時点であらかじめ計算しておけるので、未知の入力 $\boldsymbol{x}^{\mathrm{in}}$ が変わっても内積だけで計算が可能である。

2.6 複数パラメータの推定

上記のように、角度 φ だけでなく位置 x, y などの他のパラメータも同時に推定するためには、それらに対応するトラック情報を付加する。

つまり

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}^{\mathrm{in}} \\ \boldsymbol{x}^{\mathrm{out}} \\ \boldsymbol{x}^{\mathrm{out2}} \\ \boldsymbol{x}^{\mathrm{out3}} \end{pmatrix} \tag{34}$$

として学習し、推定はそれぞれのパラメータで別々に行う。導出の詳細と実際の推定実験結果は [2] を参照。

このように、一つのパラメータ (single port) だけでなく、複数のパラメータ (multi-port) を推定するため、マルチポート固有空間法と呼ぶ。

3. 連立方程式の最小ノルム推定について

式 (19) は線型方程式であり、

$$\boldsymbol{x}^{\mathrm{out}*} = A\boldsymbol{x}^{\mathrm{in}} \tag{35}$$

と書き直すことができる。つまり、この行列 A を推定すればよいことになる。ここでは、連立方程式の A を推定することが式 (19) と等価であることを（条件付で）示す。つまり本手法は、

$$Z^{\mathrm{out}} = [\boldsymbol{x}^{\mathrm{out}}{}_1, \boldsymbol{x}^{\mathrm{out}}{}_2, \ldots, \boldsymbol{x}^{\mathrm{out}}{}_M] \tag{36}$$

$$Z^{\mathrm{in}} = [\boldsymbol{x}^{\mathrm{in}}{}_1, \boldsymbol{x}^{\mathrm{in}}{}_2, \ldots, \boldsymbol{x}^{\mathrm{in}}{}_M] \tag{37}$$

を与えて、

$$Z^{\mathrm{out}} = AZ^{\mathrm{in}}, \quad A = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_{n^{\mathrm{out}}}]^T \tag{38}$$

を満たす $n^{\mathrm{out}} \times n^{\mathrm{in}}$ の行列 A を求めることと同じである。ここで各 $\boldsymbol{a}_j$ は n^{in} 次元の列ベクトルである。

まず列ベクトルを並べた行列 Z^{out} の行を取り

$$Z^{\mathrm{out}} \equiv [\boldsymbol{z}_1^{\mathrm{out}}, \boldsymbol{z}_2^{\mathrm{out}}, \ldots, \boldsymbol{z}_{n^{\mathrm{out}}}^{\mathrm{out}}]^T \tag{39}$$

とする。すると式 (38) の j 行目は

$$\boldsymbol{z}_j^{\mathrm{out}T} = \boldsymbol{a}_j^T Z^{\mathrm{in}} \tag{40}$$

$$\boldsymbol{z}_j^{\mathrm{out}} = Z^{\mathrm{in}T}\boldsymbol{a}_j \tag{41}$$

となる。ここで最小ノルム型一般逆行列を用いると

$$\boldsymbol{a}_j = Z^{\mathrm{in}}(Z^{\mathrm{in}T}Z^{\mathrm{in}})^{-1}\boldsymbol{z}_j^{\mathrm{out}} \tag{42}$$

と表せる。したがって、

$$[\boldsymbol{a}_1\ \boldsymbol{a}_2\ \cdots] = Z^{\mathrm{in}}(Z^{\mathrm{in}T}Z^{\mathrm{in}})^{-1}[\boldsymbol{z}_1^{\mathrm{out}}\ \boldsymbol{z}_2^{\mathrm{out}}\ \cdots] \tag{43}$$

$$A^T = Z^{\mathrm{in}}(Z^{\mathrm{in}T}Z^{\mathrm{in}})^{-1}Z^{\mathrm{out}T} \tag{44}$$

$$A = Z^{\mathrm{out}}(Z^{\mathrm{in}T}Z^{\mathrm{in}})^{-T}Z^{\mathrm{in}T} \tag{45}$$

となる。

ここで

$$Z = \begin{bmatrix} Z^{\mathrm{in}} \\ Z^{\mathrm{out}} \end{bmatrix} \tag{46}$$

なる行列 Z を考える。これを特異値分解して

$$Z = EDV^T \tag{47}$$

とすると、式 (5) と同様に

$$E = \begin{pmatrix} E^{\mathrm{in}} \\ E^{\mathrm{out}} \end{pmatrix} \tag{48}$$

と書けるので、

$$Z^{\mathrm{in}} = E^{\mathrm{in}}DV^T \tag{49}$$

$$Z^{\mathrm{out}} = E^{\mathrm{out}}DV^T \tag{50}$$

となる。これを式 (45) へ代入すると、

$$A = E^{\mathrm{out}}DV^T \left((E^{\mathrm{in}}DV^T)^T E^{\mathrm{in}}DV^T\right)^{-T}(E^{\mathrm{in}}DV^T)^T \tag{51}$$

$$= E^{\mathrm{out}}DV^T \left((DV^T)^{-1}(E^{\mathrm{in}T}E^{\mathrm{in}})^{-1}(VD^T)^{-1}\right)^T VD^T E^{\mathrm{in}T} \tag{52}$$

$$= E^{\mathrm{out}}DV^T (V^TD)^{-T}(E^{\mathrm{in}T}E^{\mathrm{in}})^{-T}(D^TV)^{-T}VD^TE^{\mathrm{in}T} \tag{53}$$

$$= E^{\mathrm{out}}(E^{\mathrm{in}T}E^{\mathrm{in}})^{-1}E^{\mathrm{in}T} \tag{54}$$

したがって、これは式 (19) と同等であり、本手法は入力と出力のベクトルをノルム最小の線形関係で結びつけるものに等しい。ただし上記の議論では、$M = M' > n^{\mathrm{in}} > n^{\mathrm{out}}$、つまり次元削減がなくサンプル数が入力次元より大きいことを仮定している。

計算コストの比較は [2] を参照。

4. 平面への投影について

前節で示したように、本手法の主要部分は連立方程式であり、また付録に示すように、複素平面上（もっと一般に R^2 平面）の単位円への投影と見ることができる。ここでは、連立方程式とそのような投影との関係を、見通しをよくするために 1 軸回転の姿勢パラメータ推定に議論を絞り、本手法の単位円への写像がどのようなものかを説明する。

式 (38) より、入出力関係は

$$\boldsymbol{x}^{\mathrm{out}}{}_i = \begin{pmatrix} \boldsymbol{a}_1^T\boldsymbol{x}^{\mathrm{in}}{}_i \\ \boldsymbol{a}_2^T\boldsymbol{x}^{\mathrm{in}}{}_i \\ \vdots \\ \boldsymbol{a}_j^T\boldsymbol{x}^{\mathrm{in}}{}_i \\ \vdots \\ \boldsymbol{a}_{n^{\mathrm{out}}}^T\boldsymbol{x}^{\mathrm{in}}{}_i \end{pmatrix} \tag{55}$$

である。ここで本手法では、トラック情報 $\boldsymbol{x}^{\mathrm{out}}{}_i$ は与えられたパラメータに対応する位相 φ_i をもつ周期 ω の正弦波であるので、以下のようなベクトルである。

$$\boldsymbol{x}^{\mathrm{out}}{}_i = \Big(\sin(\omega-\varphi_i), \sin(2\omega-\varphi_i), \ldots, \sin(j\omega-\varphi_i), \ldots, \sin(N\omega-\varphi_i)\Big)^T \tag{56}$$

$$= \left(\sin\left(\frac{1}{N}\pi-\varphi_i\right), \sin\left(\frac{2}{N}\pi-\varphi_i\right), \ldots, \sin\left(\frac{j}{N}\pi-\varphi_i\right), \ldots, \sin\left(\frac{N}{N}\pi-\varphi_i\right)\right)^T \tag{57}$$

ただし情報トラックの長さ n^{out} が半周期分になるような周期 $\omega = \frac{\pi}{n^{\mathrm{out}}}$ を考え、見易さのため $N = n^{\mathrm{out}}$ とおく。

すると、式 (55) の j 行に注目すると

$$\boldsymbol{a}_j^T \boldsymbol{x}^{\mathrm{in}}{}_i = \sin\left(\frac{j}{N}\pi-\varphi_i\right) \tag{58}$$

である。これは平面の方程式とみなすことができる。すなわち、法線が $\boldsymbol{a}_j$ で、原点からの距離が $\sin\left(\frac{j}{N}\pi-\varphi_i\right)$ である超平面が、点 $\boldsymbol{x}^{\mathrm{in}}{}_i$ を通る（もしくはその超平面上に $\boldsymbol{x}^{\mathrm{in}}{}_i$ が乗っている）ことを意味する。(注2)

同様に式 (55) のすべての行を考えると、法線がそれぞれ $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N$ である超平面がすべて点 $\boldsymbol{x}^{\mathrm{in}}{}_i$ を通り、それぞれの超平面の原点からの距離が $\sin\left(\frac{1}{N}\pi-\varphi_i\right), \ldots, \sin\left(\frac{N}{N}\pi-\varphi_i\right)$ である。

また出力の次元は入力の次元よりも小さい（$n^{\mathrm{in}} > n^{\mathrm{out}} = N$）ため、高々$\mathrm{rank}(A) = N$ である。1 軸回転の姿勢パラメータ推定の場合、出力は R^2 上の単位円と考えられるため、実際には $\mathrm{rank}(A) = 2$ である。つまり各法線ベクトル $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N$ のうち 2 つが独立である。

4.1 $N = 2$ の場合

$N = 2$、つまり情報トラックのベクトルの要素が 2 の場合、以下のようになる。

$$\boldsymbol{a}_1^T \boldsymbol{x}^{\mathrm{in}}{}_i = \sin\left(\frac{1}{2}\pi-\varphi_i\right) = \cos\varphi_i \tag{59}$$

$$\boldsymbol{a}_2^T \boldsymbol{x}^{\mathrm{in}}{}_i = \sin\left(\frac{2}{2}\pi-\varphi_i\right) = \sin\varphi_i \tag{60}$$

すなわち、行列 A によって点 $\boldsymbol{x}^{\mathrm{in}}$ は法線 $\boldsymbol{a}_1, \boldsymbol{a}_2$ が張る平面に投影される（図 2 参照）。また原点からの距離が $\sin\varphi_i, \cos\varphi_i$ であるため、二つの平面は直交する（つまり $\boldsymbol{a}_1 \perp \boldsymbol{a}_2$）。

4.2 $N > 2$ の場合

$N > 2$、つまり情報トラックのベクトルの要素が 2 より大きくても、1 軸回転のパラメータだけであれば、図 3 のようになる。つまり行列 A によって、点 $\boldsymbol{x}^{\mathrm{in}}$ は法線 $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N$ が張る 2 次元平面（$\mathrm{rank}(A) = 2$ なので）に投影される。また $\boldsymbol{x}^{\mathrm{in}}$ を通る平面群は、原点からの距離が sin に従うため、それぞれ等角度 $\frac{\pi}{N}$ で交わる。

4.3 内積をとった場合

2.5 節において、出力トラック情報ベクトルと $\boldsymbol{s}, \boldsymbol{c}$ との内積について述べた。これは $N > 2$ の場合に出力を S, C の 2 要素にする操作であるが、前項と同様に解釈することができる。

まず $\boldsymbol{s}, \boldsymbol{c}$ を以下のようにとる。

(注2)：距離が負になる場合には、法線ベクトルが逆を向くと考えればよい。

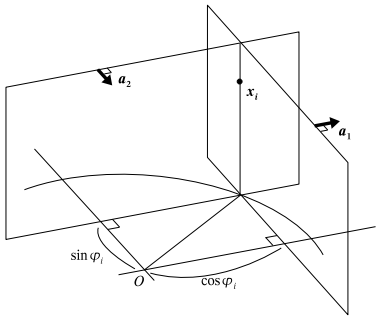


図 2　$N = 2$ の場合

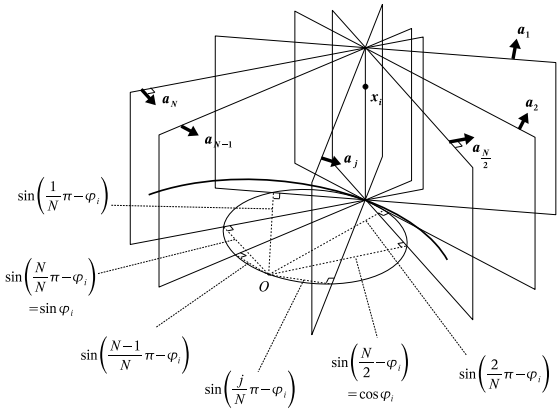


図 3　$N > 2$ の場合

$$\boldsymbol{s} = \left(\sin\frac{1}{N}\pi,\ \sin\frac{2}{N}\pi,\ \ldots,\ \sin\frac{N}{N}\pi\right)^T \tag{61}$$

$$\boldsymbol{c} = \left(\cos\frac{1}{N}\pi,\ \cos\frac{2}{N}\pi,\ \ldots,\ \cos\frac{N}{N}\pi\right)^T \tag{62}$$

これを式 (55) の両辺にかける。$\boldsymbol{s}$ の場合、まず左辺は

$$\boldsymbol{s}^T \boldsymbol{x}^{\mathrm{out}}{}_i = \sum_j^N \sin\frac{j}{N}\pi\ \sin\left(\frac{j}{N}\pi-\varphi_i\right) \tag{63}$$

$$= \frac{1}{2}\sum_j^N \left\{\cos\left(\frac{2j}{N}\pi-\varphi_i\right) - \cos\varphi_i\right\} \tag{64}$$

$$= -\frac{N}{2}\cos\varphi_i \tag{65}$$

となり、右辺は

$$\boldsymbol{s}^T \begin{pmatrix} \boldsymbol{a}_1^T \boldsymbol{x}^{\mathrm{in}}{}_i \\ \boldsymbol{a}_2^T \boldsymbol{x}^{\mathrm{in}}{}_i \\ \vdots \\ \boldsymbol{a}_j^T \boldsymbol{x}^{\mathrm{in}}{}_i \\ \vdots \\ \boldsymbol{a}_{n^{\mathrm{out}}}^T \boldsymbol{x}^{\mathrm{in}}{}_i \end{pmatrix} = \sum_j^N \left\{\sin\frac{j}{N}\pi(\boldsymbol{a}_j^T \boldsymbol{x}^{\mathrm{in}}{}_i)\right\} \tag{66}$$

$$= \left(\sum_j^N \sin\frac{j}{N}\pi \boldsymbol{a}_j^T\right) \boldsymbol{x}^{\mathrm{in}}{}_i \tag{67}$$

したがって、

$$\left(\sum_j^N \sin\frac{j}{N}\pi\ \boldsymbol{a}_j^T\right)\boldsymbol{x}^{\mathrm{in}}{}_i = -\frac{N}{2}\cos\varphi_i \tag{68}$$

となる。これは、$\boldsymbol{a}_1,\ldots,\boldsymbol{a}_N$ の sin による重み付け和を法線とし、原点からの距離が $\frac{N}{2}\cos\varphi_i$ である平面が点 $\boldsymbol{x}^{\mathrm{in}}$ を通ることを意味する。

$\boldsymbol{c}$ の場合も同様にすると、

$$\left(\sum_j^N \cos\frac{j}{N}\pi\ \boldsymbol{a}_j^T\right)\boldsymbol{x}^{\mathrm{in}}{}_i = -\frac{N}{2}\sin\varphi_i \tag{69}$$

が得られる。

式 (68) と式 (69) のうち、重み付け法線部分をそれぞれ $\boldsymbol{\alpha},\boldsymbol{\beta}$ とおいてこれらの式を書き直すと、以下の式が得られる。

$$\boldsymbol{\alpha}^T\boldsymbol{x}^{\mathrm{in}} = \sin\varphi_i,\quad \boldsymbol{\beta}^T\boldsymbol{x}^{\mathrm{in}} = \cos\varphi_i \tag{70}$$

これは $N=2$ の場合における式 (59)、式 (60) と同じ形式である。

つまり、$N>2$ の場合であっても、$\boldsymbol{s},\boldsymbol{c}$ との内積をとることによって $N=2$ と同様の平面への投影を得ている。

5. サンプル数と線形写像の学習について

前節までで、本手法の主要部分は線形連立方程式の最小ノルム推定であり、推定されたその線形写像が画像列がなす多様体上の点を 2 次元平面上の単位円に写像することを説明した。ここでは、実際にそのような写像が得られるのかどうかを、次元とサンプル数の点から説明する。

本手法の入出力を関係づける式 (38) は

$$[\boldsymbol{x}^{\mathrm{out}}{}_1,\ldots,\boldsymbol{x}^{\mathrm{out}}{}_M] = \begin{pmatrix}\boldsymbol{a}_1^T\\ \vdots\\ \boldsymbol{a}_{n^{\mathrm{out}}}^T\end{pmatrix}[\boldsymbol{x}^{\mathrm{in}}{}_1,\ldots,\boldsymbol{x}^{\mathrm{in}}{}_M] \tag{71}$$

であるが、一部分だけに注目すると

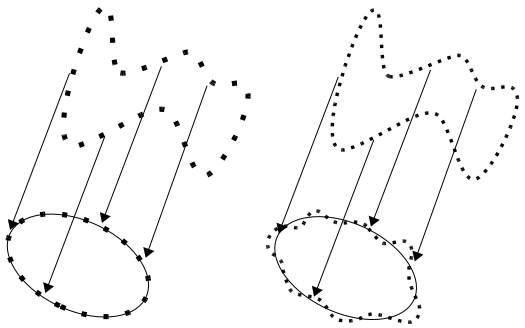
$$\boldsymbol{x}^{\mathrm{out}}{}_i = \boldsymbol{a}_j^T\boldsymbol{x}^{\mathrm{in}}{}_i \tag{72}$$

である。ここで未知数は n^{in} 次元ベクトル $\boldsymbol{a}_j$ なので、式が n^{in} あれば一意に決まる。しかし $i=1,\ldots,M$ なので、$M<n^{\mathrm{in}}$ ならば式 (71) を満たす無数の解が存在する。

つまり、学習サンプル数 M が、入力画像の次元 n^{in} よりも小さい場合、学習セットを完全に記述する行列 A が存在し、この行列により学習画像の多様体上のサンプル点は厳密に 2 次元平面上の単位円に投影される。

学習サンプル数 M が、入力画像の次元 n^{in} よりも大きい場合、学習画像は厳密には 2 次元平面上の単位円周上には写像されず、その付近に（最小二乗の意味で）投影される。

[2] の実験に用いている COIL-20 [19] は、画像サイズが $n^{\mathrm{in}}=128\times128$ であり、サンプル数は $M=72$ 枚なので、十分に上記の条件を満たしている。しかし、学習画像は単位円に厳密に投影できてはいるが、学習セットにない未知サンプルを投影した場合に単位円周上に乗るという保証はない。ただしある条件が満たされる場合には、単位円周上に乗り未知サンプルの推定が行われる（[2] 参照）。



(a) (b)

図 4　サンプル数と学習される投影の関係。(a) $M<n^{\mathrm{in}}$ の場合。(b) $M>n^{\mathrm{in}}$ の場合。

6. おわりに

本稿では、線形写像によって多様体の教師付き学習をおこなうマルチポート固有空間法について議論した。

本稿での内容をまとめると、以下のようになる。

- マルチポート固有空間法は学習時の入出力ベクトルを連結して固有空間による学習を行い（2.2 節）、BPLP により推定を行う（2.3 節）ものである。出力ベクトルが冗長であっても、ベクトルの内積により冗長さを除いている（2.5 節）。
- 本手法の処理は、学習時の入出力を線形写像で関係付けた連立方程式の最小ノルム解推定に等しい（3. 節）。
- 1 軸回転パラメータの場合、付属する出力ベクトル（情報トラック）を正弦波として定義しているため、線形写像によって入力ベクトルは平面上の単位円周 S^1 上に投影される（4. 節）。
- 学習サンプル数 M が入力画像の次元 n^{in} よりも小さい場合、線形写像 A は無数に存在し、学習時の入力ベクトルは単位円への投影は厳密にされる（5. 節）。

本手法は、画像補間のための手法である BPLP [4] から出発している。つまり、欠損画素の輝度値推定を出力ベクトルによるパラメータ推定に置き換えたものである。これが学習の入出力ベクトルを連結するということと同じになっている。学習の入出力ベクトルを連結するというアイデアは素朴なものであり、Active Appearance Model [20] や Eigen-points [21] などにも同様のものが見られる。DCCA (Dynamic Coupled component Analysis) [22] を提案した Torre らは、このようにして関係付られた場合には一方から他方の推定は最適なものではないと指摘している [22]。しかし、3. 節で述べたように、この考え方は（次元を削減しなければ）連立方程式による推定と同じであり、この意味では最適な推定になっていると思われる。逆に、出口ら [23]〜[25] は入出力を線形写像で結び付けており、こちらも同様に素朴な考え方である。

次元を削減した場合にはどうなるだろうか。出口ら [24], [25] は入出力を線形写像で直接関連付けた場合の学習能力とパラメトリック固有空間法に対して、本節と同様の議論を行っている。それによると、固有空間による次元削減が汎化性能を向上させているとされているが、本手法における学習の汎化性能も同様に議論することができると思われる。また学習サンプル数と入

力の次元についても、本稿と同じ議論がなされており [24], [25]、学習サンプル数が入力の次元より小さい場合には（当然ながら）無数の解が存在することを指摘している。

また [2] でも述べているが、式 (13) と式 (19) より、

$$\boldsymbol{x}^{\mathrm{in}*} = E^{\mathrm{in}}(E^{\mathrm{in}T}E^{\mathrm{in}})^{-1}E^{\mathrm{in}T}\boldsymbol{x}^{\mathrm{in}} \equiv E^{\mathrm{in}}\boldsymbol{\gamma} \tag{73}$$

$$\boldsymbol{x}^{\mathrm{out}*} = E^{\mathrm{out}}(E^{\mathrm{in}T}E^{\mathrm{in}})^{-1}E^{\mathrm{in}T}\boldsymbol{x}^{\mathrm{in}} \equiv E^{\mathrm{out}}\boldsymbol{\gamma} \tag{74}$$

である。つまり入力の最良近似係数 $\boldsymbol{\gamma}$ を用いて、出力を近似していることになる。この考え方は Vetter らの Linear class [26] に見られる。彼らは 3 次元ベクトルに対して剛体変換を行っても不変であることを利用しているものであるが、細井ら [27] は 2 次元の固有画像に対して同様の処理を行っている。このままの形では、係数をそのまま用いるという考え方の有効性が見えにくいが、本稿で示したように、「入力の最良近似係数を出力の近似に用いる＝入出力を線形で結びつける＝入出力ベクトルを連結して固有空間を作成する」という形で見通せると思われる。

最後に、画像もしくはベクトルの一部分（入力）から他の部分（出力）を推定するという本手法の考え方は、もっと一般的に扱えるということを指摘しておく。Iwamura ら [28] は文字検出のために部分画像中の基点から特徴点への投票を行っている。投票と連立方程式は異なる処理ではあるが、どちらも入出力を関係付ける写像とみなすことで、共通した問題を解くことができるかもしれない。

謝　　辞

有益なコメントや議論をしていただいた新潟大学自然科学系情報理工学系列 山田寛喜氏、九州大学大学院システム情報科学研究院 内田誠一氏、広島大学大学院理学研究科数学専攻 田丸博士氏に感謝致します。

文　　献

[1] 天野敏之：マルチポート固有空間法による物体の位置・姿勢検出 — 数枚の画像相関からのパラメータ推定法の提案 —, 名古屋工業大学電気情報工学科佐藤研究室 内部資料 (2000.6.27).

[2] 天野敏之, 玉木徹：Estimation-by-Completion: 3 次元物体の線形姿勢推定手法, *MIRU2006* (2006), in print.

[3] 天野敏之, 井口征士：固有空間照合法を用いた BPLP による画像補間, *MIRU2000*, Vol. 1, pp. 217–222 (2000).

[4] 天野敏之, 佐藤幸男：固有空間法を用いた BPLP による画像補間, 電子情報通信学会論文誌 D-II, Vol. J85-D-II, No. 3, pp. 457–465 (2002), http://search.ieice.org/bin/summary.php?id=j85-d2_3_457&category=D&year=2002&lang=J&abst=.

[5] 村瀬洋, シュリーナイヤー：2 次元照合による 3 次元物体認識 — パラメトリック固有空間法 —, 電子情報通信学会論文誌 DII, Vol. J77-D2, No. 11, pp. 2179–2187 (1994), http://search.ieice.org/bin/summary.php?id=j77-d2_11_2179&category=D&year=1994&lang=J&abst=.

[6] Murase, H. and Nayar, S. K.: Visual learning and recognition of 3-D objects from appearance, *IJCV*, Vol. 14, No. 1, pp. 5–24 (1995), http://dx.doi.org/10.1007/BF01421486.

[7] Scholkopf, B., Smola, A. and Müller, K.-R.: Nonlinear Component Analysis as a Kernel Eigenvalue Problem, *Neural Computation*, Vol. 10, pp. 1299–1319 (1998), http://neco.mitpress.org/cgi/content/abstract/10/5/1299.

[8] 坂野鋭：パターン認識における主成分分析 — 顔画像認識を例として —, 統計数理, Vol. 49, No. 1, pp. 23–42 (2001), http://www.ism.ac.jp/editsec/toukei/pdf/49-1-023.pdf.

[9] Law, M.: Manifold Learning, online (accessed 2006.6.28), http://www.cse.msu.edu/~lawhiu/manifold/.

[10] Smola, A. J. and Schölkopf, B.: A Tutorial on Support Vector Regression, NeuroCOLT Technical Report NC-TR-98-030, Royal Holloway College, University of London, UK (1998), http://www.kernel-machines.org/papers/tr-30-1998.ps.gz.

[11] Roweis, S. T. and Saul, L. K.: Nonlinear Dimensionality Reduction by Locally Linear Embedding, *Science*, Vol. 290, No. 5500, pp. 2323–2326 (2000), http://www.sciencemag.org/cgi/content/abstract/290/5500/2323.

[12] Tenenbaum, J. B., Silva, de V. and Langford, J. C.: A Global Geometric Framework for Nonlinear Dimensionality Reduction, *Science*, Vol. 290, No. 5500, pp. 2319–2323 (2000), http://www.sciencemag.org/cgi/content/abstract/290/5500/2319.

[13] Donoho, D. L. and Grimes, C.: Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data, *PNAS*, Vol. 100, No. 10, pp. 5591–5596 (2003), http://www.pnas.org/cgi/doi/10.1073/pnas.1031596100.

[14] Belkin, M. and Niyogi, P.: Laplacian Eigenmaps for Dimensionality Reduction and Data Representation, *Neural Computation*, Vol. 15, No. 6, pp. 1373–1396 (2003), http://people.cs.uchicago.edu/~misha/paper1.pdf.

[15] Gong, H., Pan, C., Yang, Q., Lu, H. and Ma, S.: A Semi-Supervised Framework for Mapping Data to the Intrinsic Manifold, *ICCV2005*, Vol. 1, pp. 98–105 (2005), http://doi.ieeecomputersociety.org/10.1109/ICCV.2005.18.

[16] Kouropteva, O., Okun, O. and Pietikäinen, M.: Incremental locally linear embedding, *Pattern Recognition*, Vol. 38, No. 10, pp. 1764–1767 (2005), http://dx.doi.org/10.1016/j.patcog.2005.04.006.

[17] Law, M. H. C. and Jain, A. K.: Incremental Nonlinear Dimensionality Reduction by Manifold Learning, *PAMI*, Vol. 28, No. 3, pp. 377–391 (2006), http://doi.ieeecomputersociety.org/10.1109/TPAMI.2006.56.

[18] He, X., Cai, D., Yan, S. and Zhang, H.-J.: Neighborhood Preserving Embedding, *ICCV2005*, Vol. 2, pp. 1208–1213 (2005), http://doi.ieeecomputersociety.org/10.1109/ICCV.2005.167.

[19] Nene, S. A., Nayar, S. K. and Murase, H.: Columbia Object Image Library(COIL-20), Technical Report CUCS-005-96, Columbia University (1996), http://www1.cs.columbia.edu/CAVE/software/softlib/coil-20.php.

[20] Cootes, T. F., Edwards, G. J. and Taylor, C. J.: Active Appearance Models, *ECCV98*, Vol. 2, pp. 484–498 (1998), http://www.isbe.man.ac.uk/~bim/Models/eccv98_aam.pdf.

[21] Covell, M.: Eigen-points: Control-point Location using Principle Component Analyses, *FG96*, pp. 122–127 (1996), http://doi.ieeecomputersociety.org/10.1109/AFGR.1996.557253.

[22] Torre, la F. D. and Black, M. J.: Dynamic coupled component analysis, *CVPR2001*, Vol. 2, pp. 643–650 (2001), http://doi.ieeecomputersociety.org/10.1109/CVPR.2001.991024.

[23] Deguchi, K.: A Direct Interpretation of Dynamic Images with Camera and Object Motions for Vision Guided Robot Control, *IJCV*, Vol. 37, No. 1, pp. 7–20 (2000), http://dx.doi.org/10.1023/A:1008151528479.

[24] 出口光一郎, 岡谷貴之：固有空間法はなぜうまく働くか, 情報処理学会コンピュータビジョンとイメージメディア研究会, Vol. 2001, No. 66, pp. 1–8 (2001), http://fw8.bookpark.ne.jp/cm/ipsj/search.asp?flag=6&keyword=IPSJ-CVIM01128001&mode=PDF.

[25] Okatani, T. and Deguchi, K.: Yet Another Appearance-Based Method for Pose Estimation Based on a Linear Model, *MVA2000*, pp. 258–261 (2000).

[26] Vetter, T. and Poggio, T.: Linear Object Classes

and Image Synthesis From a Single Example Image, *PAMI*, Vol. 19, No. 7, pp. 733–742 (1997), http://doi.ieeecomputersociety.org/10.1109/34.598230.

[27] 細井辰弥, 栗田多喜夫, 名取研二：任意方向からの顔画像の認識のための多方向顔画像の主成分分析, 電子情報通信学会パターン認識・メディア理解研究会, Vol. PRMU2005-266, pp. 69–74 (2006).

[28] Iwamura, M., Negishi, K. and Aso, andHirotomo S. O.: Isolated Character Recognition by Searching Feature Points, *ICDAR2005*, pp. 1035–1039 (2005), http://www.m.cs.osakafu-u.ac.jp/publication_data/365.pdf.

付　　録

1. 位相推定の原理

ここでは、式 (21) と式 (22) で推定される位相 φ について考察する。

1.1 連続の場合

まず離散的な定式化から一旦離れて、連続的な定式化について見る。

まず式 (21) における $\boldsymbol{x}^{\mathrm{out}}$ を、要素の添字 i で表されるベクトルではなく、時間 t の関数

$$\boldsymbol{x}^{\mathrm{out}}(t) = K\sin(\omega t - \varphi) + D \tag{A·1}$$

として考える。すると、

$$S = \int_0^T \boldsymbol{x}^{\mathrm{out}}(t)\sin\omega t\ dt \tag{A·2}$$

ここで $T = \frac{2\pi}{\omega}$ である。これを式変形すると、

$$\begin{aligned}
S &= \int_0^T \boldsymbol{x}^{\mathrm{out}}(t)\sin\omega t\ dt && \text{(A·3)}\\
&= K\int_0^T \sin(\omega t-\varphi)\sin\omega t\ dt + D\int_0^T \sin\omega t\ dt && \text{(A·4)}\\
&= \frac{-K}{2}\int_0^T \cos(2\omega t-\varphi)\ dt + \frac{K}{2}\int_0^T \cos\varphi\ dt && \text{(A·5)}\\
&= \frac{-K}{2}\left[\frac{1}{2\omega}\sin(2\omega t-\varphi)\right]_0^T + \frac{K}{2}\cos\varphi\,[t]_0^T && \text{(A·6)}\\
&= \frac{-K}{4\omega}\{\sin(2\omega T-\varphi)\sin(-\varphi)\} + \frac{K}{2}T\cos\varphi && \text{(A·7)}\\
&= \frac{-K}{4\omega}\{\sin(4\pi-\varphi)\sin(-\varphi)\} + \frac{K\pi}{\omega}\cos\varphi && \text{(A·8)}\\
&= \frac{-K}{4\omega}\{\sin(-\varphi)\sin(-\varphi)\} + \frac{K\pi}{\omega}\cos\varphi && \text{(A·9)}\\
&= \frac{K\pi}{\omega}\cos\varphi && \text{(A·10)}
\end{aligned}$$

C についても同様に

$$\begin{aligned}
C &= \int_0^T \boldsymbol{x}^{\mathrm{out}}(t)\cos\omega t\ dt && \text{(A·11)}\\
&= K\int_0^T \sin(\omega t-\varphi)\cos\omega t\ dt \\
&\quad + D\int_0^T \cos\omega t\ dt && \text{(A·12)}\\
&= \frac{K}{2}\int_0^T \sin(2\omega t-\varphi)\ dt \\
&\quad + \frac{K}{2}\int_0^T \sin(-\varphi)t\ dt && \text{(A·13)}\\
&= \frac{K}{4\omega}[\cos(2\omega t-\varphi)] + \frac{K}{2}\sin\varphi\,[t]_0^T && \text{(A·14)}\\
&= \frac{K}{4\omega}\{\cos(4\pi-\varphi)-\cos(\varphi)\} + \frac{K\pi}{\omega}\sin\varphi && \text{(A·15)}\\
&= \frac{K}{4\omega}\{\cos(\varphi)-\cos(\varphi)\} + \frac{K\pi}{\omega}\sin\varphi && \text{(A·16)}\\
&= \frac{K\pi}{\omega}\sin\varphi && \text{(A·17)}
\end{aligned}$$

よって

$$\frac{C}{S} = \frac{\sin\varphi}{\cos\varphi} = \tan\varphi \tag{A·18}$$

したがって

$$\tan^{-1}\left(\frac{C}{S}\right) = \varphi \tag{A·19}$$

以上のように、位相 φ が計算できることが分かる。

1.2 複素フーリエ変換

S, C は実数なので、複素数 $C + jS$ を考える。ここで j は虚数単位である。いま、正弦波の周波数を ω_0 とすると、

$$\boldsymbol{x}^{\mathrm{out}}(t) = K\sin(\omega_0 t-\varphi) + D \tag{A·20}$$

である。前項の計算式を用いると、

$$\begin{aligned}
C + jS &= \int_0^T \boldsymbol{x}^{\mathrm{out}}(t)\cos\omega_0 t\ dt \\
&\quad + j\int_0^T \boldsymbol{x}^{\mathrm{out}}(t)\sin\omega_0 t\ dt && \text{(A·21)}\\
&= \int_0^T \boldsymbol{x}^{\mathrm{out}}(t)(\cos\omega_0 + j\sin\omega_0 t)\ dt && \text{(A·22)}\\
&= \int_0^T \boldsymbol{x}^{\mathrm{out}}(t)e^{-j\omega_0 t}\ dt && \text{(A·23)}
\end{aligned}$$

これは $\boldsymbol{x}^{\mathrm{out}}(t)$ のフーリエ変換に他ならない。ここで $x(t)$ のフーリエ変換を $\mathcal{F}[x(t)](\omega)$ とすると、

$$C + jS = \mathcal{F}[\boldsymbol{x}^{\mathrm{out}}(t)](\omega_0) \tag{A·24}$$

である。また $\boldsymbol{x}^{\mathrm{out}}$ は周波数を ω_0 しか持たないので、

$$\mathcal{F}[\boldsymbol{x}^{\mathrm{out}}(t)](\omega) = \begin{cases} C + jS, & \omega = \omega_0 \\ 0, & \omega \neq \omega_0 \end{cases} \tag{A·25}$$

となる。

以上より、位相 $\varphi = \angle(C + jS)$、つまり複素数 $C + jS$ の偏角で表されている。

1.3 離散フーリエ変換

複素フーリエ変換と同様に、離散フーリエ変換も以下のように考える。

$$S = \sum_{k=0}^{n^{\mathrm{out}}-1} \boldsymbol{x}^{\mathrm{out}}[k] \sin k\omega_0 \tag{A·26}$$

$$C = \sum_{k=0}^{n^{\mathrm{out}}-1} \boldsymbol{x}^{\mathrm{out}}[k] \cos k\omega_0 \tag{A·27}$$

より、

$$\begin{aligned} C + jS &= \sum_{k=0}^{n^{\mathrm{out}}-1} \boldsymbol{x}^{\mathrm{out}}[k] \cos k\omega_0 \\ &\quad + j \sum_{k=0}^{n^{\mathrm{out}}-1} \boldsymbol{x}^{\mathrm{out}}[k] \sin k\omega_0 \qquad \text{(A·28)} \\ &= \sum_{k=0}^{n^{\mathrm{out}}-1} \Big\{ \boldsymbol{x}^{\mathrm{out}}[k] \cos k\omega_0 \\ &\qquad + j\boldsymbol{x}^{\mathrm{out}}[k] \sin k\omega_0 \Big\} \qquad \text{(A·29)} \\ &= \sum_{k=0}^{n^{\mathrm{out}}-1} \boldsymbol{x}^{\mathrm{out}}[k](\cos k\omega_0 + j \sin k\omega_0) \qquad \text{(A·30)} \\ &= \sum_{k=0}^{n^{\mathrm{out}}-1} \boldsymbol{x}^{\mathrm{out}}[k] e^{-jk\omega_0} \qquad \text{(A·31)} \\ &= \sum_{k=0}^{n^{\mathrm{out}}-1} \boldsymbol{x}^{\mathrm{out}}[k] e^{-j\frac{2\pi k n_0}{n^{\mathrm{out}}}} \qquad \text{(A·32)} \end{aligned}$$

ここで $\omega_0 = 2\pi \frac{n_0}{n^{\mathrm{out}}}$ とおいた。また $N = n^{\mathrm{out}}$, $x_k = \boldsymbol{x}^{\mathrm{out}}[k]$ とおくと、

$$C + jS = \sum_{k=0}^{N-1} x_k e^{-j\frac{2\pi k n_0}{N}} \tag{A·33}$$

これは x_k の離散フーリエ変換に他ならない。

ここで $x[i]$ の離散フーリエ変換を $\mathcal{F}[x[i]](n)$ とすると、

$$C + jS = \mathcal{F}[\boldsymbol{x}^{\mathrm{out}}[i]](n_0) \tag{A·34}$$

である。また $\boldsymbol{x}^{\mathrm{out}}$ は周波数を ω_0 しか持たないので、

$$\mathcal{F}[\boldsymbol{x}^{\mathrm{out}}[i]](n) = \begin{cases} C + jS, & n = n_0 \\ 0, & n \neq n_0 \end{cases} \tag{A·35}$$

となる。

Independent Component Analysis for Obstacle Detection

Naoya OHNISHI† and Atsushi IMIYA††

† School of Science and Technology, Chiba University
Yayoicho 1-33, Inage-ku, Chiba, Japan
†† Institute of Media and Information Technology, Chiba University
Yayoicho 1-33, Inage-ku, Chiba, Japan

Abstract An planar area in the world is called a dominant plane, which corresponds to the largest part in an image. Dominant plane detection is an essential task for autonomous navigation of a mobile robot equipped with a vision system, since it is possible to assume that a mobile robot moves on the ground plane which corresponds to the dominant plane. In this paper, we develop an algorithm to detect the dominant plane from a sequence of images using optical flow and Independent Component Analysis. Optical flow is observed through an uncalibrated camera mounted on a mobile robot. We show that optical flow is a linear combination of flows of a dominant plane and obstacle areas. Therefore, ICA allows us to separate optical flow on the dominant plane and obstacle areas from the observed optical flow. We present some experiments for dominant plane detection using a synthetic image sequence and a real image sequence obtained by a mobile robot.

Key words Independent Component Analysis, Optical flow, Obstacle detection, Robot vision

1. Introduction

In this paper, we aim to develop an algorithm for obstacle detection by applying Independent Component Analysis [7] to optical flow observed by means of a vision sensor mounted on a mobile robot.

Independent Component Analysis(ICA) is a statistical technique for separating mixture signals. The mixture signals are observed as a linear combination of blind source signals. It is assumed that the source signals are statistical independent. Recently, ICA is used in the fields of image analysis, such as texture analysis [8] [11] [17], face recognition [2] [13], natural image analysis [3] [6], and medical image analysis [12] [18]. However, ICA is not so much used in time-varying image processing, such as optical flow computed from an image sequence.

Optical flow [1] [9] [14] is an apparent motion of the scene. Additionally, optical flow can be obtained as a fundamental feature in the scene [19] comparing with a gray or color value of an image. Therefore, the use of optical flow is an appropriate method from the viewpoint of the affinity between the robot and human being. Using optical flow, algorithms for obstacle detection for robot navigation are proposed [15] [5] [16]. Enkelmann [5] proposed an obstacle-detection method using the model vectors from motion parameters. Santos-Victor and Sandini [16] also proposed an obstacle-detection algorithm for a mobile robot using the inverse projection of optical flow to ground floor, assuming that the motion of the camera system mounted on a robot is pure translation with a uniform velocity. However, even if a camera is mounted on a wheel-driven robot, the vision sensor does not move with uniform velocity due to mechanical errors of the robot and unevenness of the floor.

Optical flow can be assumed to be the mixture pattern observed through a moving camera in an environment. Therefore, ICA is suitable for the separation of optical flow. Our algorithm detects obstacles and a ground plane from optical flow observed through a uncalibrated camera mounted on a mobile robot by using ICA. First, optical flow is computed from a pair of successive images observe through a camera mounted on a mobile robot without any obstacles. This optical flow is called planar flow. Next, optical flow in the environment where obstacles exist and the planar flow is used for input signals of ICA. Then, ICA outputs two optical flows as independent components. Our algorithm detects obstacles and ground plane from these output optical flows.

In Section 2, we briefly present ICA, and we show that separation of pixels in a scene applies ICA to flow vectors of pixels. Section 3 presents an algorithm for the detection of obstacles and a ground plane from optical flow. In Section

4, we show experiments for the detection of obstacles using a real image sequence observed by the camera mounted on a mobile robot. we show the validity of own method for a robot navigation.

2. Application of ICA to optical flow

ICA [7] is a statistical technique for the separation of original signals from mixture signals. Assume that the mixture signals $x_1(t)$ and $x_2(t)$ are expressed as a linear combination of the original signals $s_1(t)$ and $s_2(t)$, that is,

$$x_1(t) = a_{11}s_1(t) + a_{12}s_2(t), \quad (1)$$

$$x_2(t) = a_{21}s_1(t) + a_{22}s_2(t), \quad (2)$$

where a_{11}, a_{12}, a_{21}, and a_{22} are weight parameters of the linear combination. Using only the recorded signals $x_1(t)$ and $x_2(t)$ as an input, ICA can estimate the original signals $s_1(t)$ and $s_2(t)$ based on the statistical properties of these signals.

We apply ICA to optical flow observed by a camera mounted on a mobile robot for detection of the feasible region on which the robot can move. The optical-flow fields are suitable for the input signals to ICA, since optical flow observed by the moving camera is expressed as the linear combination of the motion field of the ground plane and the other objects, as shown in Fig1. Assuming that the motion field of the ground plane and the other objects are spatially independent components, ICA enables us to detect the ground plane on which robot can moves. For each image in a sequence, we assume that optical flow vectors in the ground plane corresponds to an independent component. as shown in Fig.2.

3. Algorithm for obstacle detection from image sequence

In this section, we develop an algorithm for the detection of obstacles from the image sequence observed by a camera mounted on a mobile robot. When the camera mounted on the mobile robot moves on the ground plane, we obtain successive images which include a ground plane area and obstacles. Assuming that the camera is mounted on a mobile robot, the camera moves parallel to the ground plane. Since the computed optical flow from the successive images describes the motion of the ground plane and obstacles on the basis of the camera displacement, the difference between these optical flow vectors enables us to detect the ground plane area. The difference of optical flow is shown in Fig.3.

3. 1 Learning supervisor signal

First, we capture image sequence $\hat{I}(x, y, t)$ at time t without obstacles as shown in Fig.4 and compute optical flow $\hat{\boldsymbol{u}}(t) = (\frac{dx}{dt}, \frac{dy}{dt})$ as

$$\hat{\boldsymbol{u}}(t)^\top \nabla \hat{I}(x, y, t) + \hat{I}_t = 0, \quad (3)$$

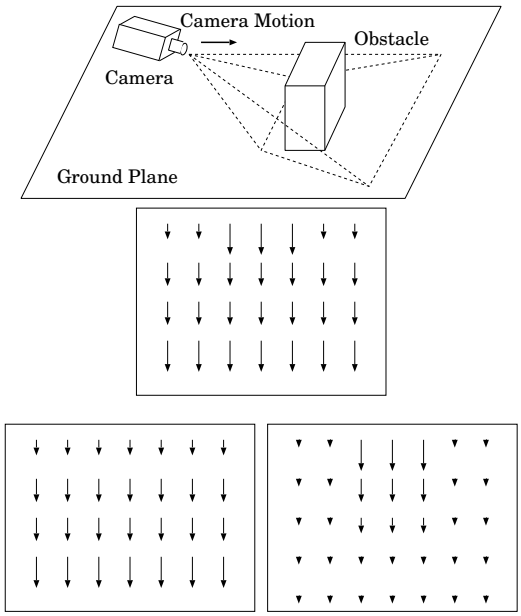


Fig. 1 Top-left: Example of camera displacement and the environment with obstacles. Top-right: Optical flow observed through the moving camera. Bottom-left: The motion field of the ground plane. Bottom-right: The motion field of the other objects. Optical flow(top-right) is expressed as the linear combination of bottom motion fields.

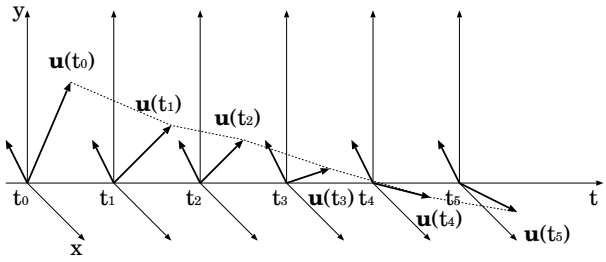


Fig. 2 Dominant vector detection in a sequence of images. $\boldsymbol{u}(t_i)$ corresponds to the dominant vector which defines the ground plane at time t_i.

where x and y are the pixel coordinates of a image. For the detail of the computation of this equation, see [1] [9] [14].

After we compute optical flow $\hat{\boldsymbol{u}}(t)$, frame $t = 0 \ldots n$, we create the supervisor signal $\hat{\boldsymbol{u}}$,

$$\hat{\boldsymbol{u}} = \frac{1}{n}\sum_{t=0}^{n} \hat{\boldsymbol{u}}(t). \quad (4)$$

3. 2 Ground plane detection using ICA

Next, we capture image sequence $I(x, y, t)$ with obstacles as shown in Fig.5 and compute optical flow $\boldsymbol{u}(t)$ in the same way.

Optical flow $\boldsymbol{u}(t)$ and the supervisor signal $\hat{\boldsymbol{u}}$ are used as an input signal for ICA. Setting $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ to be the output signals of ICA, $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are ambiguity of the order of the each components. We solve this problem using the difference between the variance of the length of $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$.

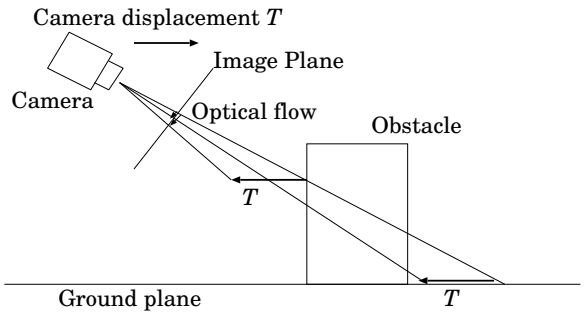


Fig. 3 The difference of optical flow between the ground plane and obstacles. If the camera moves in the distance T parallel to the ground plane, optical flow vector at the obstacles area in the image plane shows that the obstacle moves in the distance T, or optical flow vector at the ground plane area in the image plane shows that the ground plane moves in the distance T. Therefore, the camera observes difference optical flow vector between the ground plane and obstacles.

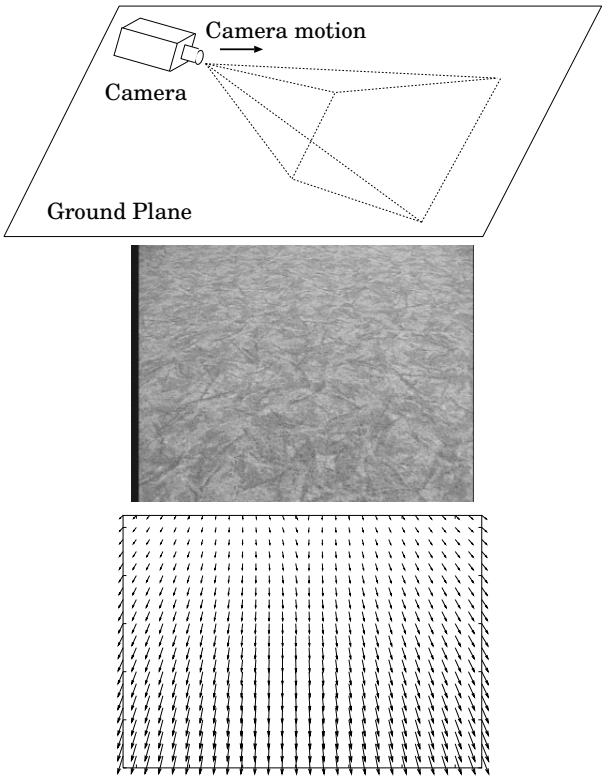


Fig. 4 Captured image sequence without obstacles. Top: Example of camera displacement and the environment without obstacles. Bottom-left: An image of the ground plane $\hat{I}(x, y, t)$. Bottom-right: Computed optical flow $\hat{\boldsymbol{u}}(t)$.

Setting $\boldsymbol{l}_1$ and $\boldsymbol{l}_2$ to be the length of $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$,

$$\boldsymbol{l}_j = \sqrt{\boldsymbol{v}_{xj}^2 + \boldsymbol{v}_{yj}^2}, \quad (j = 1, 2) \tag{5}$$

where $\boldsymbol{v}_{xj}$ and $\boldsymbol{v}_{yj}$ are arrays of x and y axis components of output $\boldsymbol{v}_j$, respectively, the variance σ_j^2 are

$$\sigma_j^2 = \frac{1}{xy}\sum_{i \in xy}(l_j(i) - \bar{l}_j)^2, \quad \bar{l}_j = \frac{1}{xy}\sum_{i \in xy} l_j(i), \tag{6}$$

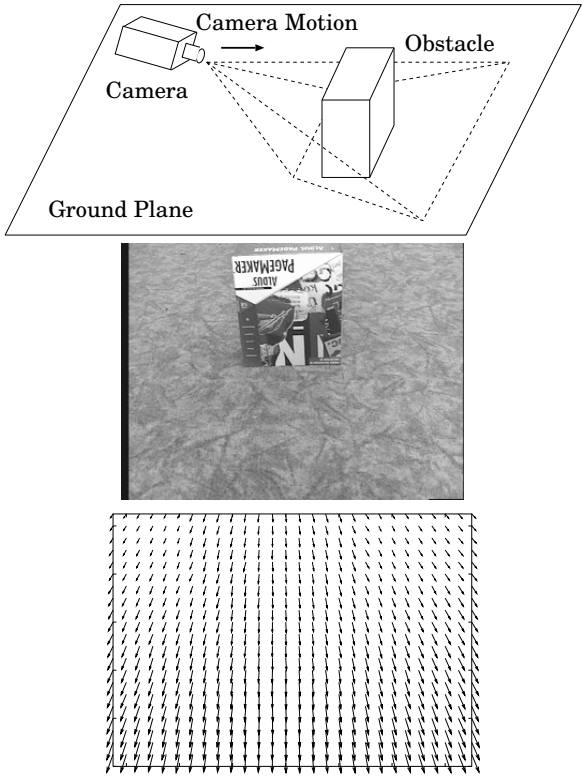


Fig. 5 Optical flow of the image sequence. Top: Example of camera displacement and the environment with obstacles. Bottom-left: An image of the ground plane and obstacles $I(x, y, t)$. Bottom-right: Computed optical flow $\boldsymbol{u}(t)$. In a top-middle area, where exists the obstacle, the lengths of optical flow vectors are longer than the flow vectors in the other area.

where $l_j(i)$ is the ith data of the array $\boldsymbol{l}_j$. Since the motions of the ground plane and obstacles in the image is different, the output which expresses the obstacle-motion has larger variance than the output which expresses the ground plane motion. Therefore, if $\sigma_1^2 > \sigma_2^2$, we detect ground plane using output signal $\boldsymbol{l}$ as $\boldsymbol{l} = \boldsymbol{l}_1$, else we use output signal $\boldsymbol{l} = \boldsymbol{l}_2$.

Since the ground plane occupies the largest domain in the image, we compute the distance between $\boldsymbol{l}$ and the median of $\boldsymbol{l}$. Setting m to be the median value of the elements in the vector $\boldsymbol{l}$, the distance $\boldsymbol{d} = (d(1), d(2), \ldots, d(xy))^\top$ is

$$d(i) = |l(i) - m|. \tag{7}$$

We detect the area on which $d(i) \approx 0$, as the ground plane.

3.3 Procedure for obstacle detection

Our algorithm is summarized as follows. Learning phase is,

(1) Robot moves on the ground plane in the small distance.

(2) Robot captures a image $\hat{I}(u, v, t)$ of ground plane.

(3) Compute optical flow $\hat{\boldsymbol{u}}(t)$ between the images $\hat{I}(u, v, t)$ and $\hat{I}(u, v, t-1)$.

(4) If time $t > n$, compute the supervisor signal $\hat{\boldsymbol{u}}$ using

Eq.(4). Else go to step 1.

Next, ground plane recognition phase is,

（1） Robot moves in the environment with obstacles in the small distance.

（2） Robot captures a image $I(u, v, t)$.

（3） Compute optical flow $\boldsymbol{u}(t)$ between the images $I(u, v, t)$ and $I(u, v, t-1)$.

（4） Input optical flow $\boldsymbol{u}(t)$ and the supervisor signal $\hat{\boldsymbol{u}}$ to ICA, and output the signal $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$.

（5） Detect the ground plane using the algorithm Section 3.2.

Figure 6 shows the procedure for ground plane detection using optical flow and ICA.

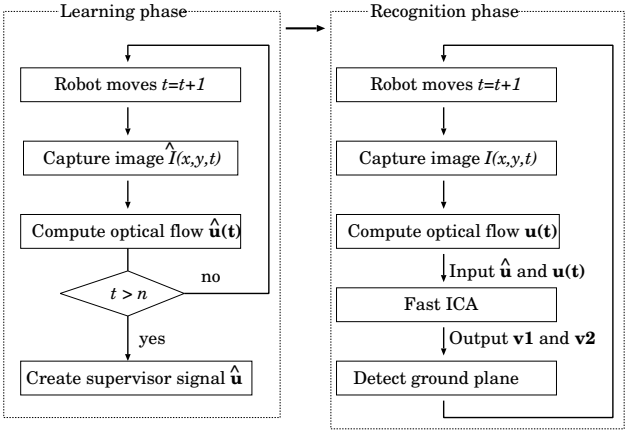


Fig. 6 Procedure for ground plane detection using optical flow and ICA.

4. Experiment

We show experiment for the ground plane detection using the procedure in Section 3.

First, the robot equipped with a single camera moves forward with uniform velocity on the ground plane and capture the image sequence without obstacles until $n = 20$. For the computation of optical flow, we use the Lucas-Kanade method with pyramids [4]. Using Eq.(4), we compute the supervisor signal $\hat{\boldsymbol{u}}$. Figure 7 shows the captured image and the computed supervisor signal $\hat{\boldsymbol{u}}$.

Next, the mobile robot moves on the ground plane toward the obstacle, as shown in Fig.8. The captured image sequence and computed optical flow $\boldsymbol{u}(t)$ is shown in the first and second rows in Fig.9, respectively. Optical flow $\boldsymbol{u}(t)$ and supervisor signal $\hat{\boldsymbol{u}}$ are used as an input signal for fast ICA. We use the *Fast ICA package for MATLAB* [10] for the computation of ICA. The result of ICA is shown in the third row in Fig.9.

Figures10-12 show other experiment results. In Fig.10, the mobile robot rotates in front of the obstacle. In Figs.11 and 12, we experiment using synthetic image sequence in a simulated environment for the translational and rotational motion.

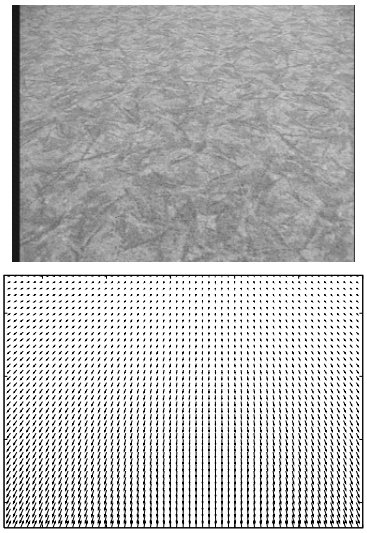


Fig. 7 Left: Image sequence $\hat{I}(x, y, t)$ of the ground plane. Right: Optical flow $\hat{\boldsymbol{u}}$ used for the supervisor signal.

Fig. 8 Experimental environment. The obstacle exists in front of the mobile robot. The mobile robot moves toward the obstacle.

5. Conclusion

We developed an algorithm for obstacle detection from a sequence of images observed through a moving uncalibrated camera. The use of the ICA for optical flow enables the robot to detect a feasible region in which robot can move without requiring camera calibration. These experimental results support the application of our method to the navigation and path planning of a mobile robot with a vision system. For each image in a sequence, the ground plane corresponds to an independent component. This relation provides us a statistical definition of the ground plane.

The future work is autonomous robot navigation using our algorithm of ground plane detection. If we project the ground plane of the image plane onto the ground plane using a camera configuration, the robot detects the movable region in front of the robot in an environment. Since we can ob-

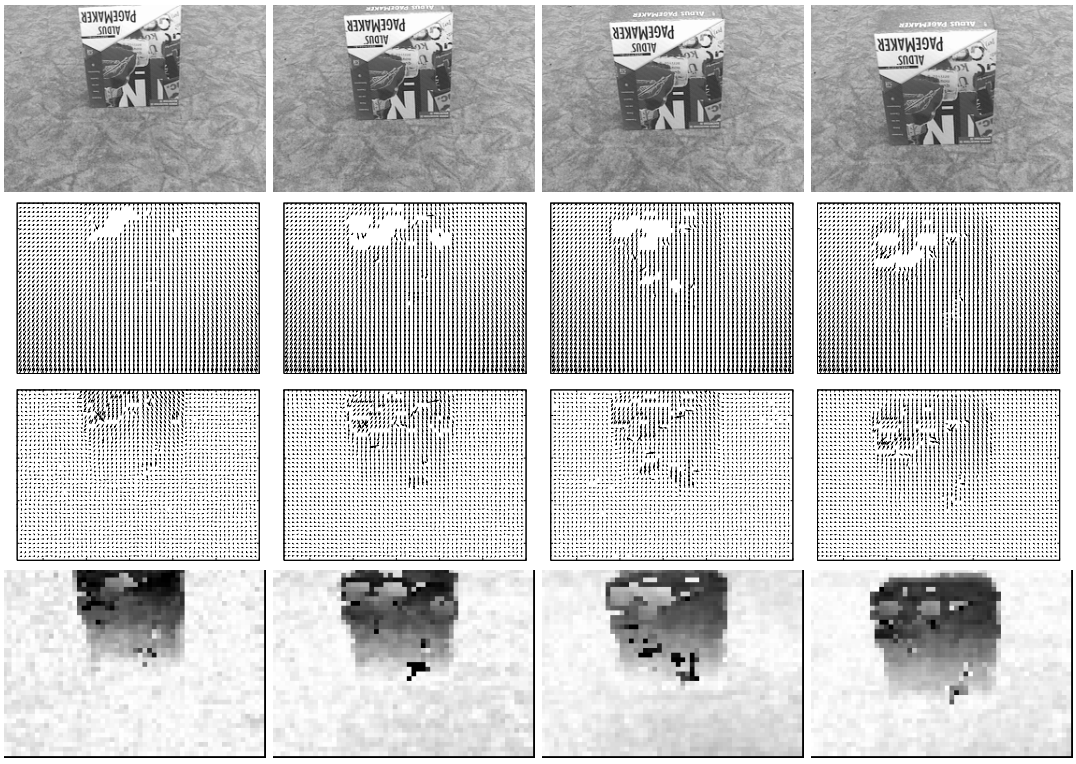


Fig. 9 The first, second, third, and forth rows show observed image $I(x, y, t)$, computed optical flow $\boldsymbol{u}(t)$, output signal $\boldsymbol{v}(t)$, and image of the ground plane $D(x, y, t)$, respectively. In the image of the ground plane, the white areas are the ground planes and the black areas are the obstacle areas. Starting from the left column, $t = 340$, 359, 374, and 393.

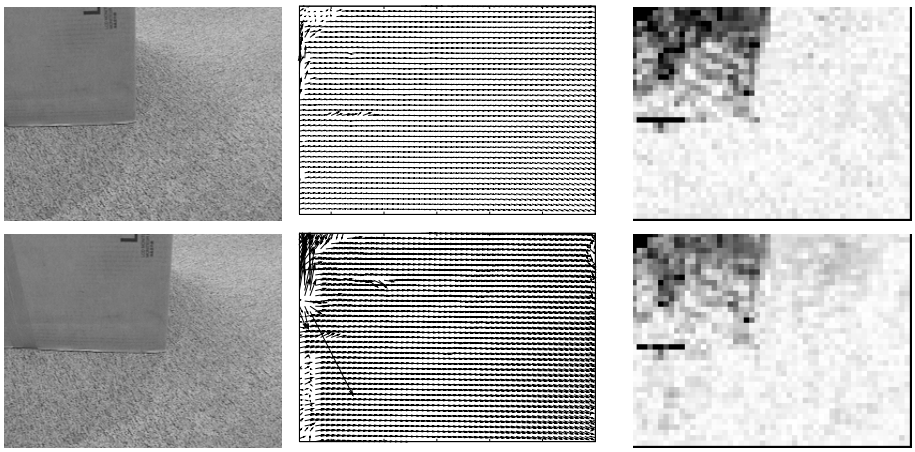


Fig. 10 Rotational motion. Image sequence(left column), computed optical flow(middle column), and detected obstacles(right column).

tain the sequence of the ground plane from optical flow, the robot can move the ground plane in a space without collision to obstacles.

References

[1] Barron, J.L., Fleet, D.J., and Beauchemin, S.S., Performance of optical flow techniques, International Journal of Computer Vision, vol.12, pp.43-77, 1994.

[2] Bartlett, M., Movellan, J., and Sejnowski, T., Face recognition by independent component analysis, IEEE Transactions on Neural Networks, vol.13, pp.1450-1464, 2002.

[3] Bell A and Sejnowski T, The 'Independent Components' of Natural Scenes are Edge Filters, Vision Research, vol.37, pp.3327-3338, 1997.

[4] Bouguet, J.-Y., Pyramidal implementation of the Lucas Kanade feature tracker description of the algorithm, Intel Corporation, Microprocessor Research Labs, OpenCV Documents, 1999.

[5] Enkelmann, W., Obstacle detection by evaluation of optical flow fields from image sequences, Image and Vision Computing, vol.9, pp.160-168, 1991.

[6] Hyvarinen, A, Hoyer, P. O., and Hurri, J., Extensions of ICA as models of natural images and visual processing, Proceedings of the Int. Symposium on Independent Component

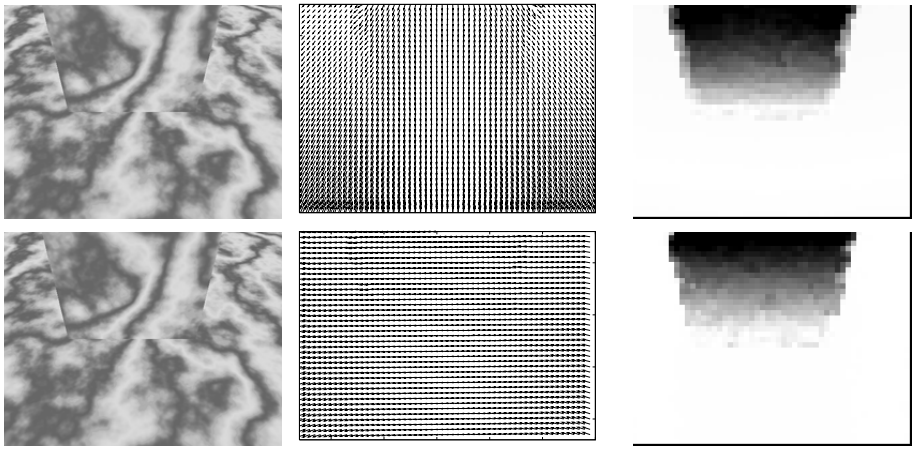


Fig. 11 Translational motion(first row). Rotational motion(second row). Image sequence(left column), computed optical flow(middle column), and detected obstacles(right column).

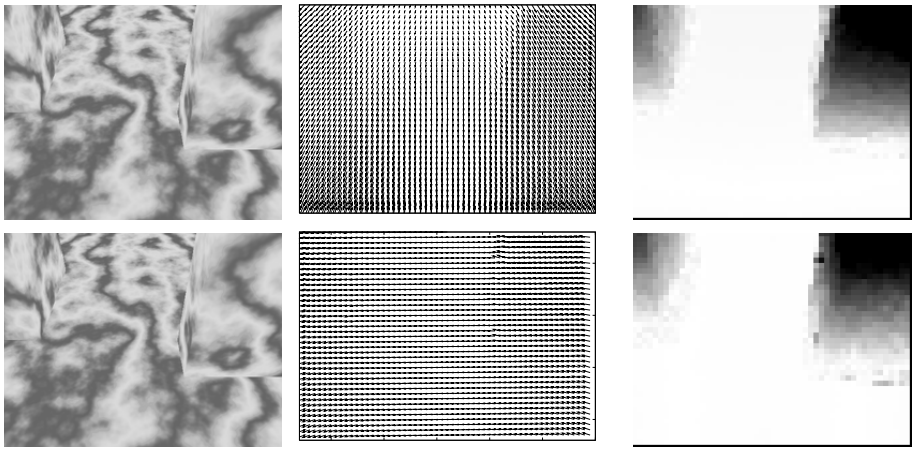


Fig. 12 Translational motion(first row). Rotational motion(second row). Image sequence(left column), computed optical flow(middle column), and detected obstacles(right column).

Analysis and Blind Source Separation, pp.963-974, 2003.

[7] Hyvarinen, A. and Oja, E., Independent component analysis: algorithms and application, Neural Networks, vol.13, pp.411-430, 2000.

[8] van Hateren, J., and van der Schaaf, A., Independent component filters of natural images compared with simple cells in primary visual cortex, Proceedings of the Royal Society of London, Series B, vol.265, pp.359-366, 1998.

[9] Horn, B. K. P. and Schunck, B.G., Determining optical flow, Artificial Intelligence, vol.17, pp.185-203, 1981.

[10] Hurri, J., Gavert, H., Sarela, J., and Hyvarinen, A., The FastICA package for MATLAB, available at http://www.cis.hut.fi/projects/ica/fastica/.

[11] Hyvarinen, A. and Hoyer, P., Topographic independent component analysis, Neural Computation, vol.13, pp.1525-1558, 2001.

[12] Inki, M., ICA features of image data in one, two and three dimensions, Proceedings of the Fourth International Symposium on Independent Component Analysis and Blind Signal Separation, pp.861-866, 2003.

[13] Liu, C., Enhanced independent component analysis and its application to content based face image retrieval, IEEE Transactions on Systems Man, and Cybernetics, Part B: Cybernetics. vol.34, no.2, pp.1117-27, 2004.

[14] Lucas, B. and Kanade, T., An iterative image registration technique with an application to stereo vision, Proceedings of 7th IJCAI, pp.674-679, 1981.

[15] Mallot, H. A., Bulthoff, H. H., Little, J. J., and Bohrer, S., Inverse perspective mapping simplifies optical flow computation and obstacle detection, Biological Cybernetics, vol.64, pp.177-185, 1991.

[16] Santos-Victor, J. and Sandini, G., Uncalibrated obstacle detection using normal flow, Machine Vision and Applications, vol.9, pp.130-137, 1996.

[17] Sezer, O.G., Ertuzun, A., and Ercil, A., Independent component analysis for texture defect detection, Pattern Recognition and Image Analysis, vol.14, pp.303-307, 2004.

[18] Stone, J., Porrill, J., Porter, N., and Wilkinson, I., Spatiotemporal Independent Component Analysis of Event-Related fMRI Data Using Skewed Probability Density Functions, NeuroImage, vol.15, pp.407-421, 2002.

[19] Vaina, L. M., Beardsley S. A. and Rushton S. K., *Optic flow and beyond*, Kluwer Academic Publishers, 2004.

部分空間法に基づく階層的自己位置推定

榎田 修一† 江島 俊朗†

†九州工業大学情報工学部 〒820-8502 福岡県飯塚市川津 680-4
E-mail: †{enokida, toshi}@ai.kyutech.ac.jp

あらまし 本論文で著者らは固有空間法、ならびに部分空間法を階層的に利用することで、画像情報のみから人工建造物の中を移動する自律型ロボットに必要な自己位置情報の推定を行っている． 全方位画像が得られた後、第1 層で進行方向を推定し，次の層では環境のトポロジカルマップを利用し高域な範囲での自己位置を大まかに推定する．さらに次の層においてより詳細な自己位置の推定を行うことで，自己位置を推定することが可能となる．

キーワード 自律型ロボット，自己位置推定，確率的オートマトン，トポロジカルマップ

Hierarchical Self-Localization Method based on Subspace Method

Shuichi ENOKIDA† Toshiaki EJIMA†

† Faculty of Computer Science and Engineering, Kyushu Institute of Technology

680-4 Kawadu, Izuka, Fukuoka, 820-8502 Japan
E-mail: †{enokida, toshi}@ai.kyutech.ac.jp

Abstract In this paper, we proposed the hierarchical self-localization method for an autonomous robot which has an omni-directional camera. Our proposal hierarchical architecture consists of two hierarchs. Upper one is designed to estimate a global position. In this hierarch, the accuracy of self-localization estimation is improved by using a probabilistic automaton based on a topological map. Lower one is designed to estimate a more detail position. In this hierarch, a probabilistic automaton is also applied to increase the capability of estimating a robot's position. The experiment was carried out with a real robot - *KITARO*.

Keyword Autonomous robot, Self-localization, Probabilistic Automaton, Topological Map

1. はじめに

近年，自律型移動ロボットに関する研究が盛んに行われ，今後もこの分野における発展は大いに期待されている．特に，日常の場で自律的に活動し，人に様々なサービスを提供するロボットの開発が期待され，活発に研究がなされている．著者等は自律型移動ロボットKITARO を作成し，人の身近な環境で動作し人とコミュニケーションをとることができるシステムの構築を目指している[1]．このような自律型移動ロボットが自律動作を行うにあたり，自己位置推定に基づく経路生成，自律移動の機能が必要である．

一般に自律型移動ロボットには距離センサや視覚センサが備えられ，それらの情報を利用し，組み合わせることで環境認識，環境地図生成，デッドレコニング等を行い，自己位置推定，及び自律走行を実現する[2-7]．文献[3]では全方位ステレオ視とレーザーレンジファインダから得られる時系列情報をセンサや環境の特徴を考慮し統合を行い，環境地図を構成する手法，ならびに自己移動量推定手法を提案している．しかし，各センサの特徴を表すモデルは設計者の経験によることが大きく，あらゆる環境で利用可能なモデルを構築することが困難である．そこで，環境が持つ多くの情報，外壁の色等を一度に把握できる視覚センサから得られる情報に基づく自己位置認識が注目されている[2]．本論文では，特に全方位視覚センサから得られる情報に着目した自己位置認識手法を提案する．視覚センサを用いた自己位置推定手法の一つに，環境内に一意に識別可能な物体（ランドマーク）を用いる手法がある[9]．ランドマークを用いる手法は自己位置の算出を簡便に，また精度良く行うことが可能である．しかし，幾つかのランドマークは常に観測可能でなければ

ならず，生活環境での利用を目的であるならば，人等の動的な障害物により隠されると自己位置推定が困難になること，ランドマークの設置等には多くのコストがかかることの問題が挙げられる．

そこで，実際に動作する環境で得られた記憶画像のみからの自己位置推定を実現する手法が注目されている[10-12]．これらの手法には，固有空間法を用いて画像情報を効率よく記憶し，照合を行うものが提案されている[10][11]．特に文献[12]では全方位画像のパノラマ展開した後、その自己相関画像を用いて固有空間を構成し自己位置推定を行うことで，全方位カメラの軸周りの回転に対し不変な情報を抽出し自己位置認識に用いている．しかし，自己相関画像は障害物の影響を受けやすく，また自己位置として推定すべきパラメータにロボットの姿勢を含む際は姿勢推定のために別の特徴を用いる必要がある．著者等も固有空間法を用いた段階的な自己位置推定[13]を提案している．提案手法では，ロボットが移動する環境全体を大域的環境と呼び，大域的環境を幾つかに分割し，それらの区間を局所領域と呼ぶ．自己位置を推定する際には，最初に姿勢推定のために構築された固有空間上で記憶画像と入力画像を比較し，姿勢の推定を行う．その後どの局所領域に存在するのかの推定を，各局所領域の記憶画像により構成された部分空間との距離に基づき行う．ここまでの処理を大域的自己位置推定と呼ぶ.その後，局所領域の自己位置変化により得られる画像群により構成された固有空間上で詳細な自己位置を推定する．また，動的物体の映り込みによる画像の部分的な変化に対応するため,全方位画像を幾つかの方向に分割し，それぞれ自己位置推定を行いそれらの結果を統合する．しかし，現時点での測定結果のみを用いて推定を行うため，時系列な情報や地図情報が付加されておらず，ノイズ的な画像が入力されることにより不安定な動作を示すことが確認されている．また，局所領域内の推定で学習画像を取得する際，画像 1 枚 1 枚の世界座標が既知である必要があり，画像取得に膨大なコストがかかる．

本論文では文献[13]の手法に基づき，確率オートマトン[14]を用いることで大域的環境においてそれぞれの局所領域間が持つ位相関係，及び時系列情報を反映した自己位置推定手法を提案する．また，その結果として得られる局所領域の確率を基に，候補領域内で詳細な自己位置推定を階層的に行う枠組を提案する．

2. 部分空間法、および固有空間法に基づく段階的自己位置推定手法［13］

本論文では，自己位置推定問題を自律型移動ロボット KITARO が動作することで環境に対して変動することが可能なパラメータを推定する．具体的には，動作環境をある 1 フロア(平面)とし，その環境内をいくつかのクラスに分類し，ロボットの姿勢推定，大域的自己位置推定(領域推定)をまず行い，次に局所的自己位置推定(位置推定)を行い詳細な自己位置を推定する．ロボットは位置情報である座標（x, y），その座標値を含む領域 ω，および姿勢 θ のパラメータを推定する．

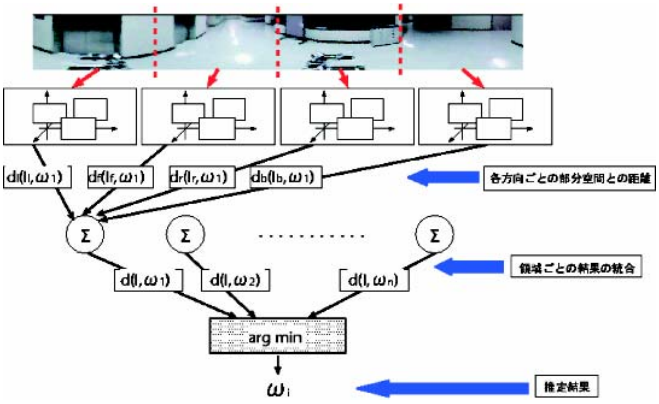


図 1 ：大域的領域推定の流れ

まず環境内でロボットを手動で操作しスタート地点からゴール地点までロボットを走行させる．その際にロボットに備え付けられた全方位視覚センサから画像を取得し，記憶する．この得られた全方位画像をパノラマ展開し，輝度画像に変換後，画像を縮小して Prewitt フィルタを用いてエッジ抽出を行ったものを特徴ベクトルとし，照明変動への対応を図る．記憶画像群に基づき，まずは姿勢推定の為の固有空間を構成する．また，局所領域毎に部分空間を構成する．自律移動時は，ロボットが観測した一枚の入力画像を特徴ベクトルへ変換し，固有空間法で次元圧縮された空間で任意の地点の各姿勢における記憶画像と距離を算出し，NN 法に基づき姿勢の推定を行う．次に，姿勢推定結果に基づきパノラマ展開画像をシフトし基準方向に合わせた画像に処理した後，前後左右 90° の画像へ 4 分割し，それぞれの方向毎に各局所領域の部分空間との距離を算出する．4 分割したことにより，現実世界でのノイズ（人物の進入等）がパノラマ画像のある箇所に局所的に発生しても頑健に位置推定を行える．

本手法では 4 方向の画像それぞれで部分空間との距離演算を行い，総和が最も小さな局所領域にロボットが存在すると推定する大域的自己位置推定を行い，その後，その部分空間内に存在する記憶画像のサンプルとの NN 法で最終的な自己位置推定の結果を返す局所的自己位置推定を行う．

3. 階層的自己位置推定手法

3.1. 階層的自己位置推定手法のながれ

ある時刻に得られた 1 枚の画像のみに基づく自己位置推定では，照明条件の突然の変化や記憶画像には存在しない物体の影響から，結果に誤差がでることが多い．このような問題に対し，時系列情報や各領域の位相情報を考慮することで解決を図る．

本論文で想定する廊下環境は，両側を壁に囲まれておりドアが一定間隔で存在する空間である.そのため，ある区間では類似性の高い空間が続くこととなり，そのなかでの詳細な位置推定は画像からでは困難である.しかし，どの区間を走行中か，その区間に入ったばかりか，それともその区間の走行を終えるのか等は画像から推定可能と考えられる．

まず始めに，設計者があらかじめ走行する廊下環境を全方位視覚センサから得られる画像の類似性に基づき局所領域に区切る．その後，各局所領域の位相関係を考慮したトポロジカルマップを構築する．最後に各局所領域での自己位置推定を行うために必要な情報に基づき取得画像を分ける．具体的には，直線の廊下であれば右端，中央，左端の 3 領域と，局所領域の移り変わる箇所付近(局所領域の両端)か中間かの 3 領域の推定を行う．ロボットはその組み合わせから考えられる 9 つの領域のどこにいるかの推定を行い，その結果を局所的領域推定とする．

姿勢と大域的領域の推定は文献[13]にて紹介する固有空間法，および部分空間法に基づく手法を用いる．この 2 階層の自己領域推定によって，大域的領域推定でロボットの姿勢，および現在ロボットが存在する局所領域の推定を行う．そこで過去の自己位置推定結果とトポロジカルマップから得られる情報に基づきロボットの存在する確率に重みをかける．その後，ロボットが存在する確率が高い局所領域のみに関して，局所的自己位置推定を行う．次節でトポロジカルマップに基づく確率的な大域的領域推定について述べる。

3.2. 階層的自己位置推定手法

文献[13]での自己位置推定における大域的領域推定は，各局所領域で獲得した画像群から構成される部分空間と入力画像との距離に基づき行われている．本論文では新たにトポロジカルマップに基づき生成される確率オートマトンに基づき自己位置推定を行う．確率オートマトンとは，ある局所領域 ω_i に対応する状態 i から j へ遷移する確率を要素として持つ行列 A で表現される．

$$A=\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix} \quad (1)$$

ここで n は状態数である．また，遷移確率 a は入力画像と各局所領域が持つ部分空間までの距離 D，およびトポロジカルマップから得られる空間的移送関係を反映した行列 M を用いて以下のように定義する．

$$a_{ij} = m_{ij}\exp(-\alpha \overline{D}_j) \quad (2)$$

$$\overline{D}_j = \frac{D_j - \min_k(D_k)}{\max_k(D_k) - \min_k(D_k)} \quad (3)$$

ここで α は定数である．具体的に図 2（上）に示すように環境を仮定すると，図 2（下）に示すトポロジカルマップが得られ，行列 M は以下に示される．

$$M=\begin{pmatrix} m_{11} & m_{21} & m_{31} & m_{41} & m_{51} \\ m_{12} & m_{22} & m_{32} & m_{42} & m_{52} \\ m_{13} & m_{23} & m_{33} & m_{43} & m_{53} \\ m_{14} & m_{24} & m_{34} & m_{44} & m_{54} \\ m_{15} & m_{25} & m_{35} & m_{45} & m_{55} \end{pmatrix} \quad (4)$$

$$=\begin{pmatrix} 1 & \varepsilon & 0 & 0 & 0 \\ \varepsilon & 1 & \varepsilon & 0 & 0 \\ 0 & \varepsilon & 1 & \varepsilon & 0 \\ 0 & 0 & \varepsilon & 1 & \varepsilon \\ 0 & 0 & 0 & \varepsilon & 1 \end{pmatrix} \quad (5)$$

ここで ε は $0<\varepsilon<1$ の定数であり接続した領域への遷移確率を反映した値である．結果，時刻 t におけるトポロジカルマップを考慮した大域的自己位置推定確率 P は，以下のように定義される．

$$P_{\omega_i}(0) = \frac{1}{n} \quad (6)$$

$$P_{\omega_i}(t) = \frac{P'_{\omega_i}(t)}{\sum_{j=1}^{n} P'_{\omega_j}(t)} \quad (7)$$

$$\begin{pmatrix} P'_{\omega_1}(t) \\ \vdots \\ P'_{\omega_n}(t) \end{pmatrix} = A \begin{pmatrix} P_{\omega_1}(t-1) \\ \vdots \\ P_{\omega_n}(t-1) \end{pmatrix} \quad (8)$$

ここで n は状態数である．また，確率 P の下限を

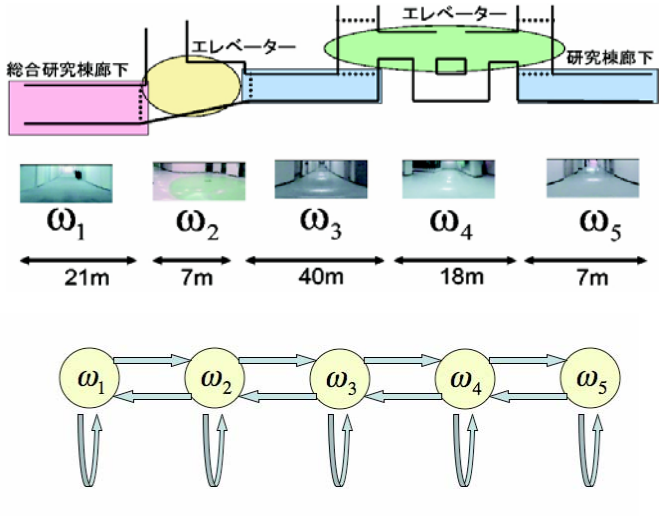


図２：実験環境（上）とトポロジカルマップ（下）

$$P'_{\omega_i}(t) > \eta \qquad (9)$$

とし，すべての状態へ遷移する確率を残す．また，

$$\max_k(P_{\omega_k}(t)) < th \qquad (10)$$

のときは照明の瞬間的な変動や同物体の接近等により記憶画像とのマッチングでは現在の状態が正確に推定できないと判断し，局所領域内での詳細な自己位置推定（局所的自己位置推定）を行わず，ロボットはその位置に停止し，一定時間画像を獲得し続ける等の処置を行うものとする．

3.3. 局所的自己位置推定手法

大域的自己位置推定の結果，時刻 t でロボットがある局所領域に存在している確率が高いと判断すると各局所領域内で大域的推定と同様に確率オートマトンによる局所的自己位置推定を行う．このとき局所領域内における状態の切り方は，得られる記憶画像の近似している領域内でも，画像に変動がおきる箇所（局所領域の切り替わる箇所等）毎に，設計者が分割する．具体的に図３（上）の環境で，状態 ω_1，ω_3，ω_5の廊下については図３で示すような９つの状態へ切り分け適用し，エレベータホールの ω_2，ω_4ではそれぞれ図４に示すように分割した．

また，ある局所領域において，別の局所領域に隣接する状態（図３中の７，８，９等）に存在する確率が高いときには，次時刻の大域的自己位置推定に用いる定数（本論分では ε）を k 倍する．これは，局所領域の遷移が発生する可能性が高いことをオートマトンの遷移確率に明確に反映させるためである．

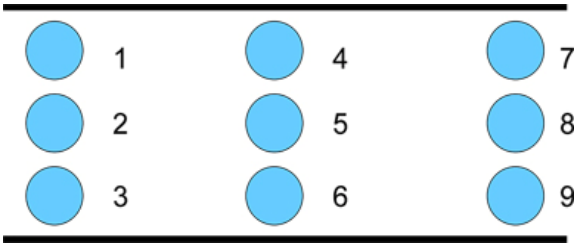


図３：廊下局所領域内状態（ω_1，ω_3，ω_5）

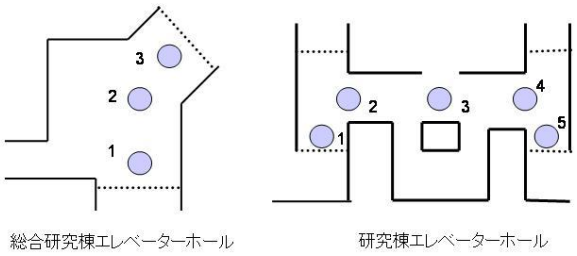


図４　：エレベータホール局所領域内状態（ω_2，ω_4）

4. 実験結果と考察

九州工業大学情報工学部の総合研究棟，及び研究棟の６階（図２）にて取得した画像データに基づき評価実験を行った．まず，大域的領域推定に用いる固有空間，ならびに部分空間構成の為に表１に示す枚数の記憶画像の取得を行った．姿勢については取得画像をパノラマ展開する開始点をずらすことで５°刻みの画像群を擬似的に生成したものを用いた．その後，別の日時に評価用の画像取得を行った．この時，実際にロボットを走行させ得られた画像から一定間隔で評価画像を取得している．局所領域に関しても同様に表２に示す．大域的領域推定における領域 ω_5に関しては領域 ω_3と同様の部分空間を用いた．また，今回の評価画像における大域的自己位置推定確率の遷移を図５に，また，領域 ω_1での局所領域推定確率を図６に示す．画像取得の際，ロボットは ω_1ー ω_5までを順番に通過し，ω_1 内では図３に示す２，５，８の順番で通過している．最後に，本実験で用いたパラメータを以下に示す．

$\alpha = 0.6$　$\varepsilon = 0.07$　$\eta = 0.05$　$th = 0.3$　$k = 4$

表１，及び表２の結果より，本提案手法により，従来手法と比較して精度の高い大域的自己位置推定が可能となったことが確認された．推定誤りと評価したほとんどは状態遷移が発生しているときであり，推定結果として特に問題ない箇所が大半である．また，提案手法により画像のみの推定では困難である領域 ω_3と

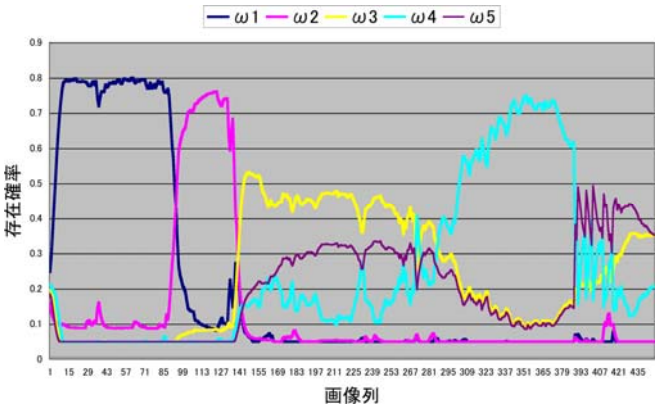


図５　：大域的自己位置推定時の確率変動

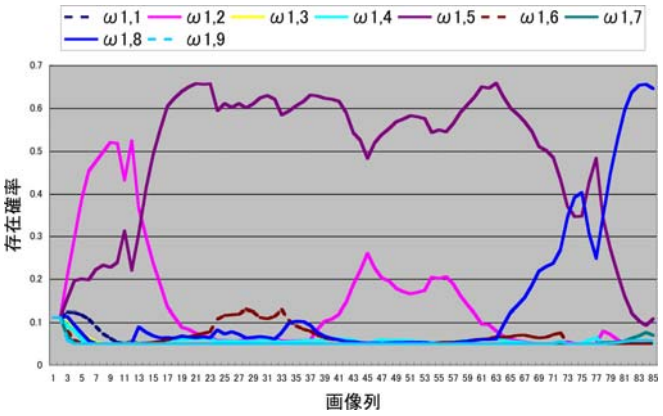


図６　：局所的自己位置推定時の確率変動

ω_5, つまり同じような画像が取得される領域での区別が可能であることが確認された．このことにより，非常に類似性の高い画像が多く入る建造物の中でも自己位置認識が実現できる可能性が示された．ただし，本実験ではスタート地点の廊下が画像から判断しやすいものであった為，状態推定結果の早期収束が確認されたと思われる．今後は類似性の高い空間が多い場所から自己位置認識を開始する等様々な実験を行い，より深い特性の解析を行う．

状態	ω_1	ω_2	ω_3	ω_4	ω_5
学習画像(枚)	1011	1056	1018	1066	—
評価画像(枚)	86	48	152	122	40
提案:正答(%)	100	100	100	100	97
従来:正答(%)	100	80	95	82	—

表１：大域的自己位置推定結果

状態	ω_1	ω_2	ω_3	ω_4	ω_5
学習画像(枚)	6977	1765	6647	1978	5407
評価画像(枚)	86	48	152	122	40
提案:正答(%)	85	100	86	88	73

表２：局所的自己位置推定結果

5. 実験結果と考察

本論文では全方位画像に基づき階層的に自己位置推定を行う手法を提案した．本手法では，まず文献[13]で示された固有空間法に基づく姿勢推定，及び部分空間法に基づく領域推定を行う．その後，前もって作成したトポロジカルマップに基づく確率オートマトンにより領域毎にロボットの存在確率を算出する手法である．結果，時系列的な部分空間との距離情報と環境のトポロジカルな関係を反映した自己領域推定を行うことが可能となった．今後の課題としては，現在，大域的領域，および局所的領域を手動で分割し部分空間を構成している点を，領域の自動的な分割へ変更することが挙げられる．

文　献

[1] Shuichi Enokida, Shuichi Kouno, Nobuyuki Kimura, Toshiaki Ejima, “KITARO: Kyutech Intelligent and Tender Autonomous Robot.”, Proc. of 12th IEEE Workshop on Robot and Human Interactive Communication,pp.341-346, 2003.

[2] Guilherme N. DeSouza, Avinash C. Kak, “Vision for Mobile Robot Navigation: A Survey”, *IEEE Transaction* Pattern Analysis and Machine Intelligence, Vol.24, No.2, pp.237-267, 2002.

[3] 根岸義朗，三浦純，白井良明，"全方位ステレオ視とレーザレンジファインダの統合による移動ロボットの地図生成"，日本ロボット学会誌，Vol.21，No.6，pp.690-696，2003.

[4] 唐立新，油田信一，"全方位画像列と移動量の記録による移動ロボットの教示再生ナビゲーション.'" 日本ロボット学会誌，Vol. 21，No.8，pp.883-892, 2003.

[5] S. Thrun, A. Bucken, W. Burgard, D. Fox, T. Frohlinghaus, D. Hennig, T. Hofmann, M. Krell, and T. Schimdt, "Map learning and high-speed navigation in RHINO", MIT/AAAI Press, Cambridge, MA, 1998.

[6] 友納正裕，"スキャンテンプレートマッチングによる移動ロボットの場所認識"，第 22 回日本ロボット学会学術講演大会予稿集，1B12，2004.

[7] 友納正裕，"大域スキャンマッチングと複数仮説追跡を用いたロバストな自己位置推定"，第 22 回日本ロボット学会学術講演大会予稿集，1B13，2004.

[8] 中村恭之，大原正満，小笠原司，石黒浩，"全方位ビジョンセンサを搭載した複数台の移動ロボットの自己位置同定法"，日本ロボット学会誌，Vol.21，No.1，pp.109-117，2003.

[9] 松本吉央，稲葉雅幸，井上博允，``ビューベーストアプローチに基づく移動ロボットナビゲーション"，日本ロボット学会誌，Vol.20，No.5，pp.506-514，2002.

[10] 城殿清澄，三浦純，白井良明，"誘導による移動経験に基づく視覚移動ロボットの自律走行"，日本ロボット学会誌，Vol. 19，No.6，pp.1003-1009，2001.

[11] 前田佐嘉志，久野義徳，白井良明，"固有空間解

析に基づく移動ロボット位置認識'', 電子情報通信学会誌文誌, Vol.J80-D-II, No.6, pp.1502-1511, 1997.

[12]岩佐英彦, 栗飯原述宏, 横矢直和, 竹村治雄, ''全方位画像を用いた記憶に基づく位置推定'', 電子情報通信学会論文誌, Vol.J84-D-II, No.2, pp.310-320, 2001.

[13]中野徹, 川口大介, 榎田修一, 江島俊朗, “全方位画像の固有空間に基づく段階的自己位置推定手法'', 第 10 回ロボティクス・シンポジア, 2B3, 2005.

[14]馬場則夫, 佐藤剣, ”ニューラルネットの学習アルゴリズムに関する一考察 -慣性項を用いた BP 法に対する階層構造確率オートマトンの適用-'' 計測自動制御学会論文誌, Vol.34, No.6, pp.621-628, 1998.

Network-based Noise Reduction for Microarray Data

Tsuyoshi KATO†, Kiyoshi ASAI†,††, Paul B. HORTON††, Koji TSUDA††, and Wataru FUJIBUCHI††

†† AIST Computational Biology Research Center, 2-43, Aomi, Koto-ku, Tokyo, 135–0064 Japan
† Graduate School of Frontier Sciences, The University of Tokyo, 5-1-5, Kashiwanoha, Kashiwa city, 277–8562, Japan

Abstract Prediction of human cell response to anti-cancer drugs (compounds) from microarray data is a challenging problem, due to the noise properties of microarrays as well as the high variance of living cell responses to drugs. Hence there is a strong need for more practical and robust methods than standard methods for real-value prediction. We devised an extended version of the off-subspace noise-reduction (de-noising) method to incorporate heterogeneous network data such as sequence similarity or protein protein interactions into a single framework. Experimental results show that prediction performance is improved by combining a prediction method with our de-noising method.

Key words microarray data, anti-cancer drugs, network-based de-noising, off-subspace de-noising method

1. Introduction

Cancer diagnosis based on gene expression data has been widely and extensively explored in the clinical research field since the earlier papers on gene expression arrays were published. Early studies mainly focused on the classification of cancer types, for example, discrimination of leukemia classes, a field in which powerful classifiers such as support vector machines are applied and the predictions are largely successful [6].

Recent cancer phenotype analysis is shifting from predicting a class to predicting a real-valued response. For example, predicting the effects of anti-cancer drugs(注1) is an important problem in cancer therapy, since a careful choice of proper not only drug but also dosage is required for different cancer cells and patients to maximize effectiveness and minimize deleterious side-effects. Although the drug response itself is a continuous quantity, this problem has often been simplified to the binary classification problem of drug sensitive vs. drug resistant [9], [12], [13]. Among the drug sensitivity classification studies, Staunton et al. [13] selected 232 out of 5,084 compounds or drugs to classify 60 cells(注2) by a sum of vote type classifier. According to their results, the rate of correct classification is significantly better than random classification.

In constrast to the simplified problem of classification, direct prediction of real-valued responses of a cancer drug from microarray data is not an easy task due to the noisy properties of both microarray technology and living cell experiments. Despite the limitation of available data, Mariadason et al [11] attempted to predict the cell apoptosis response against a chemotherapeutic agent (5-FU) by principal component regression (PCR). The leave-one-out test for 30 different cells in their analysis gives correlation coefficients of predicted and observed responses as low as 0.46. Gruvberger-Saal et al [7] also tried to predict the real-valued response of an estrogen receptor from gene expressions using artificial neural networks used in their earlier study.

In this paper, we focus on the noise and errors in microarray data that potentially degrade prediction performance. De-noising is similar to missing value estimation. Both infer the true values. Typical methods for missing value imputation (e.g. [4], [15]) capture the important dimensions by principal component analysis (PCA). However, they do not exploit the *side information* about genes, such as sequence similarity, GO classification, or protein-protein interactions, though those heterogeneous data sources are expected to be useful for identifying related genes, and furthermore to effectively correct noisy data.

We devise a new de-noising method using the side information represented as a *network*, where the nodes correspond to

(注1)：In this paper, compounds are referred to as drugs.
(注2)：To be exact, the term 'cell' should be called a cell line.

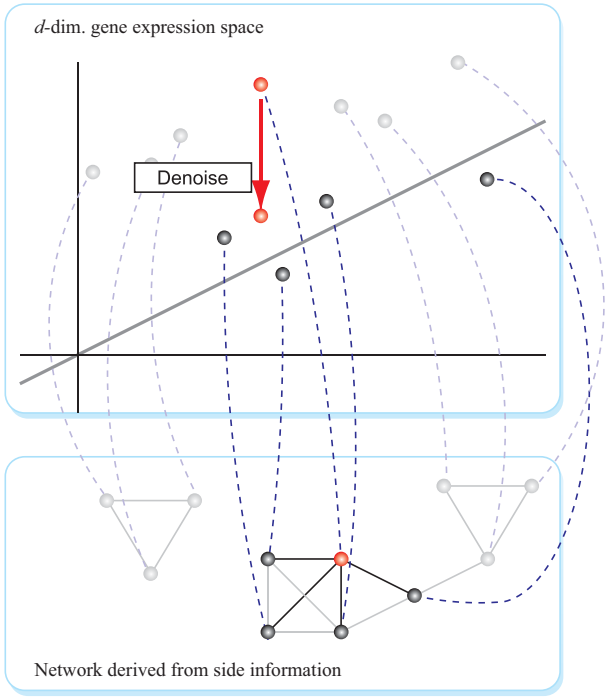


Fig. 1 De-noising based on a network. In the top figure, the target expression vector to be de-noised is depicted as a red point. Black points are the neighbors in the network (below) derived from side information; gray points are vectors which are not directly connected (i.e. related) to the target vector in the network. The edge of the network is depicted by a solid line. A dashed curve indicates the correspondence between data in a network and the expression vector. De-noising is done by robust projection onto the principal subspace made from only the neighbors. In this case, the subspace (gray line) is obtained by PCA of the target and the four neighbors.

the genes, and the edges represent relations among the genes. In de-noising the expression data of a certain gene, we only look at its neighborhood genes in the network. A principal subspace is made only from the neighborhood expression data, and the target vector is de-noised by robust projection (Figure 1). Here, we use a so-called "off-subspace" projection method [16] to prevent over-de-noising. This projection algorithm is formulated as a linear program, which can be efficiently solved even for a large number of neighborhood genes. The basic idea of our method is similar to local PCA approaches [14], where de-noising is done by projection to local subspaces. However, the novelty of our method is that the neighborhood relation is determined by the network.

Typically we have multiple data sources as the side information, which are represented as *multiple networks*. When a principal subspace is derived from each network, we have a set of subspaces for each expression vector. A simple way is to take the sum of all subspaces, and project each target expression vector onto the combined subspace. However, since some of the networks might not be useful for de-noising, it is preferable to select important subspaces automatically, and then take the sum of those subspaces. To this end, we extend the off-subspace projection method to deal with multiple networks. Network selection is implemented by giving a non-negative weight parameter to each network, optimizing the weight vector, and removing the networks with zero weights. The de-noising problem for multiple networks is also formulated as a linear program.

In predicting drug responses, it is often the case that the responses for many drugs are predicted from a microarray gene expression dataset. In this case, our problem is to learn a vector-to-vector mapping, where the output vector is composed of drug responses. Since the output vector provided for training is also noisy, our network-based de-noising method can also be applied to the output vector. As we have no side information about drugs, the correlation coefficients among the output vectors are used to construct a network.

In numerical prediction experiments using the drug response data by Staunton et al. [13], our method with multiple networks outperformed standard prediction method, PCR, significantly. Note that the output de-noising was also effective to enhance the accuracy of prediction. The improvement of correlation between true responses and predictions was observed for 930 drugs out of 1,427 drugs. The number (930/1,427=0.65) is statistically significant ($p < 10^{-2}$) in a cumulative binomial distribution model under the null hypothesis that half of the drugs (714) are chosen by chance.

2. De-Noising with a Single Network

The dataset we analyzed contains one microarray hybridization experiment [1] for each cell sample. Let d denote the number of hybridizations and N denote the number genes used in the analysis. We consider the d dimensional space of hybridizations populated by N points representing individual genes. In our drug response case, which has expressions of 60 cell samples for each gene, $d = 60$. Unfortunately in our application, the N points are generally quite noisy. Thus our goal is to "correct" the N points in a way which effectively reduces noise while maintaining signal.

2.1 Derivation of Subspaces

Let us define $\boldsymbol{x}_i$ as the d-dimensional gene expression vector of the i-th gene. Our task is to de-noise $\boldsymbol{x}_i$ using other vectors and a network which is represented by the $N \times N$ symmetric matrix W. The (i, j) element w_{ij} represents the strength of the edge between two nodes i and j. If there is no edge, $w_{ij} = 0$. The principal subspace for the i-th gene is computed using the neighborhood nodes only, i.e., the nodes with $w_{ij} \neq 0$. The basis vectors of the subspace are obtained as the principal eigenvectors $\boldsymbol{z}_{is}, s = 1, \ldots, n_i$, of the

following covariance matrix,

$$S_i = \frac{\sum_{j=1}^{N} w_{ij} \boldsymbol{x}_j \boldsymbol{x}_j^\top}{\sum_{j=1}^{N} w_{ij}} \quad (1)$$

The weighting covariance matrix represents the distribution of neighbors, and thereby yields the subspace by taking major eigenvectors as the basis vectors. We determine the number of basis vectors, n_i according to the Kaiser-Guttman rule [8]: the number of basis vectors is set as the number of eigenvalues greater than one in the normalized covariance matrix $\tilde{S}_i$ in which $[\tilde{S}_i]_{kl} = [S_i]_{kl} / \sqrt{[S_i]_{kk}[S_i]_{ll}}$.

2.2 Off-subspace Projection

Most simply, de-noising is done by projecting $\boldsymbol{x}_i$ to the subspace in terms of the least squares error. However, when the number of neighborhood nodes is small, or the non-zero weights w_{ij} are concentrated in only a few neighbors, the dimensionality of the local subspace can be too small. In that case, simple projection may result in an unacceptably large loss of signal, called *over-de-noising*.

Tsuda and Rätsch [16] addressed this by devising an off-subspace projection method (Figure 2) which corrects the data to a point generally closer to, but not necessarily in the subspace. Let $\bar{\boldsymbol{x}}$ denote the de-noised result of $\boldsymbol{x}_i$, which is obtained by solving the following optimization problem,

$$\min_{\bar{\boldsymbol{x}}, v_s} d\|\bar{\boldsymbol{x}} - \sum_{s=1}^{n_i} v_s \boldsymbol{z}_{is}\|_\infty + \beta_0 \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}\|_1, \quad (2)$$

where β_0 is a non-negative constant parameter. Multiplying the first term by d is a convenient normalization which reduces the dependence of the numerical value of β_0 on d. With a suitable transformation in Eq. 2 can be efficiently minimized with standard linear programming.

As can be seen, the objective function in Eq. 2 is the sum of two distances; the distance between the subspace and the off-subspace point $\bar{\boldsymbol{x}}$, and the distance between $\bar{\boldsymbol{x}}$ and the input vector $\boldsymbol{x}_i$. The two distances are measured with different norms; the former with ℓ_∞, the later with ℓ_1.

If $\beta_0 \to \infty$ and if ℓ_∞-norm is replaced with ℓ_2-norm, this becomes a least square method and the resulting off-subspace solution $\bar{\boldsymbol{x}}$ is equal to $\boldsymbol{x}_i$. As β_0 decreases, $\bar{\boldsymbol{x}}$ becomes close to the subspace.

Now let us describe the ℓ_1-norm and ℓ_∞ norm. Since the ℓ_1-norm regularizer yields a sparse vector as the optimal solution, this algorithm almost corrects only the contaminated elements in $\boldsymbol{x}_i$ without change of the non-contaminated elements. The ℓ_∞ norm has similar behavior to the ℓ_2-norm. Indeed, experiments in [16] show that the ℓ_∞ and ℓ_2 achieve similar de-noising performance. If ℓ_2 is used, the optimization problem can be transformed into a quadratic programming problem. If ℓ_∞ is used, the problem is a linear program, which can be solved more quickly and more stably than quadratic programs.

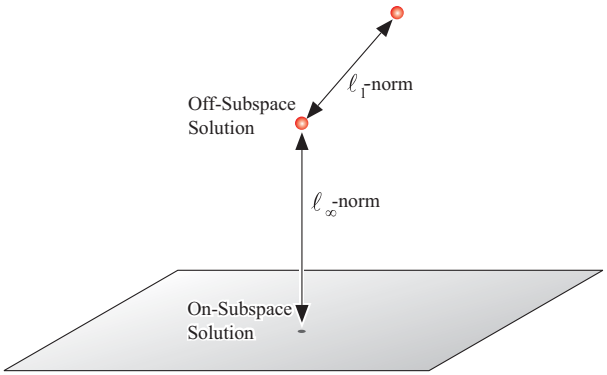


Fig. 2 Off-subspace de-noising method. This figure illustrates how to de-noise an expression vector using a principal subspace. Instead of simple projection, we simultaneously find two points (off-subspace solution and on-subspace solution) which minimize the sum of ℓ_1-norm distance and ℓ_∞-norm distance given in Eq. 2.

3. De-Noising with Multiple Networks

We further introduce a way to incorporate heterogeneous data sources into the noise reduction process by weighting the multiple networks W_k, $k = 1, \ldots, m$. The advantage of this method is its ability to incorporate various kinds of biological knowledge into a single framework.

Denote the s-th basis vector obtained from the k-th network by $\boldsymbol{z}_{is}^{(k)}$. To put it more precisely, each network yields the weighted covariance matrix $S_i^{(k)}$ and we take $n_i^{(k)}$ basis vectors $\boldsymbol{z}_{is}^{(k)}, (s = 1, \cdots, n_i^{(k)})$ from $S_i^{(k)}$. To use all the subspaces gained from the networks, one can take the sum of the subspaces and apply the off-subspace projection method. Then, the optimization problem can be described as

$$\min_{\bar{\boldsymbol{x}}, v_{is}^{(k)}} d\|\bar{\boldsymbol{x}} - \sum_{k=1}^{m} \sum_{s=1}^{n_i^{(k)}} v_{is}^{(k)} \boldsymbol{z}_{is}^{(k)}\|_\infty + \beta_0 \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}\|_1. \quad (3)$$

Note that any point in the sum of the subspaces can be represented as $\sum_{k=1}^{m} \sum_{s=1}^{n_i^{(k)}} v_{is}^{(k)} \boldsymbol{z}_{is}^{(k)}$. To automatically select important subspaces, we need to introduce a regularization term to the above optimization problem so that all the coefficients $v_{is}^{(k)}$ of unnecessary networks degenerate to zero.

For that, we introduce the upper bound of the absolute values of the coefficients of the k-th subspace as

$$t_k = \max_{1 \leqq s \leqq n_i} |v_{is}^{(k)}| = \| \left[v_{i,1}^{(k)}, \quad \cdots, \quad v_{i,n_i}^{(k)} \right] \|_\infty,$$

and penalize the ℓ_1-norm of the vector of upper bounds $\boldsymbol{t} = (t_1, \cdots, t_m)^\top$ as follows,

$$\min_{\bar{\boldsymbol{x}}, v_{is}^{(k)}} d\|\bar{\boldsymbol{x}} - \sum_{k=1}^{m} \sum_{s=1}^{n_i} v_{is}^{(k)} \boldsymbol{z}_{is}^{(k)}\|_\infty + \beta_0 \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}\|_1 + \beta_1 \|\boldsymbol{t}\|_1 \quad (4)$$

Due to the regularizer, some elements of $\boldsymbol{t}$ are exactly zero at

the optimal solution Moreover one can control the number of nonzero elements with the constant parameter β_1. If $t_k = 0$, all the coefficients of the k-th network are zero, implying that that network is not used at all in deriving the de-noised result $\bar{\boldsymbol{x}}$. This optimization problem can also be transformed into a linear program which can be solved efficiently (details not shown).

4. Experimental Settings

In the following experiments, our task is to predict the drug resistance levels of cells based on their gene expression data. By combining our novel de-noising methods and a standard prediction method, we wish to predict the resistance levels of the *test cells* accurately by learning from the input-output relations of the *training cells*.

The drug resistance dataset by Staunton et al. [13] contains the expression level of 6,817 genes from 60 human cancer cells. Among them, we pre-selected 2,067 highly variant genes (details not shown). For each cell, the drug resistance levels for 5,084 drugs are available as well. Those resistance levels are measured on a continuous scale by growth inhibition score (GI50). Prediction would be too easy, when the drug resistance levels are almost constant among cells. So we chose 1,427 drugs whose gap between the maximum and minimum levels was more than 0.5 after log-normalization.

We applied our network-based de-noising method in two ways. First, the expression profiles, i.e., the input of prediction, are de-noised with various networks. In the following experiments, we built three networks based on the correlation coefficients of the profiles, the gene ontology (GO), and the protein-protein interactions. Here, the vector being the de-noised $\boldsymbol{x}_i$ is the 60-dimensional expression vector of the i-th gene. Second, we also de-noised the vector of drug resistance levels, i.e., the output of prediction. Here, the vector $\boldsymbol{x}_i$ is composed of resistance levels of training cells for the i-th drug, and apply Eq. 4 to de-noise $\boldsymbol{x}_i$. In this case, we used only one network based on the correlation coefficients of the resistance level vectors. Namely, $m = 1$.

A schematic representation of the entire process is shown in Figure 3.

For predicting drug responses from the de-noised expression data, we tested a standard prediction algorithm, principal component regression (PCR). PCR is also used in Mariadason et al. [11]. For training cells, the Pearson correlation between the de-noised expression data of each of the 3,725 genes and (de-noised) responses of a drug of interest were calculated, and the 50 highest absolute value correlations (i.e., corresponding to 50 genes) were selected. To reduce the number of genes to a smaller set of variables, PCA was performed. From the PCA, the principal components (PCs) having the d_{pca} largest eigenvalues were selected. Next we get the parameters, $\boldsymbol{a} \in \Re^{d_{\mathrm{pca}}}$ and $b \in \Re$, of a linear regression function $g(\boldsymbol{z}|\boldsymbol{a}, b) = \boldsymbol{a}^\top \boldsymbol{z} + b$ by least square manner, where $\boldsymbol{z} \in \Re^{d_{\mathrm{pca}}}$ is the d_{pca} PCs of a cell. Namely, we find the values of $\{\boldsymbol{a}, b\}$ which minimize the sum-of-square errors: $\sum_i \left(g(\boldsymbol{z}_i|\boldsymbol{a}, b) - y_i\right)^2$ where $z_i \in \Re^{d_{\mathrm{pca}}}$ and y_i are the PCs and the drug response of i-th training cell, respectively. Once the regression function $g(\boldsymbol{z}|\boldsymbol{a}, b)$ was derived, the d_{pca} PCs corresponding to the test cell were computed and substituted into the derived regression function to yield a prediction of response of the test cell.

4.1 Building Networks

Here we describe the details of network construction, namely, the computation of matrix W. The first four networks are used for de-noising the input (i.e., expression data), and the last one is for the output (i.e., drug resistance levels).

a) Expression correlation

Presumably, as seen in missing value estimation, the most informative relations between genes for noise reduction can be obtained from co-expression. Thus, we use the Pearson correlation coefficient of expression as one of the heterogeneous data sources. The correlation is converted to probability under the hypothesis of $H0 : r = 0$ using one sample t-test, and the probability value is assigned to the weight w_{ij}. Notice that the expression data are not only the input of prediction, but also used for de-noising themselves.

b) Sequence similarity

To build a network based on the sequence similarity, the RNA sequences corresponding to the genes were extracted via GenBank accession numbers [3] described in the annotation fields of the gene expression data. The sequence similarity was computed by FASTA. Only the highest e-value between two sequences was used when there were multiple local alignment candidates. If the e-value was more than 10^{-5}, w_{ij} was set to zero, otherwise w_{ij} was set to the negative logarithm of the e-value. Among the 2,046 highly variant genes, only 550 were found to have edges to other genes.

c) Gene ontology

Gene ontology data were downloaded from the GO annotation project (GOA) [5]. The GenBank accessions [3] were translated to protein IDs and checked for any GO-relationships for gene pairs at the protein level. The edge strength w_{ij} was determined as the number of GO categories into which both proteins of a pair are classified. We obtained a total of 141,402 GO relations for 1,371 highly variant genes.

d) Protein-protein interaction

We obtained the protein-protein interaction data from the Biomolecular Interaction Network Database (BIND) [2]. This network has binary edges, i.e., w_{ij} is 0 or 1.

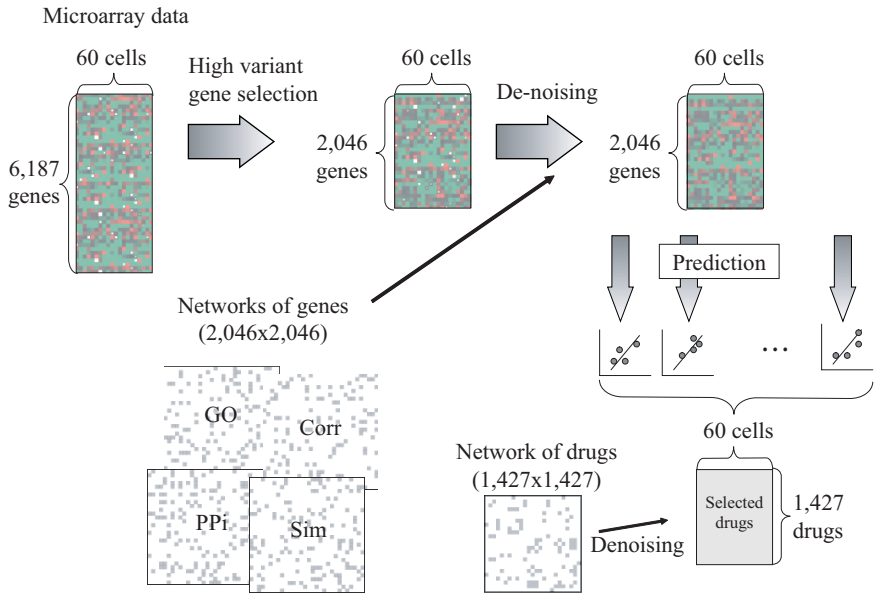


Fig. 3 Method overview

e) Drug response correlation

For de-noising drug responses, we calculated the p-values of the correlation coefficients of compound pairs with respect to their drug response data to generate the matrix W.

4.2 Parameter Selection

Our de-noising method has the two parameters, β_0 and β_1. In addition, PCR has one constant parameter, d_{pca}. For this purpose, we performed a joint grid search over the following values: $\beta_0 = 0.1, 0.2, 0.5, 1.0, 2.0, 3.0, 4.0$, $\beta_1 = 0.00, 0.02, 0.05, 0.1, 0.2, 0.5$, $d_{\mathrm{pca}} = 1, 2, 3, 4, 5, 7, 10, 20, 30, 40, 50$, and chose the parameter values yielding the best regression performance.

5. Results

Prediction performances of PCR with and without off-subspace noise reductions for different combinations of networks are shown in Figure 4. The accuracy of prediction is measured by the mean correlation coefficients in 12-fold cross validation. The leftmost bar 'None' corresponds to the performance without any de-noising. As anticipated, integration of both input and output de-noising yields the highest mean correlation coefficient ('All&Drug': 0.439). Among the other cases where only the input is de-noised, we obtained the best result when all networks are combined with weights ('All': 0.428). The weightless combination ($w_{ij} = 1$ for$\forall i, j$) was significantly poorer ('Unweighted'). In the case of de-noising only the input, noise reduction without using side-information degrades the prediction performance, but the use of side-information improves prediction. We also counted the drugs achieving statistically-significant predictions (Table 1).

Further experimental results are detailed in our journal paper [10].

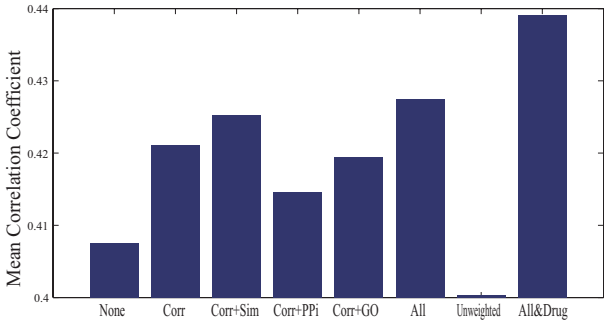


Fig. 4 Improvement of prediction by noise reduction with various combinations of networks. The mean correlation coefficients of prediction for 1,427 drugs before and after the off-subspace noise reduction are shown. Abbreviations are Corr: correlation coefficient for gene expressions; Sim: sequence similarity; GO: gene ontology; PPi: protein-protein interaction; All: Corr+Similarity+GO+PPi; Corr&Drug: input de-noising only via Corr and output de-noising; and All&Drug: input de-noising using All and output de-noising.

6. Concluding Remarks

The prediction of drug response data is critical for the field of cancer therapeutics, which demands improved diagnostics for determining the appropriate choice and dosage of anti-cancer drugs. Combining gene relations from various biological resources to adjust values of gene expressions or drug response data is a new approach in this field. This approach requires effective methods, such as the one presented here, for utilizing heterogeneous data.

This algorithm is invariant if the network weights are multiplied by a constant, as shown in Eq. 1. However, the change of the ratio among weights may have an influence

Table 1 The number of drugs given statistically-significant prediction of the responses. Here we define a drug that achieves the correlation coefficient more than 0.33 as a successfully predicted, which is derived from one sample t-test with the probability less than 0.01 examining the null hypothesis of "no correlation."

	None	Corr	Corr+Sim	Corr+GO	Corr+PPi	All	Unweighted	Corr&Drug	All&Drug
PCR	983	1,027	1,043	1,018	995	1,037	988	1,065	1,085

on the de-noising performance. Although various weighting schemes could be considered, we did not systematically investigate that issue in this work. However we did confirm that off-subspace noise reduction with the continuous weights defined above for sequence similarity, expression correlation, and GO heterogeneous data sources was more effective than using a 0-1 weighting scheme based on some threshold. The current weighting scheme might not yet be optimal and tuning would yield improvement to some extent.

We extended the off-subspace noise reduction method of Tsuda and Rätsch [16] and applied it to the noise reduction of gene expression data in the context of real-value prediction to drug response data. Our results show the method to be robust to noisy data and more effective than the traditional principal component regression, improving the prediction of 868 out of 1,427 drugs. We expect it will prove generally useful for correcting the values of noisy microarray data.

References

[1] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. *Molecular Biology of the Cell. 4th ed.*, chapter Microarrays Monitor the Expression of Thousands of Genes at Once, III. Methods / 8. Manipulating Proteins, DNA, and RNA / Studying Gene Expression and Function. Garland Publishing, New York, 2002.

[2] C. Alfarano, C. E. Andrade, K. Anthony, N. Bahroos, M. Bajec, K. Bantoft, D. Betel, B. Bobechko, K. Boutilier, E. Burgess, K. Buzadzija, R. Cavero, C. D'Abreo, I. Donaldson, D. Dorairajoo, M. J. Dumontier, M. R. Dumontier, V. Earles, R. Farrall, H. Feldman, E. Garderman, Y. Gong, R. Gonzaga, V. Grytsan, E. Gryz, V. Gu, E. Haldorsen, A. Halupa, R. Haw, A. Hrvojic, L. Hurrell, R. Isserlin, F. Jack, F. Juma, A. Khan, T. Kon, S. Konopinsky, V. Le, E. Lee, S. Ling, M. Magidin, J. Moniakis, J. Montojo, S. Moore, B. Muskat, I. Ng, J. P. Paraiso, B. Parker, G. Pintilie, R. Pirone, J. J. Salama, S. Sgro, T. Shan, Y. Shu, J. Siew, D. Skinner, K. Snyder, R. Stasiuk, D. Strumpf, B. Tuekam, S. Tao, Z. Wang, M. White, R. Willis, C. Wolting, S. Wong, A. Wrong, C. Xin, R. Yao, B. Yates, S. Zhang, K. Zheng, T. Pawson, B. F. Ouellette, and C. W. Hogue. The biomolecular interaction network database and related tools 2005 update. *Nucleic Acids Res.*, 33(Database issue):D418–D424, Jan 2005.

[3] D. A. Benson, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and D. L. Wheeler. Genbank. *Nucleic Acids Res.*, 33(Database issue):D34–D38, Jan 2005.

[4] T. H. Bo, B. Dysvik, and I. Jonassen. Lsimpute: accurate estimation of missing values in microarray data with least squares methods. *Nucleic Acids Res.*, 32(3):e34, Feb 2004.

[5] E. Camon, M. Magrane, D. Barrell, V. Lee, E. Dimmer, J. Maslen, D. Binns, N. Harte, R. Lopez, and R. Apweiler. The gene ontology annotation (goa) database: sharing knowledge in uniprot with gene ontology. *Nucleic Acids Res.*, 32(1):D262–D266, Jan 2004.

[6] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler. Free full text support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906–914, 2000.

[7] S. K. Gruvberger-Saal, P. Eden, M. Ringner, B. Baldetorp, G. Chebil, A. Borg, M. Ferno, C. Peterson, and P. S. Meltzer. Predicting continuous values of prognostic markers in breast cancer from microarray gene expression profiles. *Mol. Cancer Ther.*, 3(2):161–168, Feb 2004.

[8] L. Guttman. Some necessary conditions for common-factor analysis. *Psychometrika*, 19:149–161, 1954.

[9] Y. Kaneta, Y. Kagami, T. Katagiri, T. Tsunoda, I. Jin-nai, H. Taguchi, H. Hirai, K. Ohnishi, T. Ueda, N. Emi, A. Tomida, T. Tsuruo, Y. Nakamura, and R. Ohno. Prediction of sensitivity to STI571 among chronic myeloid leukemia patients by genome-wide cDNA microarray analysis. *Jpn. J. Cancer. Res.*, 93(8):849–856, Aug. 2002.

[10] T. Kato, Y. Murata, K. Miura, K. Asai, P. B. Horton, K. Tsuda, and W. Fujibuchi. Network-based de-noising improves prediction from microarray data. *BMC Bioinformatics*, 7(Suppl 1):S4, March 2006.

[11] J. M. Mariadason, D. Arango, Q. Shi, A. J. Wilson, G. A. Corner, C. Nicholas, M. J. Aranes, M. Lesser, E. L. Schwartz, and L. H. Augenlicht. Gene expression profiling-based prediction of response of colon carcinoma cells to 5-fluorouracil and camptothecin. *Cancer Res.*, 63(24):8791–8812, Dec 2003.

[12] J. Okutsu, T. Tsunoda, Y. Kaneta, T. Katagiri, O. Kitahara, H. Zembutsu, R. Yanagawa, S. Miyawaki, K. Kuriyama, N. Kubota, Y. Kimura, K. Kubo, F. Yagasaki, T. Higa, H. Taguchi, T. Tobita, H. Akiyama, A. Takeshita, Y. H. Wang, T. Motoji, R. Ohno, and Y. Nakamura. Prediction of chemosensitivity for patients with acute myeloid leukemia, according to expression levels of 28 genes selected by genome-wide complementary DNA microarray analysis. *Mol. Cancer Ther.*, 1(12):1035–1042, Oct 2002.

[13] J. E. Staunton, D. K. Slonim, H. A. Coller, P. Tamayo, M. J. Angelo, J. Park, U. Scherf, J. K. Lee, W. O. Reinhold, J. N. Weinstein, J. P. Mesirov, E. S. Lander, and T. R. Golub. Chemosensitivity prediction by transcriptional profiling. *Proc. Natl. Acad. Sci. U.S.A.*, 98(19):10787–10792, Sept 2001.

[14] M.E. Tipping and C.M. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11(2):443–482, 1999.

[15] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17(6):520–525, Jun 2001.

[16] K. Tsuda and G. Rätsch. Image reconstruction by linear programming. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, Cambridge, MA, 2004. MIT Press.

Subspace of Deformations

Seiichi UCHIDA†

† Faculty of Info. Science and Electrical Eng., Kyushu University, Fukuoka-shi, 812-8581 Japan

Abstract This paper describes subspace-related methods based on deformations of image patterns and sequential patterns. One of the merits of using subspace of deformations instead of conventional subspaces of appearance features is that the subspace of deformations will compactly represent the variation of deformed patterns. This paper also describes the roles of elastic matching techniques for establishing the subspace of deformations.

Key words subspace, elastic matching, eigen-deformation, image pattern, sequential pattern

1. Introduction

Conventional subspace methods are often based on appearance features of image patterns. For example, the eigenface method [1] construct a subspace of intensity-level features of face images. In contrast, the methods described in this paper are based on *deformations* instead of appearance features. The aim of this paper is the formalization of the procedure to establish subspaces of those deformations and to introduce the applications of the subspaces.

The first half of this paper focuses on subspace-related methods based on deformations of *image patterns*. The subspace of the deformations can represent deformed image patterns in a compact manner. For example, a set of translated image patterns can be tackled with only two basis vectors; i.e., , horizontal displacement vector and vertical displacement vector. In contrast, it is hard to represent the set of those images compactly by a subspace of their appearance feature space.

The deformations of image patterns can be extracted automatically by elastic matching, which is formulated as an optimization problem of the pixel-to-pixel correspondence between two image patterns. Since the resulting pixel-to-pixel correspondence represents the displacement of individual pixels, i.e., the deformation of one image pattern from the other. The subspace of the deformations can be obtained by applying principal component analysis (PCA) to the deformations extracted by elastic matching.

The latter half of this paper focuses on subspace-related methods based on deformations of sequential patterns. The deformations of sequential patterns often appear as the difference of pattern length. Due to this deformations, sequential patterns have rarely been handled in the framework of the conventional subspace methods. In addition, sequential patterns often undergo heavy nonlinear temporal/spatial fluctuation. Elastic matching to extract the deformation between two sequential patterns solves these problems and helps to establish the subspace of the deformations of sequential patterns.

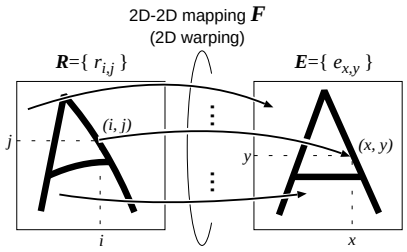


Fig. 1 Elastic matching between two image patterns.

2. Subspace-Related Methods Based on Deformations of Image Patterns

2.1 Extraction of Deformations by Elastic Matching

The deformation of an image pattern can be extracted by elastic matching, which is formulated as the following optimization problem. Consider two $I \times I$ image patterns $\boldsymbol{R} = \{\boldsymbol{r}_{i,j}\}$ and $\boldsymbol{E} = \{\boldsymbol{e}_{x,y}\}$, where $\boldsymbol{r}_{i,j}$ and $\boldsymbol{e}_{x,y}$ are pixel feature vectors at pixel (i,j) on $\boldsymbol{R}$ and (x,y) on $\boldsymbol{E}$, respectively. Let $\boldsymbol{F}$ denote a 2D-2D mapping from $\boldsymbol{R}$ to $\boldsymbol{E}$, i.e., $\boldsymbol{F} : (i,j) \mapsto (x,y)$. As shown in Figure 1, the mapping $\boldsymbol{F}$ determines the pixel-to-pixel correspondence from $\boldsymbol{R}$ to $\boldsymbol{E}$. Elastic matching between $\boldsymbol{R}$ and $\boldsymbol{E}$ is formulated as the minimization problem of the following objective function with respect to $\boldsymbol{F}$:

$$J_{\boldsymbol{R},\boldsymbol{E}}(\boldsymbol{F}) = \|\boldsymbol{R} - \boldsymbol{E}_{\boldsymbol{F}}\|, \tag{1}$$

where $\boldsymbol{E}_{\boldsymbol{F}}$ is the image pattern obtained by fitting $\boldsymbol{E}$ to $\boldsymbol{R}$, i.e., $\boldsymbol{E}_{\boldsymbol{F}} = \{\boldsymbol{e}_{x_{i,j},y_{i,j}}\}$, and $(x_{i,j}, y_{i,j})$ denotes the pixel of

$\boldsymbol{E}$ corresponding to the (i,j)th pixel of $\boldsymbol{R}$ under $\boldsymbol{F}$. On the minimization, several constraints (such as a smoothness constraint and boundary constraints) are often assumed to regularize $\boldsymbol{F}$.

Let $\tilde{\boldsymbol{F}}$ denote the mapping $\boldsymbol{F}$ which minimizes $J_{\boldsymbol{R},\boldsymbol{E}}(\boldsymbol{F})$ of (1). This mapping $\tilde{\boldsymbol{F}}$ represents the relative deformation of $\boldsymbol{E}$ from $\boldsymbol{R}$. Specifically, the deformation of $\boldsymbol{E}$ is extracted as the following $2I^2$-dimensional vector, called *deformation vector*,

$$\boldsymbol{v} = ((1-x_{1,1}, 1-y_{1,1}), \ldots, (i-x_{i,j}, j-y_{i,j}), \ldots, (I-x_{I,I}, I-y_{I,I})). \quad (2)$$

The constrained minimization of (1) with respect to $\boldsymbol{F}$ (i.e., the extraction of $\boldsymbol{v}$) is done by various optimization strategies. If the mapping $\boldsymbol{F}$ is defined as a parametric function, iterative strategies and exhaustive strategies are often employed for optimizing the parameters of $\boldsymbol{F}$. In contrast, if the mapping $\boldsymbol{F}$ is a non-parametric function, combinatorial optimization strategies, such as dynamic programming and local perturbation, and deterministic relaxation strategies are employed. The relation of the formulation of the elastic matching problem and its optimization strategy is summarized in [2].

2.2 Estimations of Eigen-Deformations by PCA

2.2.1 The Procedure of Estimation

Eigen-deformations of image patterns belonging to a category are intrinsic deformations of the category and defined as principal axes spanning a subspace of deformations. Thus, eigen-deformations can be estimated statistically from the actual deformations of N training samples $\{\boldsymbol{E}_1, \ldots, \boldsymbol{E}_n, \ldots, \boldsymbol{E}_N\}$. Let $\boldsymbol{v}_n$ denote the deformation of $\boldsymbol{E}_n$, which is obtained through the elastic matching between $\boldsymbol{R}$ and $\boldsymbol{E}_n$. Then the eigen-deformations $\{\boldsymbol{u}_m\}$ of the category can be estimated by applying PCA to the collected deformation vectors. Specifically, the eigen-deformations are obtained as the eigen-vectors of the covariance matrix $\Sigma = \sum_n (\boldsymbol{v}_n - \overline{\boldsymbol{v}})(\boldsymbol{v}_n - \overline{\boldsymbol{v}})^T / N$, where $\overline{\boldsymbol{v}}$ is the mean vector of $\{\boldsymbol{v}_n\}$.

2.2.2 Example: Eigen-Deformations of Handwritten Characters

Figure 2 shows the first three eigen-deformations estimated from 500 handwritten characters of the category "A". The most frequent deformation of "A" (i.e., $\boldsymbol{u}_1$) was the global slant transformation. Figure 3 shows the patterns $\boldsymbol{R}$ deformed by the first three eigen-deformations $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, and $\boldsymbol{u}_3$ with the amplification with $k\sqrt{\lambda_m}$ $(k = -2, -1, 0, 1, 2)$. These figures show that frequent deformations were extracted as the eigen-deformation at each category.

Figure 4 shows the cumulative proportion of each category. In all categories, the cumulative proportion exceeded

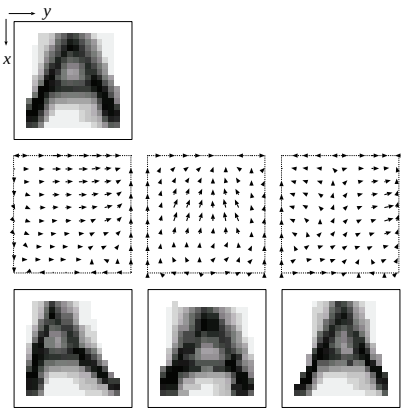


Fig. 2 Eigen-deformations of handwritten characters.

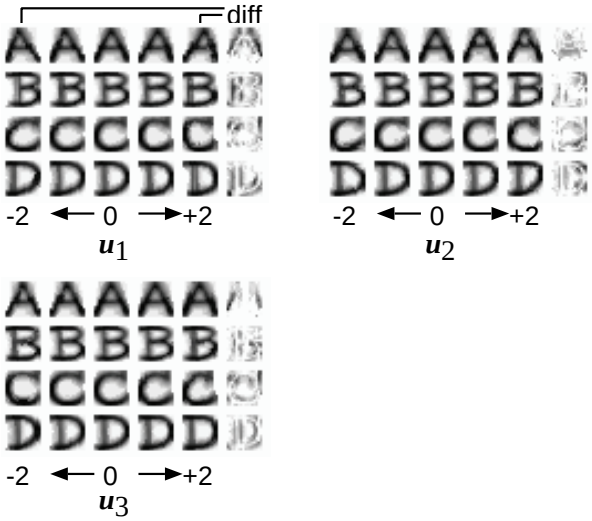


Fig. 3 Pattern $\boldsymbol{R}$ deformed by eigen-deformations.

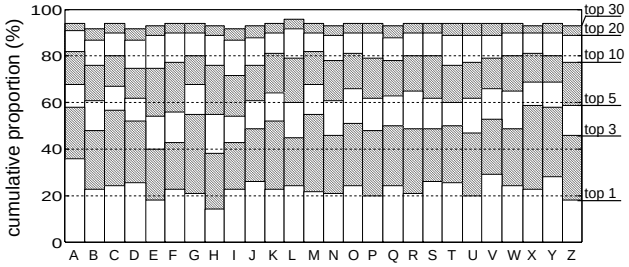


Fig. 4 Cumulative proportion of eigen-deformations.

50% with the top 3 $\sim$ 5 eigen-deformations and 80% with the top 10 $\sim$ 20 eigen-deformations. Thus, the distribution of deformation vectors was not isotropic and can be approximated by a small number of eigen-deformations. In other words, there existed a low-dimensional and efficient subspace of deformations.

2.3 Recognition with Eigen-Deformations

The simplest way to utilize the eigen-deformations for recognizing image patterns will be the subspace method where the similarity of an unknown input pattern $\boldsymbol{E}$ to a reference pattern $\boldsymbol{R}$ is evaluated by the norm of the vector

$\boldsymbol{v}' = \sum_{m=1}^{M} \langle \boldsymbol{v} - \overline{\boldsymbol{v}}, \boldsymbol{u}_m \rangle \boldsymbol{u}_m$, where $M (\leqq 2I^2)$ is a positive constant. The vector $\boldsymbol{v}$ is the deformation vector between $\boldsymbol{R}$ and $\boldsymbol{E}$ and the vector $\boldsymbol{v}'$ is its projection onto the subspace spanned by the M principal eigen-deformations $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_M$ of the category of $\boldsymbol{R}$. The similarity $\|\boldsymbol{v}'\|$ becomes small when the deformation vector $\boldsymbol{v}$ is rare one in the category.

The recognition performance by this similarity alone, however, will not be satisfactory. This is because this the similarity $\|\boldsymbol{v}'\|$ completely neglects the similarity of appearance features (e.g., gray-level). The remaining part of this section describes two image pattern recognition methods where the eigen-deformations are utilized together with the appearance features.

2.3.1 Recognition Method 1 [3]

The first method is based on the combination of similarities (or distances) in the appearance feature space and the deformation subspace spanned by the eigen-deformations, that is,

$$D_{\mathrm{hybrid}}(\boldsymbol{R}, \boldsymbol{E}) = (1-\alpha) D_{\mathrm{feat}}(\boldsymbol{R}, \boldsymbol{E}) + w D_{\mathrm{disp}}(\boldsymbol{R}, \boldsymbol{E}), \quad (3)$$

where $D_{\mathrm{feat}}(\boldsymbol{R}, \boldsymbol{E})$ is the elastic matching distance in the appearance feature space, i.e.,

$$D_{\mathrm{feat}}(\boldsymbol{R}, \boldsymbol{E}) = J_{\boldsymbol{R},\boldsymbol{E}}(\tilde{\boldsymbol{F}}), \quad (4)$$

$D_{\mathrm{disp}}(\boldsymbol{R}, \boldsymbol{E})$ is the distance in the deformation space, and w is a constant ($0 \leqq w \leqq 1$). The distance D_{disp} is defined as the following Mahalanobis distance,

$$\begin{aligned} D_{\mathrm{disp}}(\boldsymbol{R}, \boldsymbol{E}) &= (\boldsymbol{v} - \overline{\boldsymbol{v}})^T \Sigma^{-1} (\boldsymbol{v} - \overline{\boldsymbol{v}}) \\ &= sum_{m=1}^{2I^2} \frac{\langle \boldsymbol{v} - \overline{\boldsymbol{v}}, \boldsymbol{u}_m \rangle^2}{\lambda_m}. \end{aligned} \quad (5)$$

In practice, the following modified Mahalanobis distance [16] is employed instead of (5). Specifically, the higher-order eigenvalues λ_m ($m = M+2, \ldots, 2N^2$) are replaced by λ_{M+1}, to suppress the estimation errors of higher-order eigenvalues in (5). According to this replacement, (5) is reduced to

$$\begin{aligned} &D_{\mathrm{disp}}(\boldsymbol{R}, \boldsymbol{E}) \\ &\sim \frac{1}{\lambda_{M+1}} \|\boldsymbol{v} - \overline{\boldsymbol{v}}\| + \sum_{m=1}^{M} \left(\frac{1}{\lambda_m} - \frac{1}{\lambda_{M+1}} \right) \langle \boldsymbol{v} - \overline{\boldsymbol{v}}, \boldsymbol{u}_m \rangle^2. \end{aligned} \quad (6)$$

The parameter M is to be determined experimentally considering the cumulative proportion.

2.3.2 Recognition Method 2 [4]

The above recognition method has a weak-point that two heterogeneous distances D_{feat} and D_{disp} are added naively to create the single distance D_{hybrid}. In contrast, the second method can avoid this situation by embedding the eigen-deformations into an elastic matching procedure.

Consider that the mapping $\boldsymbol{F}$ is defined as a linear combination of eigen-deformations, i.e.,

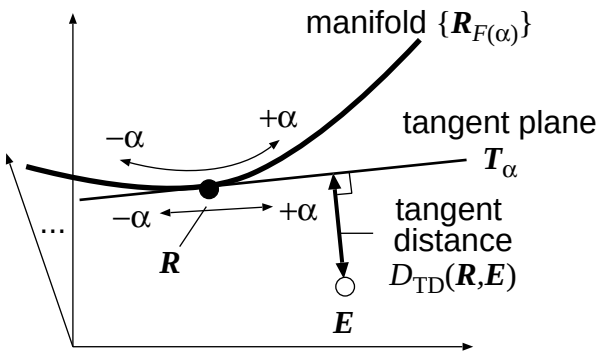


Fig. 5 Manifold $\boldsymbol{R}_\alpha$, its tangent plane $\boldsymbol{T}_\alpha$, and tangent distance $D_{\mathrm{TD}}(\boldsymbol{R}, \boldsymbol{E})$.

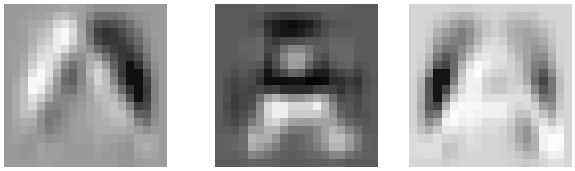


Fig. 6 Tangent vectors of the category "A", derived from $\boldsymbol{R}$ and eigen-deformations $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, and $\boldsymbol{u}_3$.

$$\boldsymbol{F}(\alpha) = \sum_{m=1}^{M} \alpha_m \boldsymbol{u}_m, \quad (7)$$

where $\alpha = \{\alpha_1, \ldots, \alpha_m, \ldots, \alpha_M\}$. Then an elastic matching problem with $\boldsymbol{F}(\alpha)$ can be formulated as the minimization problem of the following objective function:

$$J_{\boldsymbol{R},\boldsymbol{E}}(\alpha) = \left\| \boldsymbol{R}_{\boldsymbol{F}(\alpha)} - \boldsymbol{E} \right\|, \quad (8)$$

where $\boldsymbol{R}_{\boldsymbol{F}(\alpha)}$ is the reference pattern deformed by the mapping $\boldsymbol{F}(\alpha)$.

The set of deformed reference patterns, $\{\boldsymbol{R}_{\boldsymbol{F}(\alpha)} | \forall \alpha\}$, will form M-dimensional manifold in an I^2-dimensional appearance feature space. Thus the minimum value of $J_{\boldsymbol{R},\boldsymbol{E}}(\alpha)$ is equivalent to the shortest distance between the M-dimensional manifold and $\boldsymbol{E}$.

The minimization problem (8) with respect to α is hard to solve directly. This is because the M-dimensional parameter vector α to be optimized are involved in the nonlinear 2D-1D function $\boldsymbol{R}$. Thus, some approximation is required to solve the optimization problem.

In [4], the approximation scheme used in the tangent distance method [5] has been employed for the above minimization problem. As shown in Fig. 5, the minimum distance $\min_\alpha J_{\boldsymbol{R},\boldsymbol{E}}(\alpha)$ can be approximated by the following *tangent distance*,

$$D_{\mathrm{TD}}(\boldsymbol{R}, \boldsymbol{E}) = \min_{\alpha} \left\| \boldsymbol{T}_\alpha - \boldsymbol{E} \right\|, \quad (9)$$

where $\boldsymbol{T}_\alpha$ is the tangent plane of the manifold at $\alpha = 0$. The tangent plane is an M-dimensional hyperplane in the feature space and linear with respect to α. Thus the minimiza-

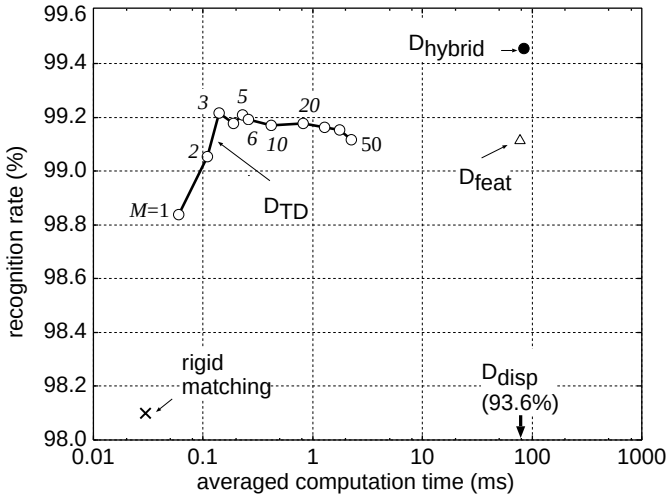


Fig. 7 Relation between computation time (ms) and recognition rate (%).

tion problem of (9) has a closed-form solution. Intuitively speaking, the distance $D_{\mathrm{TD}}(\boldsymbol{R}, \boldsymbol{E})$ is the Euclidean distance between the input $\boldsymbol{E}$ and its closest point on the tangent plane. Figure 6 shows three tangent vectors which span the tangent plane of the category "A". For more details, see [4].

2.3.3 Example: Recognition of Handwritten Characters

Figure 7 shows results of a handwritten character recognition experiment using 26 (categories) $\times$ 1100 (samples) isolated handwritten English uppercase character images from the standard character image database ETL6. The first 100 samples of each category were simply averaged to create one reference pattern $\boldsymbol{R}$ and the next 500 samples were used as training samples $\boldsymbol{E}_n$ to estimate the eigen-deformations. The remaining 500 samples ($13000 = 26 \times 500$ samples in total) were used as test samples $\boldsymbol{E}$.

The highest recognition rate (99.47%) was attained by D_{hybrid}. The recognition rate by D_{disp}, i.e., the recognition rate by evaluating only the deformation $\boldsymbol{v}$, was not sufficient. Thus, for the (dis)similarity of two image patterns, their appearance features should not be neglected. The recognition rates by D_{TD} were saturated around $M = 3$. This result is supported by the fast saturation of the cumulative proportion of Fig. 4.

2.4 Why the subspace of deformations?

One of the superiority of the eigen-deformations (i.e., the subspace of deformations) over the eigen-vectors in the feature space (i.e., conventional subspace of appearance features) is that the eigen-deformations will represent a set of deformed patterns in a compact manner. Consider an image pattern $\boldsymbol{R}$ and the set of image patterns created by translating $\boldsymbol{R}$. The number of the eigen-deformations estimated from the image pattern set is two; one will represent horizontal shift and the others vertical shift. In contrast, the number of the principal eigen-vectors in the feature space will be far larger than two. This superiority will hold for other deformations whenever they can be represented by a linear combination of few eigen-deformations. Thus, the subspace of deformations can be a more efficient representation of a pattern distribution than the subspace of the appearance features.

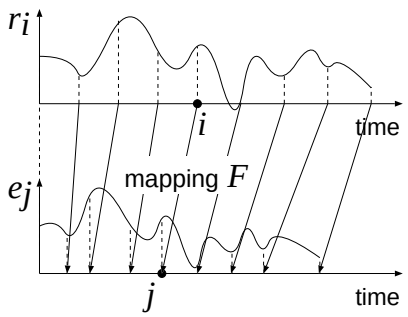


Fig. 8 Elastic matching between two sequential patterns.

2.5 Related Work

The original idea of the eigen-deformations, i.e., PCA of deformations, can be found in the point distribution models (PDM) proposed by Cootes et al. [6]. Shen and Davatzikos [7] have introduce an automatic deformation collection scheme into the PDM. PDM for curvilinear patterns has been applied to face recognition [8], Chinese character recognition [9], and hand posture recognition [10]. Uchida et al. [3] have extended the PDM to deal with fully 2D deformations and have applied to an elastic matching-based handwritten character recognition system.

Iwai et al. [11] have applied PCA to interframe motion vector fields obtained by block matching, which can be considered as the simplest elastic matching. Bing et al. [12] have proposed a face expression recognition method based on a subspace of face deformations. Naster et al. [13] have analyzed a deformation vector extended to deal with the distortion of appearance features. (Specifically, they have extended XY displacement to XYI displacement.)

3. Subspace-Related Methods Based on Deformations of Sequential Patterns

3.1 Extraction of Deformations by Elastic Matching

Consider two sequential patterns, $\boldsymbol{R} = \boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_i, \ldots, \boldsymbol{r}_I$ and $\boldsymbol{E} = \boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_j, \ldots, \boldsymbol{e}_J$. Their elements $\boldsymbol{r}_i$ and $\boldsymbol{e}_j$ are d-dimensional feature vectors representing the features at i and j. For handwriting patterns, the feature vector $\boldsymbol{r}_i$ may be a 3-dimensional vector comprised of x-coordinate, y-coordinate, and local direction at time i.

Let $\boldsymbol{F}$ denote a 1D-1D mapping from $\boldsymbol{R}$ to $\boldsymbol{E}$, i.e., $\boldsymbol{F} : i \mapsto$

j. Figure 8 depicts $\boldsymbol{F}$. Elastic matching between $\boldsymbol{R}$ and $\boldsymbol{E}$ is formulated as the minimization of the following objective function with respect to $\boldsymbol{F}$,

$$J_{\boldsymbol{R},\boldsymbol{E}}(\boldsymbol{F}) = \|\boldsymbol{R} - \boldsymbol{E}_{\boldsymbol{F}}\|, \tag{10}$$

where $\boldsymbol{E}_{\boldsymbol{F}}$ is the sequential pattern obtained by fitting $\boldsymbol{E}$ to $\boldsymbol{R}$, i.e., $\boldsymbol{E}_{\boldsymbol{F}} = \boldsymbol{e}_{j_1}, \ldots, \boldsymbol{e}_{j_i}, \ldots, \boldsymbol{e}_{j_I}$, where j_i represents the $i - j$ correspondence under $\boldsymbol{F}$. On the minimization, several constraints (such as the monotonicity and continuity constraint defined as $j_i - j_{i-1} \in \{0, 1, 2\}$ and boundary constraints $j_1 = 1$ and $j_I = J$) are often assumed to regularize $\boldsymbol{F}$. This constrained minimization problem can be solved effectively by a DP algorithm and its detail are omitted here.

The deformation of $\boldsymbol{E}$ from $\boldsymbol{R}$ is represented by the following $I \cdot d$-dimensional deformation vector,

$$\boldsymbol{v} = (\boldsymbol{e}_{j_1} - \boldsymbol{r}_1, \ldots, \boldsymbol{e}_{j_i} - \boldsymbol{r}_i, \ldots, \boldsymbol{e}_{j_I} - \boldsymbol{r}_I)^T. \tag{11}$$

It should be noted that the dimension of the above deformation vector $\boldsymbol{v}$ is fixed at $I \cdot d$ and independent of the length of $\boldsymbol{E}$, i.e., J. This property is very important to apply various statistical methods, such as PCA, to sequential patterns.

Also note that it is possible to define $\boldsymbol{v}$ as

$$\boldsymbol{v} = (j_1 - 1, \ldots, j_i - i, \ldots, j_I - I)^T.$$

Although this definition is a straightforward modification of the deformation vector of (2), we will use $\boldsymbol{v}$ of (11) as a deformation vector here. This is because for sequential patterns (e.g., online character patterns) $\boldsymbol{r}_i$ and $\boldsymbol{e}_j$ are often spatial features and thus their difference represents a deformation.

3.2 Estimation of Eigen-Deformations

3.2.1 The Procedure of Estimation

Eigen-deformations of sequential patterns are also estimated by the procedure of 2.2.1; that is, they can be estimated by applying PCA to the covariance matrix of $\boldsymbol{v}$.

3.2.2 Example: Eigen-Deformations of Online Character Patterns

Figure 9 shows online character patterns generated as $\boldsymbol{R} + \overline{\boldsymbol{v}} + \pm 2\sqrt{\lambda_m}\boldsymbol{u}_m$ $(m = 0, 1, 2)$ [14]. Those patterns are reference patterns deformed by their mean deformation vector $\overline{\boldsymbol{v}}$ and the first two eigen-deformations $\boldsymbol{u}_m$. At each of 10 categories, the eigen-deformations were estimated by using about 1,000 online digit samples from UNIPEN Train-R01/V07 database (1a) [15]. Note that the effect of $\overline{\boldsymbol{v}}$ was not significant because $\boldsymbol{R}$ was set around the center of the set of the training samples by a clustering technique and thus the norm of $\overline{\boldsymbol{v}}$ was small.

This figure shows that deformations frequently observed in actual characters were detected as principal eigen-deformations. For example, the first eigen-deformation of "6" represents the vertical variation of its loop part, and the second one represents the horizontal variation of the loop part.

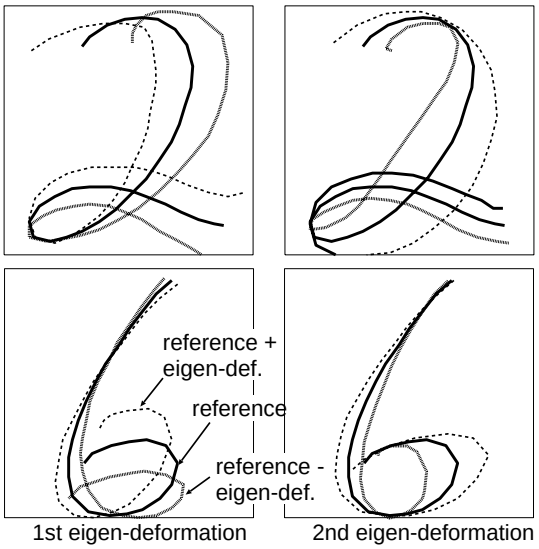


Fig. 9 Reference pattern deformed by the first two eigen-deformations of "2" and "6".

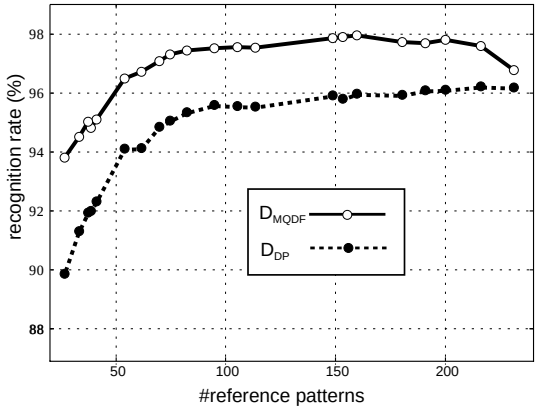


Fig. 10 Accuracy of online character recognition based on eigen-deformations.

3.3 Recognition with Eigen-Deformations [14]

3.3.1 Methodology

For sequential pattern recognition based on the eigen-deformations, the following quadratic discrimination function (QDF) is a possible choice. The QDF is the Bayes discrimination function under the assumption that the deformation vectors have a Gaussian distribution and defined as

$$\begin{aligned} & D_{\mathrm{QDF}}(\boldsymbol{R}, \boldsymbol{E}) \\ &= (\boldsymbol{v} - \overline{\boldsymbol{v}})^T \Sigma^{-1} (\boldsymbol{v} - \overline{\boldsymbol{v}}) + \log|\Sigma| + I \cdot d \log 2\pi \\ &= \sum_{m=1}^{I \cdot d} \frac{\langle \boldsymbol{v} - \overline{\boldsymbol{v}}, \boldsymbol{u}_m \rangle^2}{\lambda_m} + \log \prod_{m=1}^{I \cdot d} \lambda_m + I \cdot d \log 2\pi. \end{aligned} \tag{12}$$

The last term, $I \cdot d \log 2\pi$, cannot be omitted here because each category has a different dimension of $\boldsymbol{v}$ (i.e., $I \cdot d$).

As noted 2.3.1, the estimation errors of higher-order eigenvalues are amplified in (12). Thus, the modified quadratic

discriminant function (MQDF) [16] was employed, where the higher-order eigenvalues λ_m $(m = M+1, \ldots, I \cdot d)$ are replaced by λ_{M+1}, i.e.,

$$D_{\mathrm{MQDF}}(\boldsymbol{R}_c, \boldsymbol{E}) \sim \frac{1}{\lambda_{M+1}} \|\boldsymbol{v} - \overline{\boldsymbol{v}}\|^2 + \sum_{m=1}^{M} \left(\frac{1}{\lambda_m} - \frac{1}{\lambda_{M+1}} \right) \langle \boldsymbol{v} - \overline{\boldsymbol{v}}, \boldsymbol{u}_m \rangle^2 + \log \left\{ (\lambda_{M+1})^{I \cdot d - M} \prod_{m=1}^{M} \lambda_m \right\} + I \cdot d \log 2\pi. \quad (13)$$

The parameter M is to be determined experimentally.

3.3.2 Example: Recognition of Online Characters [14]

Figure 10 shows the results of an online character recognition experiment using digit samples from the UNIPEN database. Recognition rates attained by D_{MQDF} are plotted as a function of the number of total reference patterns. The recognition rates attained by the conventional DP-matching distance (D_{DP}), which equals to the minimum value of (10), are also plotted.

As shown in Fig. 10, MQDF with the eigen-deformations outperformed the DP-matching distance. This will be because elastic matching results $\boldsymbol{F}$ which were deviated from the distribution of the deformations of the category were penalized by the eigen-deformations in MQDF. Thus, the above recognition method can avoid misrecognitions due to overfitting, which is the phenomenon that the distance between $\boldsymbol{E}$ and $\boldsymbol{R}$ of a wrong category is underestimated by unnatural mapping $\boldsymbol{F}$.

This result also proves that D_{MQDF} outperforms that statistical dynamic time warping (SDTW) [17], which is a recent and sophisticated online character recognition technique. In fact, it has been reported in [17] that SDTW attained 97.10% on the same UNIPEN data set by 150 reference patterns.

3.4 Prediction with Eigen-Deformations

Assume that the current frame j of an input sequential pattern $\boldsymbol{E}$ corresponds to the ith frame of the reference pattern $\boldsymbol{R}$ of a certain category. Then, we can predict that the future input frame $\widehat{\boldsymbol{e}}_{j+\tau}$ $(\tau > 0)$ will be similar to $\boldsymbol{r}_{i+\tau}$. Figure 11 (a) illustrates this prediction technique based on this simple extrapolation. The prediction accuracy depends on the similarity between $\boldsymbol{R}$ and $\boldsymbol{E}$. In other words, the prediction accuracy will be degraded severely if the input gesture $\boldsymbol{E}$ deviates from $\boldsymbol{R}$.

Figure 11(b) shows an alternative prediction technique using the eigen-deformations. Since the eigen-deformations span the subspace of frequent variations, the derivation of the input pattern from the reference pattern can be approximated by a weighted linear combination of the eigen-deformations. Using the eigen-deformations, the future input frame $\widehat{\boldsymbol{e}}_{j+\tau}$ is predicted by

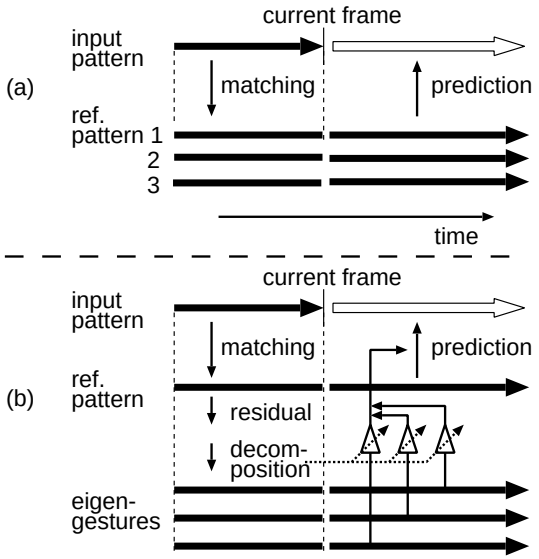


Fig. 11 Illustration of two gesture prediction methods: (a) Prediction based on simple extrapolation. (b) Prediction based on a linear combination of eigen-deformations.

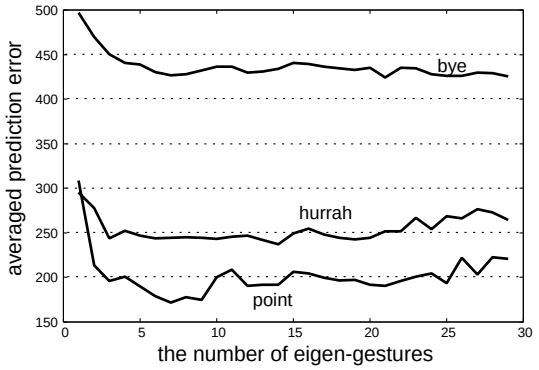


Fig. 12 The number of eigen-deformations versus prediction error.

$$\widehat{\boldsymbol{e}}_{j+\tau} = \boldsymbol{r}_{i+\tau} + \sum_{m=1}^{M} \alpha_m u_{m,i+\tau}, \quad (14)$$

where $u_{m,i}$ is the ith frame of the eigen-deformation $\boldsymbol{u}_m$.

The weight α_k is estimated by fitting the model (14) to the observed part of the input, that is, by minimizing the following criterion,

$$\sum_{t=1}^{i} \left\| \left(\boldsymbol{r}_t + \sum_{m=1}^{M} \alpha_m u_{m,t} \right) - \boldsymbol{e}_{j_t} \right\|. \quad (15)$$

Figure 12 shows averaged prediction errors (at $j = 0.5/J$) for three gesture patterns, "bye", "hurrah", and "point". The prediction errors are plotted as a function of M, the number of the eigen-deformations used for the prediction. The highest accuracy was attained with about 6∼7 eigen-deformations for each category. This result indicates that each gesture category has a low-dimensional subspace suitable for predicting subsequent postures.

The prediction rule of (14) assumes that the speeds of the

input pattern and the reference pattern are the same. This assumption will not be acceptable for many cases. Future work will focus on the guess and the utilization of the difference of the speeds to reduce the prediction errors.

3.5 Related Work

Sequential patterns are often re-sampled to have the same dimension in advance to applying PCA or other statistical analysis techniques. For example, Deepu et al. [18] have proposed an online character recognition technique based on a subspace method where all online character patterns are re-sampled to have a constant number of data points. The online character recognition technique by Zheng et al. [19] is more radical because they used only two points (i.e., the start point and the end point) for each character stroke segment. In the handwriting synthesis technique by Wang et al. [20], online cursive handwritings are firstly aligned to be the same dimension and then PCA is applied to them. PCA-based gesture/motion analysis techniques [21]∼[23] also re-sampled gesture patterns to have the same dimension. An exception is Martens and Claesen [24], which employed elastic matching to extract a fixed-dimensional deformation vector from online signatures.

The prediction schemes based on PCA can be found image interpolation techniques [13], [25].

4. Conclusion

Subspace-related methods based on the deformations of image patterns and sequential patterns have been discussed. In those methods, elastic matching techniques are fully utilized not only to extract deformations of target patterns but also to adjust dimensional differences of (especially sequential) patterns.

It has been shown experimentally that estimated principal deformations, called eigen-deformations, represent frequently observed deformations. The eigen-deformations of image patterns can be used for recognizing the image patterns, while the help of appearance features is inevitable. In contrast, the eigen-deformations of sequential patterns can be used for recognizing the sequential patterns by themselves.

Acknowledgment

The author thanks Prof. H. Sakoe (Kyushu Univ.) for his invaluable supervision. The author also thanks Mr. H. Mitoma (Automotive Systems, Hitachi Ltd.) and Mr. M. Nakajima (Kyushu Univ.) for their efforts at online character recognition and gesture prediction, respectively.

References

[1] M. Turk and A. Pentland, "Eigenfaces for recognition," Journal of Cognitive Neuroscience, vol. 3, no. 1, pp. 71–86, 1991.

[2] S. Uchida and H. Sakoe, "A survey of elastic matching techniques for handwritten character recognition," IEICE Trans. Inf. & Syst., vol. E88-D, no. 8, pp. 1781–1790, 2005.

[3] S. Uchida and H. Sakoe, "Eigen-deformations for elastic matching based handwritten character recognition," Pattern Recognit., vol. 36, no. 9, pp. 2031–2040, 2003.

[4] S. Uchida and H. Sakoe, "Handwritten character recognition using elastic matching based on a class-dependent deformation model," Proc. ICDAR, vol. 1 of 2, pp. 163–167, 2003.

[5] P. Simard, Y. Le Cun, J. Denker, and B. Victorri, "An efficient algorithm for learning invariances in adaptive classifier," Proc. ICPR, vol. 2, pp. 651–655, 1992.

[6] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models - their training and application," Comput. Vis. Image Und., vol. 61, no. 1, pp. 38–59, 1995.

[7] D. Shen and C. Davatzikos, "An adaptive-focus deformable model using statistical and geometric information," IEEE Trans. PAMI, vol. 22, no. 8, pp. 906-913, 2000.

[8] A. Lanitis, C. J. Taylor, and T. F. Cootes, "Automatic interpretation and coding of face images using flexible models," IEEE Trans. PAMI, vol. 19, no. 7, pp. 743–756, 1997.

[9] D. Shi, S. R. Gunn, and R. I. Damper, "Handwritten Chinese radical recognition using nonlinear active shape models," IEEE Trans. PAMI, vol. 25, no. 2, pp. 277–280, 2003.

[10] T. Ahmad, C. J. Taylor, A. Lanitis, and T. F. Cootes, "Tracking and recognising hand gestures, using statistical shape models" Image Vis. Computing, vol. 15, pp. 345–352, 1997.

[11] Y. Iwai, T. Hata, M. Yachida, "Gesture recognition based on subspace method and hidden Markov model," Proc. IROS, vol. 2 of 2, pp. 960–966, 1997.

[12] Y. Bing, C. Ping, and J. Lianfu, "Recognizing faces with expressions: within-class space and between-class space," Proc. ICPR, vol. 1 of 4, pp. 139–142, 2002.

[13] C. Naster, B. Moghaddam, and A. Pentland, "Flexible images: matching and recognition using learned deformations," Comput. Vis. Image Und., vol. 65, no. 2, pp. 179–191, 1997.

[14] H. Mitoma, S. Uchida, and H. Sakoe "Online character recognition based on elastic matching and quadratic discrimination," Proc. ICDAR, vol. 1 of 2, pp.36–40, 2005.

[15] I. Guyon, L. Schomaker, R. Plamondon, M. Liberman, and S. Janet, "UNIPEN project of on-line data exchange and recognizer benchmarks," Proc. ICPR, pp. 29–33, 1994.

[16] F. Kimura, K. Takashina, S. Tsuruoka, "Modified quadratic discriminant functions and the application to Chinese character recognition," IEEE Trans. PAMI, vol. 9, no. 1, pp. 149-153, 1987.

[17] C. Bahlmann and H. Burkhardt, "The writer independent online handwriting recognition system flog on hand and cluster generative statistical dynamic time warping," IEEE Trans. PAMI, vol. 26, no. 3, pp. 299–310, 2004.

[18] V. Deepu, S. Madhvanath, A. G. Ramakrishnan, "Principal component analysis for online handwritten character recognition," Proc. ICPR, vol. 2 of 4 , pp. 327–330, 2004.

[19] J. Zheng, X. Ding, Y. Wu, and Z. Lu, "Spatio-temporal unified model for on-line handwritten Chinese character recognition," Proc. ICDAR, pp. 649–652, 1999.

[20] J. Wang, C. Wu, Y.-Q. Xu, and H.-Y. Shum, "Combining shape and physical models for online cursive handwriting synthesis," Int. J. Doc. Ana. Recog., vol. 7, no. 4, pp. 219–227, 2005.

[21] T. D. Sanger, "Optimal movement primitives," Advances in Neural Info. Proc. Systems, vol. 7, pp. 1023–30, 1995.

[22] Y. Yacoob and M. Black, "Parameterized modeling and

recognition of activities," Comput. Vis. Image Und., vol. 73, no. 2, pp. 232–247, 1999.

[23] A. Fod, M. Mataric,and O. C. Jenkins, "Automated derivation of primitives for movement classification," Autonomous Robots, vol. 12, no. 1, pp. 39–54, 2002.

[24] R. Martens and L. Claesen, "On-line signature verification by dynamic time-warping," Proc. ICPR, pp. 38–42, 1996.

[25] T. Amano, "Image interpolation by high dimensional projection based on subspace method," Proc. ICPR, vol. 4 of 4, pp. 665-668, 2004.

部分空間法における原点の位置と認識性能に関する考察

岩村　雅一† 　大町真一郎†† 　阿曽　弘具††

† 大阪府立大学大学院工学研究科　〒 599–8531 堺市中区学園町 1–1
†† 東北大学大学院工学研究科　〒 980–8579 仙台市青葉区荒巻字青葉 6–6–05
E-mail: †masa@cs.osakafu-u.ac.jp, ††{machi,aso}@ecei.tohoku.ac.jp

あらまし　部分空間法は，簡潔で，かつ望ましい性質を持つため，パターン認識においてよく用いられる認識手法の一つである．様々な改良法が提案されているが，そのほとんど全ては「暗に」特徴ベクトルの原点に依存している．ところが，原点の位置が部分空間法の認識性能にどのような影響を与えるのか，十分に解明されていない．本稿では，「コアアフィン部分空間」と「誤認識穴」という概念を導入し，この問題について議論する．また，よく知られた部分空間法である CLAFIC 法と投影距離法の，ある種の類似性についても言及する．

キーワード　パターン認識，部分空間法，原点，CLAFIC 法，投影距離法

On the Position of the Origin and Recognition Performance of Subspace Method

Masakazu IWAMURA†, Shinichiro OMACHI††, and Hirotomo ASO††

† Graduate School of Engineering, Osaka Prefecture University
1–1 Gakuencho, Naka, Sakai, 599–8531 Japan
†† Graduate School of Engineering, Tohoku University
6–6–05 Aoba, Aramaki, Aoba, Sendai, 980–8579 Japan
E-mail: †masa@cs.osakafu-u.ac.jp, ††{machi,aso}@ecei.tohoku.ac.jp

Abstract　Subspace method is one of the most popular recognition methods in pattern recognition because of its simplicity and some desirable properties. Many varieties of the subspace methods have been proposed. Though most of them *implicitly* depend on the origin of feature vectors, the effect of the position of the origin to the recognition performance have not analyzed enough. In this paper, we discuss and address this problem by introducing the concepts of *core affine subspace* and *misclassification hole*. We also mention the similarity between the CLAFIC method and the projection distance in a sense under an assumption.

Key words　pattern recognition, subspace method, origin, CLAFIC method, projection distance

1. はじめに

部分空間法は比較的少ない計算量で高い認識性能を実現できることや，少数サンプル問題に対する頑健性を持つことで知られており，パターン認識においてよく用いられる認識手法の一つである．これまでに多種多様な部分空間法が提案されている．

Watanabe によって提案された CLAFIC 法 [1] は最も単純な手法といえる．複合類似度法と混合類似度法は飯島によって提案され，文字認識に適用された [2], [3]．部分空間を逐次的に学習する学習部分空間法は Kohonen によって提案された [3], [4]．また，全てのクラスの部分空間が直交する直交部分空間法 [3] も知られている．その他，顔画像の時系列から計算される部分空間と，あらかじめ学習しておいた顔画像の部分空間の類似度を計算することにより頑健な個人認証が可能な相互部分空間法 [5] や制約相互部分空間法 [6] なども提案されている．さらに，非線形な識別境界を持つ非線形部分空間法も提案されている [7]〜[9]．

上記の全ての部分空間法において，各クラスの分布を表現する部分空間は座標の原点を通るため，「暗に」特徴ベクトルの原点に依存している．しかし，この原点は特徴量の原点に過ぎず，必ずしも識別に有利な原点であるとは限らない．ところが，部分空間法における原点の位置に関する解析は，一部の実験的な試み (例えば文献 [10]) を除いてほとんど行われていない．

そこで本稿では，原点の位置が部分空間法に与える影響を

CLAFIC 法を用いて考察する．具体的には，まず特徴ベクトルの平均ベクトルを固有ベクトルであるとみなし，CLAFIC 法の類似度の式を原点の位置に依存する項と依存しない項に分解する．このうち，原点の位置に依存せず，パターンの分布にのみ依存する項を「コアアフィン部分空間」と名付ける．

CLAFIC 法の類似度の式について解析すると，クラスを表す部分空間 (コアアフィン部分空間) 上に認識不可能な超球状の領域が発生し，誤認識が起こり得ることがわかった．この領域を「誤認識穴」と名付け，さらに解析すると，誤認識穴の大きさと位置は原点の位置に依存して決まり，誤認識穴とサンプルの分布の関係が CLAFIC 法における認識性能を左右することがわかった．従って，CLAFIC 法において正しく認識できるためには，原点の位置が特定の条件を満たしている必要があると考えられる．

そこで，認識に有効な原点の位置について考察と実験を行い，妥当な原点の位置を求める方法 “adaptive origin subspace method” を提案する．手書き数字を用いた認識実験により，原点の位置と認識性能の関係についてのさらなる考察を行う．

本稿ではさらに，CLAFIC 法と投影距離法 [11] の関係についても言及する．多くの原点の位置に依存する部分空間法に対して，投影距離法 [11] は原点に依存しないある種の部分空間法であると考えられている．CLAFIC 法の族と投影距離法 (擬似ベイズ) の族に関しては黒澤の研究 [12] があるが，本稿では CLAFIC 法と投影距離法のある種の類似性を文献 [12] とは別の形で述べる．

なお、本稿では部分空間やアフィン部分空間(注1)の基底を表すときのように $\{\boldsymbol{\phi}_i\}$ の一部のみを指し示すときは，$\{\boldsymbol{\phi}_i\}_1^{r'}$ のように表記する．これは共分散行列の第 1 固有ベクトルから第 r' 固有ベクトルまでで張られたアフィン部分空間の基底であることを意味している．

2. 部分空間法

本節では，古典的な部分空間法である CLAFIC 法と投影距離法について簡単に述べる．

2.1 CLAFIC 法 [1]

2.1.1 学　　習

$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t$ を学習サンプルの特徴ベクトルとすると，自己相関行列 $\boldsymbol{R}$ は

$$\boldsymbol{R} = \frac{1}{t}\sum_{i=1}^{t} \boldsymbol{x}_i \boldsymbol{x}_i^{\mathrm{T}} \tag{1}$$

で与えられる．$\boldsymbol{\psi}_i$ を $\boldsymbol{R}$ の第 i 固有ベクトルとする．

2.1.2 認　　識

$\boldsymbol{x}$ をテストサンプルの特徴ベクトルとする．CLAFIC 法の類似度 $s(\boldsymbol{x})$ は

$$s(\boldsymbol{x}) = \sum_{i=1}^{r} \left(\boldsymbol{x}^{\mathrm{T}} \boldsymbol{\psi}_i\right)^2 \tag{2}$$

で表わされる．ここで r は認識に用いる部分空間の次元数を表すパラメータである．テストサンプルは，$\boldsymbol{x}$ とクラスの類似度 $s(\boldsymbol{x})$ が最大になるクラスに属すると判断される．

2.2 投影距離法 [11]

2.2.1 学　　習

平均ベクトル $\boldsymbol{\mu}$ と共分散行列 $\boldsymbol{\Sigma}$ はそれぞれ

$$\boldsymbol{\mu} = \frac{1}{t}\sum_{i=1}^{t} \boldsymbol{x}_i \tag{3}$$

$$\boldsymbol{\Sigma} = \frac{1}{t}\sum_{i=1}^{t} (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^{\mathrm{T}} \tag{4}$$

で与えられる．λ_i と $\boldsymbol{\phi}_i$ をそれぞれ $\boldsymbol{\Sigma}$ の第 i 固有値，固有ベクトルとする．固有値は降順に並んでいるとする．

2.2.2 認　　識

投影距離法の投影距離 $d(\boldsymbol{x})$ は

$$d(\boldsymbol{x}) = ||\boldsymbol{x} - \boldsymbol{\mu}||^2 - \sum_{i=1}^{r'} \left\{(\boldsymbol{x} - \boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\phi}_i\right\}^2 \tag{5}$$

で表わされる．ここで r' は認識に用いるアフィン部分空間の次元数を表すパラメータである．テストサンプルは，$\boldsymbol{x}$ とクラスの投影距離 $d(\boldsymbol{x})$ が最小になるクラスに属すると判断される．

3. CLAFIC 法における誤認識穴

本節では,「誤認識穴」という概念を導入する．これは，クラスを表す部分空間にあたかも穴が開いたように誤認識が起こる領域のことを指す．これを導入するために，平均ベクトル (の全体もしくは一部のみ) を自己相関行列 $\boldsymbol{R}$ の固有ベクトルであると解釈する．本節の構成は，まず 3.1 節で，説明が比較的容易であるが，仮定が若干厳しい場合について述べ，その後 3.2 節で，より一般的な場合について述べる．

3.1 平均ベクトルと自己相関行列の第 1 固有ベクトルの方向が近似的に等しい場合 (仮定 A)

3.1.1 固有ベクトルとしての平均ベクトル

式 (1) と式 (4) から，$\boldsymbol{\Sigma}$ と $\boldsymbol{R}$ の関係が次式のように導ける．

$$\boldsymbol{R} = \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^{\mathrm{T}} \tag{6}$$

$\boldsymbol{\Sigma}$ は，固有値 $\{\lambda_i\}$ と固有ベクトル $\{\boldsymbol{\phi}_i\}$ に分解することで，

$$\boldsymbol{\Sigma} = \sum_{i=1}^{d} \lambda_i \boldsymbol{\phi}_i \boldsymbol{\phi}_i^{\mathrm{T}} \tag{7}$$

と変形できる．ここで d は特徴ベクトルの次元数である．式 (6) と式 (7) を用いると，次式が得られる．

$$\boldsymbol{R} = \|\boldsymbol{\mu}\|^2 \left(\frac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|}\right)\left(\frac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|}\right)^{\mathrm{T}} + \sum_{i=1}^{d} \lambda_i \boldsymbol{\phi}_i \boldsymbol{\phi}_i^{\mathrm{T}} \tag{8}$$

この式は自己相関行列 $\boldsymbol{R}$ を共分散行列の固有ベクトルで表現したものである．式 (8) において，第 d 固有ベクトルまででではなく，第 r' 固有ベクトルまでしか使わない場合を考えると，

$$\boldsymbol{R}_{\boldsymbol{r'}} \equiv \|\boldsymbol{\mu}\|^2 \left(\frac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|}\right)\left(\frac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|}\right)^{\mathrm{T}} + \sum_{i=1}^{r'} \lambda_i \boldsymbol{\phi}_i \boldsymbol{\phi}_i^{\mathrm{T}} \tag{9}$$

(注1)：原点を含まない部分空間は，アフィン部分空間と呼ばれる．詳細については文献 [4] の 2.1 節を参照のこと．

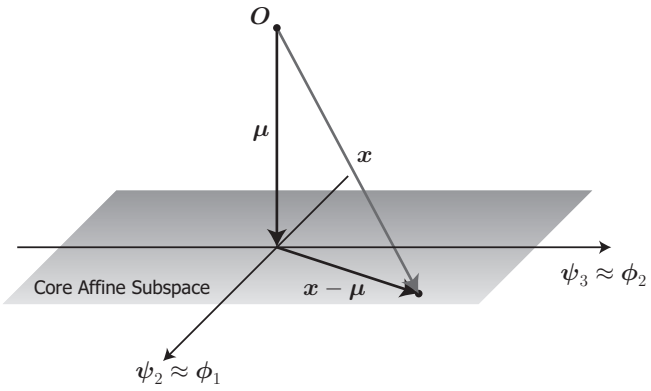


図 1　Core affine subspace in the CLAFIC method in the orthogonal case.

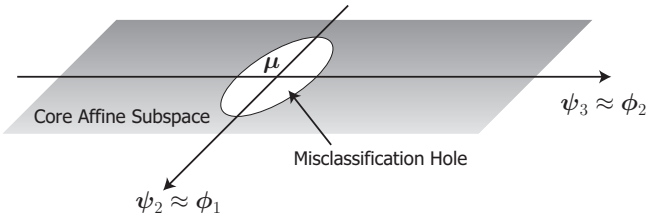


図 2　The region that a test sample is misclassified only with the core affine subspace in the orthogonal case.

となる．式 (9) において，もし $\frac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|}$ と $\{\boldsymbol{\phi}_i\}_1^{r'}$ が直交していれば，$\|\boldsymbol{\mu}\|^2$ と $\frac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|}$ はそれぞれ $\boldsymbol{R}$ の固有値と固有ベクトルとみなすことができる．直交条件に加えてさらに，$\|\boldsymbol{\mu}\|^2 > \lambda_1$ を満たせば，$\|\boldsymbol{\mu}\|^2$ と $\frac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|}$ は第 1 固有値と第 1 固有ベクトルとみなすことができる．経験的に，正規化した平均ベクトルは自己相関行列 $\boldsymbol{R}$ の第 1 固有ベクトルと似ていることが多く，$\boldsymbol{R}$ の第 1 固有値は第 2 固有値よりもかなり大きいため，この仮定は妥当であると考えられる．本節では，$\frac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|}$ と $\{\boldsymbol{\phi}_i\}_1^{r'}$ の直交性と $\|\boldsymbol{\mu}\|^2 > \lambda_1$ を仮定して，以降の議論を進める．

3.1.2　コアアフィン部分空間

式 (9) から，(i) $\frac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|}$ と $\{\boldsymbol{\phi}_i\}_1^{r'}$ が直交している，(ii) $\|\boldsymbol{\mu}\|^2 > \lambda_1$ である，(iii) $r = r' + 1$ である，という条件を満たせば，$\boldsymbol{R}$ の第 2 から第 r 固有ベクトルが張るアフィン部分空間と，$\boldsymbol{\Sigma}$ の第 1 から第 r' 固有ベクトルが張るアフィン部分空間は近似的に同一であるとみなせることがわかる．この近似的に同一であるアフィン部分空間は，サンプルの分布をよく表現しており，「コアアフィン部分空間」と呼ぶことにする．

このようなコアアフィン部分空間を定義した理由は，コアアフィン部分空間が原点の位置には影響されず，後の解析に便利だからである．本節の後半では，コアアフィン部分空間が原点の位置とは独立であることを導く．

まず，CLAFIC 法における類似度を表す式 (2) を原点の位置に依存する項と依存しない項に分解する．$\frac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|}$ と $\{\boldsymbol{\phi}_i\}_1^{r'}$ が直交であり，$\|\boldsymbol{\mu}\|^2 > \lambda_1$ であるという条件の下で，$\boldsymbol{R}$ の第 1 固有ベクトル $\boldsymbol{\psi}_1$ は次のように近似できる．

$$\boldsymbol{\psi}_1 \approx \frac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|} \tag{10}$$

第 2 固有ベクトル以降は第 1 固有ベクトル $\boldsymbol{\psi}_1$ と直交するので，

$$\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\psi}_i \approx \begin{cases} \|\boldsymbol{\mu}\|, & i = 1 \\ 0, & i \geqq 2 \end{cases} \tag{11}$$

のように近似できる．さらに，図 1 のように $\boldsymbol{x} - \boldsymbol{\mu}$ はコアアフィン部分空間上にあり，$\boldsymbol{\psi}_1$ はコアアフィン部分空間と直交しているので，

$$(\boldsymbol{x} - \boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\psi}_1 \approx 0 \tag{12}$$

が成り立つ．ここで式 (10)，式 (11)，式 (12) を用い，$\{\boldsymbol{\phi}_i\}_1^{r'}$ と $\{\boldsymbol{\psi}_i\}_2^{r}$ が張るアフィン部分空間が近似的に同一であることを仮定すると，式 (2) は次のように変形できる．

$$\begin{aligned} s(\boldsymbol{x}) &= \sum_{i=1}^{r} \left(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\psi}_i\right)^2 \\ &= \sum_{i=1}^{r} \left\{(\boldsymbol{x} - \boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\psi}_i + \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\psi}_i\right\}^2 \\ &= \|\boldsymbol{\mu}\|^2 + \sum_{i=2}^{r} \left\{(\boldsymbol{x} - \boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\psi}_i\right\}^2 \qquad (13) \\ &\approx \|\boldsymbol{\mu}\|^2 + \sum_{i=1}^{r'} \left\{(\boldsymbol{x} - \boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\phi}_i\right\}^2 \qquad (14) \end{aligned}$$

式 (14) の右辺第 1 項は平均ベクトルだけから成り，第 2 項はコアアフィン部分空間に関連した項である．$\boldsymbol{x} - \boldsymbol{\mu}$ も $\boldsymbol{\phi}_i$ も原点の位置に依存しないため，コアアフィン部分空間は原点の位置に依存しない．従って，第 1 項のみが原点の位置に依存する．

ところで，式 (14) を

$$\begin{aligned} d'(\boldsymbol{x}) &= -s(\boldsymbol{x}) \\ &= -\|\boldsymbol{\mu}\|^2 - \sum_{i=1}^{r'} \left\{(\boldsymbol{x} - \boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\phi}_i\right\}^2 \end{aligned} \tag{15}$$

のように表現すれば，式 (15) の右辺第 1 項は単なるバイアスであるため，式 (5) と類似しているといえる．このことは，$\frac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|}$ と $\{\boldsymbol{\phi}_i\}_1^{r'}$ が直交しており，$\|\boldsymbol{\mu}\|^2 > \lambda_1$ であり，かつ $\{\boldsymbol{\phi}_i\}_1^{r'}$ と $\{\boldsymbol{\psi}_i\}_2^{r}$ が張るアフィン部分空間が近似的に同一であるとみなせるならば，CLAFIC 法と投影距離法の認識性能は近くなる可能性があることを示唆している．

3.1.3　誤認識穴

式 (14) を用いて 2 クラス問題の識別境界の位置について考察する．2 クラスのうち，正解クラスの類似度が $s(\boldsymbol{x})$ であるとし，もう一方のクラスの類似度は s_{other} 以上であると仮定する(注2)．このようにおき，誤認識が起こる条件について考えてみると，

$$s(\boldsymbol{x}) \leqq s_{\mathrm{other}} \tag{16}$$

のときに誤認識が起こる．式 (16) に式 (14) を代入して変形す

(注2)：別のクラスの類似度は一定ではなく，位置によって変わるため，実際のところ，s_{other} は $\boldsymbol{x}$ に依存する．そのため，後述する誤認識穴の超球は形が歪むと考えられる．

れば，次式を満たす $\boldsymbol{x}$ は別のクラスと誤認識される．

$$\sum_{i=1}^{r'} \left\{ (\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\phi}_i \right\}^2 \leqq s_{\mathrm{other}} - ||\boldsymbol{\mu}||^2 \tag{17}$$

式 (17) によると，もし $s_{\mathrm{other}} - ||\boldsymbol{\mu}||^2 \geqq 0$ であるならば，アフィン部分空間上に識別境界が存在し，$\boldsymbol{\mu}$ を中心とし，半径 $\sqrt{s_{\mathrm{other}} - ||\boldsymbol{\mu}||^2}$ の超球の内側は誤認識される (図 2 参照)．この超球を「誤認識穴」と名付ける．従って，$||\boldsymbol{\mu}||$ が小さい場合は，平均ベクトルの近傍に存在するテストサンプルが誤認識される可能性がある．これを避けるためには，正解クラスの類似度 $s(\boldsymbol{x})$ を十分大きくする目的で，$||\boldsymbol{\mu}||$ が十分大きいほうが良い．このことは原点の位置が各クラスから離れているほうが良いことを示唆している．

3.2 より一般的な場合 (仮定 B)

3.1 節では，式 (9) において平均ベクトルと共分散行列 $\boldsymbol{\Sigma}$ の固有ベクトルの直交性等を仮定することで，正規化した平均ベクトルを $\boldsymbol{R}$ の第 1 固有ベクトルであると解釈した．ここではそのような仮定を行わずに，平均ベクトルの一部を固有ベクトルであると解釈する．本節の議論は 3.1 節の議論に沿っている．

3.2.1 固有ベクトルとしての平均ベクトル

共分散行列 $\boldsymbol{\Sigma}$ の固有ベクトル $\{\phi_i\}$ を用いると $\boldsymbol{\mu}$ は

$$\boldsymbol{\mu} = \sum_{i=1}^{d} \left(\boldsymbol{\mu}^{\mathrm{T}} \boldsymbol{\phi}_i \right) \boldsymbol{\phi}_i \tag{18}$$

のように分解できる．ここで $\boldsymbol{\nu}$ と $\boldsymbol{\eta}$ を

$$\boldsymbol{\nu} = \sum_{i=r'+1}^{d} \left(\boldsymbol{\mu}^{\mathrm{T}} \boldsymbol{\phi}_i \right) \boldsymbol{\phi}_i \tag{19}$$

$$\boldsymbol{\eta} = \sum_{i=1}^{r'} \left(\boldsymbol{\mu}^{\mathrm{T}} \boldsymbol{\phi}_i \right) \boldsymbol{\phi}_i \tag{20}$$

のように定義すると

$$\boldsymbol{\mu} = \boldsymbol{\nu} + \boldsymbol{\eta} \tag{21}$$

であることから，$\boldsymbol{\mu}$ を図 3 のように，共分散行列 $\boldsymbol{\Sigma}$ の第 1 から第 r' 固有ベクトルまでが張る部分空間上の成分 $\boldsymbol{\eta}$ と，その部分空間と直交する成分 $\boldsymbol{\nu}$ に分解したことになる．このようにすれば，$\frac{\boldsymbol{\nu}}{\|\boldsymbol{\nu}\|}$ と $\{\boldsymbol{\phi}_i\}_1^{r'}$ は直交する．

さらに，$\boldsymbol{R}$ の第 2 から第 r 固有ベクトル ($\{\boldsymbol{\psi}_i\}_2^r$) が張るアフィン部分空間と，$\boldsymbol{\Sigma}$ の第 1 から第 r' 固有ベクトル ($\{\boldsymbol{\phi}_i\}_1^{r'}$) が張るアフィン部分空間が近似的に同一とみなせるのであれば，$\boldsymbol{\mu}$ が部分空間と直交していない場合においても $\frac{\boldsymbol{\nu}}{\|\boldsymbol{\nu}\|}$ を $\boldsymbol{R}$ の固有ベクトルとみなすことができる．

3.2.2 コアアフィン部分空間

CLAFIC 法の類似度の式 $s(\boldsymbol{x})$ が

$$s(\boldsymbol{x}) \approx ||\boldsymbol{\nu}||^2 + \sum_{i=1}^{r'} \left\{ (\boldsymbol{x}-\boldsymbol{\nu})^{\mathrm{T}} \boldsymbol{\phi}_i \right\}^2 \tag{22}$$

となるため，この場合のコアアフィン部分空間は，3.1 節のコアアフィン部分空間の $\frac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|}$ と $\{\boldsymbol{\phi}_i\}_1^{r'}$ を $\frac{\boldsymbol{\nu}}{\|\boldsymbol{\nu}\|}$ と $\{\boldsymbol{\phi}_i\}_1^{r'}$ に置き換えたものになる．

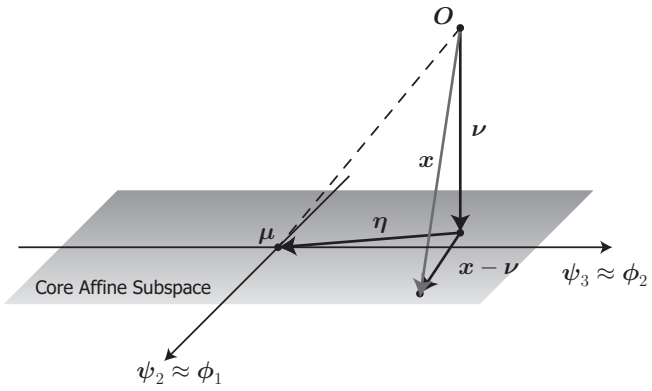


図 3 Core affine subspace in the CLAFIC method in the general case.

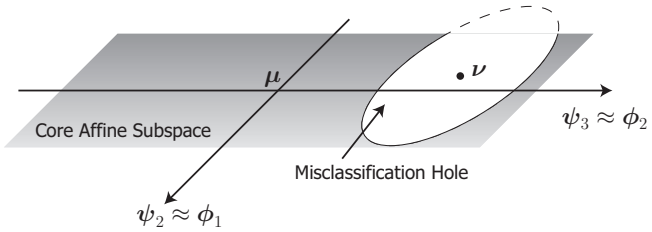


図 4 The region that a test sample is misclassified only with the core affine subspace.

また，式 (22) を式 (15) と同様に

$$\begin{aligned} d'(\boldsymbol{x}) &= -s(\boldsymbol{x}) \\ &= -||\boldsymbol{\nu}||^2 - \sum_{i=1}^{r'} \left\{ (\boldsymbol{x}-\boldsymbol{\nu})^{\mathrm{T}} \boldsymbol{\phi}_i \right\}^2 \end{aligned} \tag{23}$$

のように表現すれば，$\boldsymbol{\mu}$ と $\boldsymbol{\nu}$ の違いはあるものの，投影距離法の式 (5) との類似性が示唆される．

3.2.3 誤認識穴

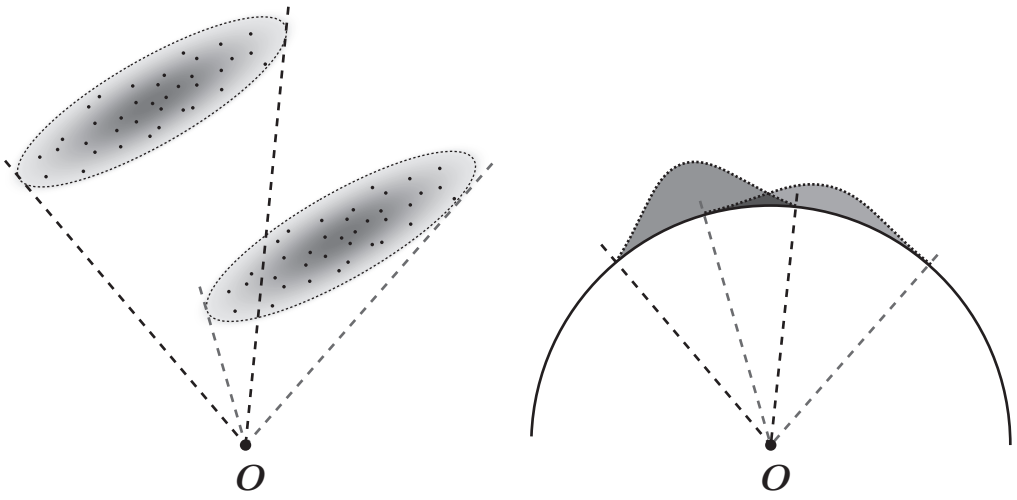
式 (22) を用いて 2 クラス問題における識別境界の位置について考える．3.1.3 節と同様に，2 クラスのうち，正解クラスの類似度が $s(\boldsymbol{x})$ であるとし，もう一方のクラスの類似度は s_{other} 以上であると仮定する．

$$s(\boldsymbol{x}) \leqq s_{\mathrm{other}} \tag{24}$$

を満たすときに誤認識が起こるので，式 (24) に式 (22) を代入して変形すれば，次式を満たす $\boldsymbol{x}$ は別のクラスと誤認識される．

$$\sum_{i=1}^{r'} \left\{ (\boldsymbol{x}-\boldsymbol{\nu})^{\mathrm{T}} \boldsymbol{\phi}_i \right\}^2 \leqq s_{\mathrm{other}} - ||\boldsymbol{\nu}||^2 \tag{25}$$

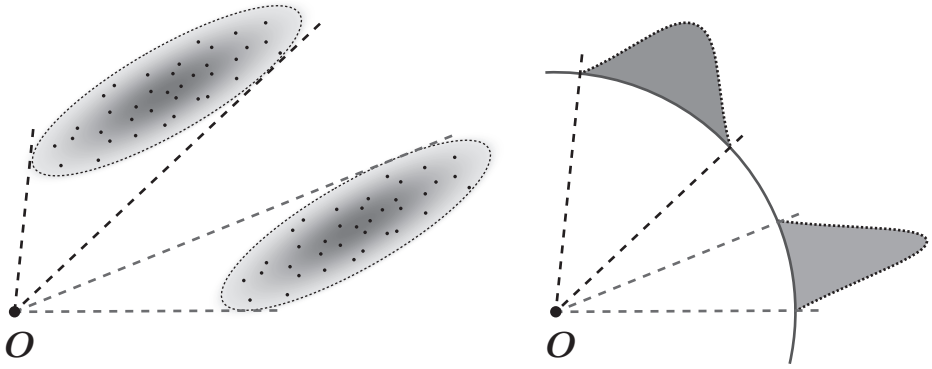
式 (25) はアフィン部分空間上の $\boldsymbol{\nu}$ を中心とし，半径 $\sqrt{s_{\mathrm{other}} - ||\boldsymbol{\nu}||^2}$ の超球の内側を表している (図 4 参照)．従って，$||\boldsymbol{\nu}||$ が小さい場合は，アフィン部分空間付近のテストサンプルが誤認識される可能性がある．式 (25) は式 (17) の $\boldsymbol{\mu}$ を単に $\boldsymbol{\nu}$ に置き換えただけであるが，以下のような重要な意味を持つ．(1) $||\boldsymbol{\nu}|| \leqq ||\boldsymbol{\mu}||$ より，誤認識穴の半径は仮定 A (3.1.3 節)



(a) Feature vectors are not normalized.

(b) Feature vectors are normalized so that their norms to be 1 .

図 5　The position of the origin that the distributions overlap.



(a) Feature vectors are not normalized.

(b) Feature vectors are normalized so that their norms to be 1.

図 6　The position of the origin that the distributions do not overlap.

のときよりも大きくなるため，アフィン部分空間上のテストサンプルが誤認識されてしまう可能性が高い．その一方で，(2) 誤認識穴の中心はパターンの分布の中心である $\boldsymbol{\mu}$ から離れる．このとき，誤認識穴の中心と分布の中心の距離は $||\boldsymbol{\eta}||$ である．つまり，誤認識穴がどれだけ大きくても，誤認識穴がパターンのほとんど存在しない領域にある限り，実際には誤認識は起こらない．

従って，誤認識を避けるには，次の条件の両方もしくはいずれかを満たす必要がある．(a) 誤認識穴の半径が小さくなるように，$||\boldsymbol{\nu}||$ を十分に大きくする．(b) $||\boldsymbol{\nu}||$ を平均ベクトルから十分離す．この場合，$\boldsymbol{\nu}$ と $\boldsymbol{\mu}$ の距離 (すなわち，$||\boldsymbol{\eta}||$) は少なくとも半径 $\sqrt{s_{\mathrm{other}} - ||\boldsymbol{\nu}||^2}$ よりも大きくなければならない．

4. 妥当な原点の位置に関する考察と実験

本節では，3.1 節で述べた，平均ベクトルと自己相関行列の第 1 固有ベクトルの方向が近似的に等しい場合について考える．

4.1 高精度な認識のために原点が満たすべき条件

4.1.1 原点から見たクラスの分布の方向

原点から見た分布の方向は識別能力に影響する．例えば，図 5 では，原点は 2 クラスの分布が重なる位置にあり，誤認識の原因となる．特徴ベクトルを正規化した場合も同様である．一方，図 6 では，原点は分布が重ならない位置にあり，誤認識が起こらない．そのため，分布が重なるかどうかは認識性能を左右する重要な要素である．

4.1.2 原点とクラスの分布の距離

3.1.3 節で述べたように，$||\boldsymbol{\mu}||$ が小さい場合には平均ベクトル付近のテストサンプルは誤認識される可能性がある．誤認識穴の大きさは $||\boldsymbol{\mu}||$ の大きさで決まるため，誤認識を避けるためには $||\boldsymbol{\mu}||$ が大きい必要がある．従って，原点の位置はクラスの分布と離れているほうがよい．以後，クラスの分布を表す

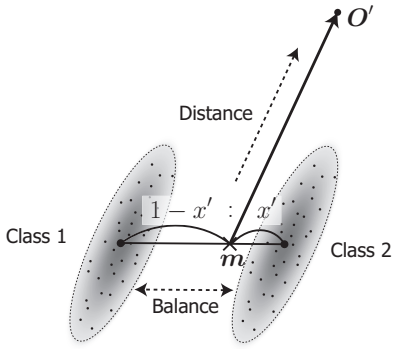


図 7 “Balance” and “distance”.

アフィン部分空間と原点との距離を括弧をつけて「距離」と表記することにする.

4.1.3 距離のバランス

原点と 2 クラスの分布との距離のバランス (例えば，識別境界が 2 本の平均ベクトルの中点を通っている場合や，片方のクラスに寄っている場合など) も識別境界の位置を変更するため，認識性能を決定する重要な要素である．仮に片方のクラスの分布と原点との「距離」が大きく，もう片方の「距離」が小さい場合は，式 (14) の右辺第 1 項がクラスによって大きく偏るため，CLAFIC 法の類似度にも偏りが生じる．結果として，テストサンプルは「距離」が大きいクラスに分類され易くなる．以後，2 クラスの平均ベクトルと原点との「距離」のバランスを括弧をつけて「バランス」と表記することにする.

4.2 適切な「距離」と「バランス」の調査

原点の位置が認識性能に及ぼす効果を調査するために，図 7 に示す「距離」と「バランス」を変更しながら，そのときの認識率を調査した．図 7 の $\boldsymbol{O}'$ は移動後の原点を表し，$\boldsymbol{m}$ は 2 クラスの平均ベクトルの内分点，x' は内分比を表す．つまり，$\boldsymbol{m}$ は次式で与えられる.

$$\boldsymbol{m} = (1-x')\boldsymbol{\mu}_1 + x'\boldsymbol{\mu}_2 \tag{26}$$

調査用データとして，等方性正規分布に従う人工サンプルを 3 クラス分作成した．3 クラスの分布パラメータは，クラス 1 の平均ベクトルを $\boldsymbol{\mu}_1 = (1, 1, \cdots, 1)$，クラス 2 と 3 の平均ベクトルを $\boldsymbol{\mu}_2 = \boldsymbol{\mu}_3 = (-1, -1, \cdots, -1)$ とし，クラス 1 と 2 の分散を 1，クラス 3 の分散を 0.1 とした．そして，3 つのクラスから，クラス 1 とクラス 2，クラス 1 とクラス 3 という 2 種類の 2 クラス問題を作成した.

実験では特定の「距離」と「バランス」になるように原点を移動し，原点移動後は通常の CLAFIC 法で認識を行った．少数サンプル問題 [13] を避けるために，学習用のサンプル数を 100000 とし，テスト用のサンプル数を 1000 とした．学習用とテスト用のサンプルは別である．部分空間の次元数を表す式 (2) のパラメータ r は全てを試し，最も高い認識率のみを採用した.「距離」と「バランス」のパラメータを固定し，同条件で 100 回ずつ実験を行ったので，最終的な認識率としては，得られた最高の認識率 100 個の平均値を採用した.

図 8 が実験結果である.「バランス」を表す図の横軸は，x' ではなく，$x = 2x' - 1$ で定義される値 x を用いている．つまり，$\boldsymbol{m}$ は，$x = -1$ の場合はクラス 1 の平均ベクトルの位置に，$x = 1$ の場合はクラス 1 の平均ベクトルの位置に，$x = 0$ の場合は中点にある．図中の 1 から 1000 の数字は，$\|\boldsymbol{O}' - \boldsymbol{m}\|$ で与えられる「距離」を表す.

クラス 1 とクラス 2 の認識率を図 8(a) と図 8(b) に示す．図 8(a) は特徴ベクトルをそのまま用いた場合，図 8(b) はあらかじめ特徴ベクトルのノルムを 1 に正規化した場合の結果である．図から次の 3 つのことがわかった．(1) 「距離」が増加すると，認識率は単調増加した．(2) 認識率は原点を 2 クラスの平均ベクトルの中点 (つまり，$x = 0$，$x' = 0.5$) としたときに最高になった．(3) $\boldsymbol{m}$ が中点から離れると，認識率は単調減少した．ただし (3) に関して，距離が十分大きければ「バランス」は認識率にさほど影響を与えないともいえる．上記の傾向は図 8(a) と図 8(b) の両方で確認できた.

同様に，クラス 1 とクラス 3 の認識率を図 8(c) と図 8(d) に示す．図 8(c) は特徴ベクトルをそのまま用いた場合，図 8(d) はあらかじめ特徴ベクトルのノルムを 1 に正規化した場合の結果である．図 8(c)，図 8(d) を図 8(a)，図 8(b) と比べると，少しだけ左に傾いていることがわかる．これは分散の差が原因であると考えられる．それ以外の傾向は図 8(a)，図 8(b) と同じであった.

図 8 より，特徴ベクトルが正規化されている，されていないにかかわらず,「距離」を大きくして，2 クラスの平均ベクトルの中点を用いるのが無難な方策であると考えられる.

4.3 Adaptive origin subspace method

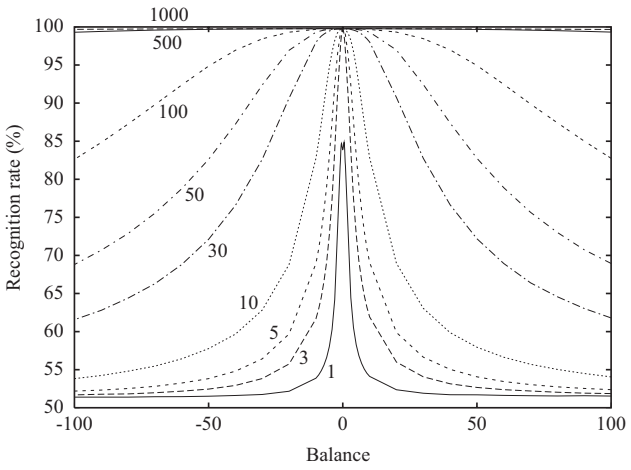
本節では，4.2 節の調査を踏まえて，4.1 節で示した条件を満たす認識に妥当な原点の位置を定める方法を提案する．この手法を “adaptive origin subspace method” と呼ぶことにする．提案手法の手順を Algorithm 1 に示し，図 9 に図示する.

提案手法では，まず 2 本の平均ベクトルの中点 $\boldsymbol{m}_{\mathrm{mid}}$ を求める.「バランス」を保つために，識別境界はこの点を通る．次に，Fisher の線形判別分析 [14] を用いて識別境界の向きを決定する．Fisher の線形判別分析では，2 クラスを最もよく分離する軸が得られるので，この軸と直交するベクトルを 1 つ選び，$\boldsymbol{n}$ とおく．つまり，$\boldsymbol{m}_{\mathrm{mid}}$ を起点とし，ベクトル $\boldsymbol{n}$ を含む超平面を識別境界とする．この条件を満たす $\boldsymbol{n}$ は一意には定まらないため，適当に選ぶことにする．最後に，式 (27) を用いて，クラスの分布を表すアフィン部分空間と原点との「距離」を十分大きくする位置に原点を定める(注3)

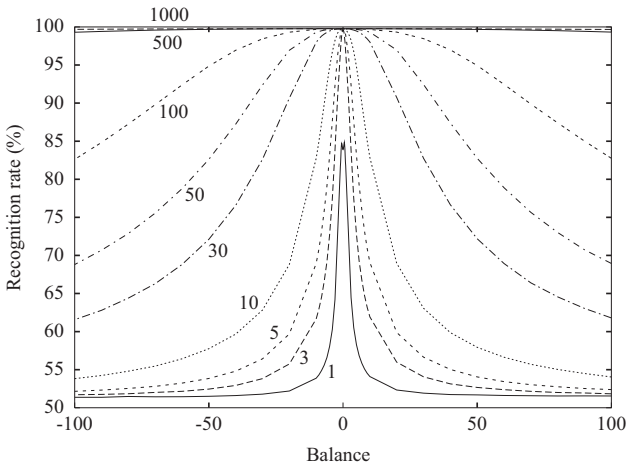
4.4 文字認識実験

提案手法の有効性を確認するため，提案手法を 45 組の 2 クラス問題に適用した．特徴ベクトルとしては，NIST Special Database 19 [15] の数字サンプルを 64×64 の大きさに非線形正規化 [16] した後，196 次元の方向線素特徴量 [17] を抽出して用いた.

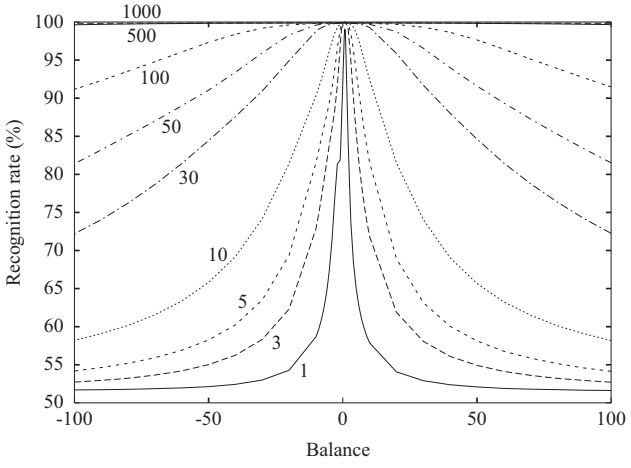
(注3)：Algorithm 1 では，実際にはアフィン部分空間と原点との「距離」ではなく，平均ベクトルと原点との距離を大きくしている．ただし，平均ベクトルとアフィン部分空間は直交していると仮定しているため，問題はないと思われる.



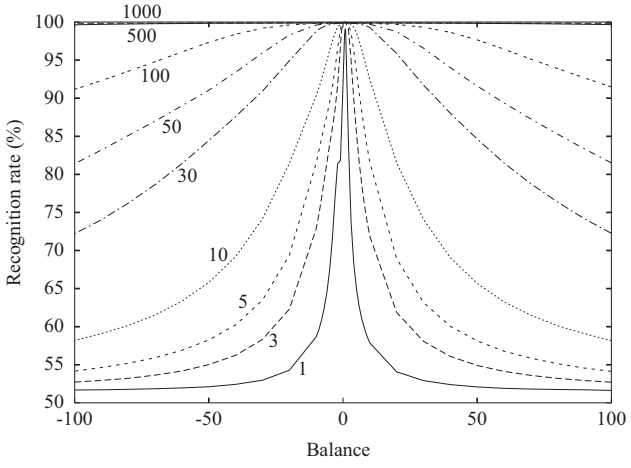
(a) Class 1 vs Class 2.



(b) Class 1 vs Class 2 with normalized feature vectors.



(c) Class 1 vs Class 3.



(d) Class 1 vs Class 3 with normalized feature vectors.

図 8　Effect of “distance” and “balance”.

Algorithm 1 Appropriate position of the origin

1: Calculate the midpoint of two mean vectors (denoted as $\boldsymbol{m}_{\mathrm{mid}}$).
2: Execute Fisher’s linear discriminant analysis.
3: Calculate a vector orthonormal to Fisher’s linear discriminator (denoted as $\boldsymbol{n}$).
4: An appropriate origin $\boldsymbol{O}'$ is given by

$$\boldsymbol{O}' = D\boldsymbol{n} + \boldsymbol{m}_{\mathrm{mid}}, \tag{27}$$

where D is the distance.

4.2 節と同様に，式 (2) のパラメータ r を変えながら最も高い認識率を選ぶ実験を 100 回行い，得られた最高の認識率 100 個の平均値を用いた．学習用のサンプル数を 10000 とし，テスト用のサンプル数を 1000 とした．学習用とテスト用のサンプルは別である．原点の位置は「距離」が 10000 になるようにした．

認識実験を行う前に，用いた特徴ベクトルが仮定 A (3.1 節)

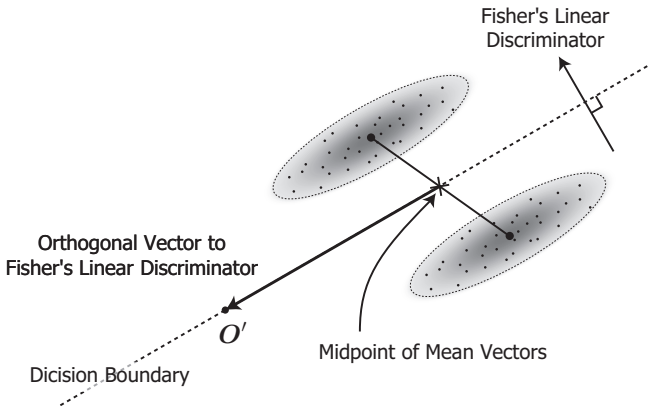


図 9　Appropriate position of the origin for subspace method (the proposed method).

で仮定した 2 つの条件

（a）　$\psi_1 \approx \frac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|}$

（b）　$\|\boldsymbol{\mu}\|^2 > \lambda_1$

表 1　Examinations of recognition problems whether “condition A” is satisfied. The angle between $\boldsymbol{\psi}_1$ and $\boldsymbol{\mu}$ shown in (a) is calculated as $\cos^{-1}\left(\boldsymbol{\psi}_1 \cdot \frac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|}\right)$. Each cell consists of two smaller cells: the upper cell is the average and the lower one is the standard deviation.

			“0”	“1”	“2”	“3”	“4”	“5”	“6”	“7”	“8”	“9”
(a)	Angle (degree)	Ave.	10.2	5.67	11.9	10.5	9.20	8.25	11.5	10.9	11.8	8.97
		Std. ($\times 10^{-2}$)	5.7	3.6	3.7	11	4.9	7.5	12	2.0	5.0	3.2
(b)	$\|\|\boldsymbol{\mu}\|\|^2$	Ave.	91180	405400	69670	58000	45140	40180	89010	64730	62560	84930
		Std.	294	1810	212	182	129	61.2	243	98.7	146	476
	λ_1	Ave.	40210	171700	38090	21950	23020	12670	44680	33790	28430	33380
		Std.	230	588	252	266	171	89.0	429	97.1	178	187

を満たすかどうか調べた．学習用に用いる 10000 個のサンプルから計算した値を調査結果を表 1 に示す．(a) については，$\boldsymbol{\psi}_1$ と $\boldsymbol{\mu}$ が成す角度 $\cos^{-1}\left(\boldsymbol{\psi}_1 \cdot \frac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|}\right)$ を度を単位として求め，文字毎に 100 回の平均と分散を示した．角度はおおよそ，5 度〜10 度程度であった．196 次元という高次元であることを考えると，比較的 $\boldsymbol{\psi}_1$ と $\boldsymbol{\mu}$ の方向は近いといえる．(b) については，$\|\boldsymbol{\mu}\|^2$ と λ_1 をそれぞれ文字毎に求め，100 回の平均と分散を示した．どの文字においても $\|\boldsymbol{\mu}\|^2 > \lambda_1$ であることが確認できた．このことから，3.1 節で仮定した条件はおおよそ満たされていると考えられる(注4)．

認識結果を表 2 と表 3 に示す．表 2 は特徴ベクトルをそのまま用いた場合，表 3 はあらかじめ特徴ベクトルのノルムを 1 に正規化した場合の結果である．2 クラス問題の認識結果はそれぞれ 3 つに区切られている．一番上は原点を移動しない場合の認識率 (通常の CLAFIC 法)，真ん中が原点を移動した場合の認識率 (提案手法)，一番下が両手法の認識率の差で，提案手法の認識率から通常の CLAFIC 法の認識率を引いた値である．

結果をまとめると，特徴ベクトルを正規化しなかった場合は，29 組で認識率が向上し，15 組で認識率が減少し，1 組は認識率が変わらなかった．認識率は平均では 0.06%上昇した．認識率の改善効果が最も大きかったのは “4” と “9” の組で，0.66%上昇した．逆に認識率が最も減少したのは “1” と “2” の組で，0.07%減少した．特徴ベクトルを正規化した場合は，全 45 組で認識率が向上し，認識率は平均では 0.32%上昇した．認識率の改善効果が最も大きかったのは “4” と “9” の組で，1.50%上昇した．最も小さかったのは “0” と “6” の組で，0.02%上昇少した．

提案した “adaptive origin subspace method” を適用することで，平均認識率は僅かに上昇したが，有意な差であるとは言い難い．ただし，認識実験に用いた特徴ベクトルが 3.1 節で仮定した 2 つの条件をおおよそ満たしていることや，4.2 節で原点の位置が認識性能に影響を与えることを確認していることから，原点の位置は認識性能に影響を及ぼすものの，少なくとも今回用いたデータに関しては移動前の原点の位置で十分高精度な認識が可能であったとも考えられる．このように，これまで部分空間法で高精度な認識が報告されている認識対象では，程度の差こそあれ，もともと認識に有利な位置に原点が位置していることも考えられる．そのため，今後は部分空間法を適用したものの十分な認識性能が得られなかった認識対象に提案手法を適用することを考えている．

(注4)：$\{\boldsymbol{\phi}_i\}_1^{r'}$ と $\{\boldsymbol{\psi}_i\}_2^{r}$ が張るアフィン部分空間が近似的に同一であるという仮定に関しては未確認であり，今後の課題である．

5. ま と め

本稿では，部分空間法の認識性能，特に CLAFIC 法の認識性能に原点の位置が及ぼす影響について考察を行った．具体的には，2 つの条件を仮定して，CLAFIC 法の類似度の式を原点の位置に依存する項と依存しない項に分割した．そして，原点に依存する項が認識性能にどのような影響を及ぼすかについて考察した．原点の位置に依存しない項はクラスの分布を表す部分空間である「コアアフィン部分空間」に関連している．考察の結果，原点の位置によってはコアアフィン部分空間上に認識不可能な超円状の領域「誤認識穴」が発生し，誤認識が起こり得ることがわかった．

そこで，誤認識穴が発生しないように原点の位置が満たすべき条件を考察し，実験によって検証した．そして，そのような条件を満たすように原点を定める方法 “adaptive origin subspace method” を提案した．提案手法を文字認識に適用した結果，認識率はほんの僅かだけ上昇した．この原因として考えられるのは，原点がもともと認識に有効な位置にあったということである．このように，部分空間法で高精度な認識が報告されている認識対象では，もともと認識に有利な位置に原点が位置しているが多いと考えられる．そのため，提案手法を適用しても，必ずしも認識率が向上するとは限らない．そこで，文字以外の対象に提案手法を適用することとが今後必要になってくると思われる．また，3.2 節で導いた理論の検証も今後の課題である．

謝辞　本研究を進めるにあたり有益な情報を提供してくださるとともに，激励してくださった NTT データ株式会社の坂野鋭氏に感謝いたします．

文　献

[1] S. Watanabe, Knowing & guessing : a quantitative study of inference and information, John Wiley & Sons, Inc., 1969.
[2] 飯島泰蔵，パターン認識理論，森北出版，1989.
[3] 石井健一郎，上田修功，前田英作，村瀬洋，わかりやすいパターン認識，オーム社，東京，1998.
[4] E. Oja, Subspace methods of pattern recognition, Wiley, New York, 1983.
[5] 前田賢一，渡辺貞一，“局所的構造を導入したパターン・マッチング法，” 信学論 (D)，vol.J68-D, no.3，pp.345–352，Mar.,

表 2 Recognition rates of handwritten numerals in NIST Special Database 19. Each recognition result consists of three cells. The top cell is recognition rate without changing the origin (the conventional method). The middle one is that of changing the origin (the proposed method). The bottom one is the difference of these two recognition rates.

		“1”	“2”	“3”	“4”	“5”	“6”	“7”	“8”	“9”
“0”	Conv.	99.37	99.15	99.74	99.69	99.45	99.07	99.24	99.56	99.65
	Prop.	99.40	99.15	99.72	99.74	99.47	99.05	99.45	99.53	99.66
	Diff.	**0.03**	0	-0.02	**0.05**	**0.02**	-0.02	**0.21**	-0.03	**0.01**
“1”	Conv.		99.15	99.62	99.17	99.52	99.39	98.71	98.96	99.19
	Prop.		99.08	99.64	99.29	99.56	99.42	99.13	98.97	99.50
	Diff.		-0.07	**0.02**	**0.12**	**0.04**	**0.03**	**0.42**	**0.01**	**0.31**
“2”	Conv.			98.06	99.46	99.40	99.42	98.61	98.22	99.18
	Prop.			98.07	99.50	99.39	99.39	98.56	98.23	99.15
	Diff.			**0.01**	**0.04**	-0.01	-0.03	-0.05	**0.01**	-0.03
“3”	Conv.				99.72	97.90	99.59	99.14	97.73	99.09
	Prop.				99.77	97.86	99.58	99.29	97.67	99.13
	Diff.				**0.05**	-0.04	-0.01	**0.15**	-0.06	**0.04**
“4”	Conv.					99.65	99.55	98.79	99.23	97.94
	Prop.					99.63	99.65	98.97	99.24	98.60
	Diff.					-0.02	**0.10**	**0.18**	**0.01**	**0.66**
“5”	Conv.						97.58	99.73	97.95	99.55
	Prop.						97.65	99.78	98.04	99.53
	Diff.						**0.07**	**0.05**	**0.09**	-0.02
“6”	Conv.							99.84	99.06	99.81
	Prop.							99.85	99.07	99.84
	Diff.							**0.01**	**0.01**	**0.03**
“7”	Conv.								98.88	97.97
	Prop.								98.84	98.12
	Diff.								-0.04	**0.15**
“8”	Conv.									98.28
	Prop.									98.18
	Diff.									**0.10**

1985.

[6] 福井和広，山口修，鈴木薫，前田賢一，“制約相互部分空間法を用いた環境変動にロバストな顔画像 認識—照明変動の影響を抑える制約部分空間の学習—，” 信学論 (D-II)，vol.J82-D-II, no.4，pp.613–620，Apr.，1999.

[7] 津田宏治，“ヒルベルト空間における部分空間法，” 信学論 (D-II)，vol.J82-D-II, no.4，pp.592–599，Apr.，1999.

[8] 前田英作，村瀬洋，“カーネル非線形部分空間法によるパターン認識，” 信学論 (D-II)，vol.J82-D-II, no.4，pp.600–612，Apr.，1999.

[9] 坂野鋭，武川直樹，中村太一，“核非線形相互部分空間法による物体認識，” 信学論 (D-II)，vol.J84-D-II, no.8，pp.1549–1556，Aug.，2001.

[10] 石川則之，有木康雄，部分空間法による顔認識の比較実験，画像の認識・理解シンポジウム (MIRU’96)，第 II 巻，pp.145–150，July，1996.

[11] 池田正幸，田中英彦，元岡達，“手書き文字認識における投影距離法，” 情報処理学会論文誌，vol.24, no.1，pp.106–112，1983.

[12] 黒澤由明，“球面ガウス分布から導出される部分空間法，” 信学論 (D-II)，vol.J81-D, no.6，pp.1205–1212，June，1998.

[13] 岩村雅一，大町真一郎，阿曽弘具，“標本共分散行列の固有ベクトルを用いた真のマハラノビス距離の推定法，” 信学論 (D-II)，vol.J86-D-II, no.1，pp.22–31，Jan.，2003.

[14] R. O. Duda and P. E. Hart, Pattern classification and scene analysis, A Wiley-Interscience Publication, 114–118 pp., 1973.

[15] P. J. Grother, NIST special database 19 — handprinted forms and characters database, Technical report, National Institute of Standards and Technology, Mar., 1995.

[16] 山田博三，斉藤泰一，山本和彦，“線密度イコライゼーション—相関法のための非線形正規化法，” 信学論 (D)，vol.J67-D, no.11，pp.1379–1383，Nov.，1984.

[17] 孫寧，田原透，阿曽弘具，木村正行，“方向線素特徴量を用いた高精度文字認識，” 信学論 (D-II)，vol.J74-D-II, no.3，pp.330–339，Mar.，1991.

表 3 Recognition rates of handwritten numerals in NIST Special Database 19 with normalized feature vectors. Each recognition result consists of three cells. The top cell is recognition rate without changing the origin (the conventional method). The middle one is that of changing the origin (the proposed method). The bottom one is the difference of these two recognition rates.

		"1"	"2"	"3"	"4"	"5"	"6"	"7"	"8"	"9"
"0"	Conv.	99.30	99.05	99.60	99.52	99.09	99.02	98.91	99.35	99.51
	Prop.	99.40	99.15	99.72	99.73	99.48	99.04	99.46	99.54	99.66
	Diff.	**0.10**	**0.10**	**0.12**	**0.21**	**0.39**	**0.02**	**0.55**	**0.19**	**0.15**
"1"	Conv.		98.96	99.54	98.90	99.26	99.31	98.89	98.72	98.95
	Prop.		99.08	99.64	99.29	99.56	99.40	99.11	98.97	99.50
	Diff.		**0.12**	**0.10**	**0.39**	**0.30**	**0.09**	**0.22**	**0.25**	**0.55**
"2"	Conv.			97.81	99.31	99.22	99.03	97.90	97.78	98.96
	Prop.			98.08	99.49	99.40	99.39	98.58	98.22	99.15
	Diff.			**0.27**	**0.18**	**0.18**	**0.36**	**0.68**	**0.44**	**0.19**
"3"	Conv.				99.58	97.35	99.37	98.92	97.26	98.78
	Prop.				99.77	97.88	99.58	99.30	97.71	99.13
	Diff.				**0.19**	**0.53**	**0.21**	**0.38**	**0.45**	**0.35**
"4"	Conv.					99.55	99.33	98.28	98.89	97.09
	Prop.					99.63	99.65	98.96	99.24	98.59
	Diff.					**0.08**	**0.32**	**0.68**	**0.35**	**1.50**
"5"	Conv.						97.33	99.67	97.35	99.18
	Prop.						97.63	99.77	98.08	99.53
	Diff.						**0.30**	**0.10**	**0.73**	**0.35**
"6"	Conv.							99.73	98.70	99.63
	Prop.							99.85	99.09	99.85
	Diff.							**0.12**	**0.39**	**0.22**
"7"	Conv.								98.58	96.79
	Prop.								98.83	98.12
	Diff.								**0.25**	**0.33**
"8"	Conv.									97.86
	Prop.									98.19
	Diff.									**0.33**

差分部分空間を用いた混合マハラノビス関数に関する一考察

平山　淳一†　　中山　英久†　　加藤　　寧†

† 東北大学大学院情報科学研究科
〒 980-8579 宮城県仙台市青葉区荒巻字青葉 6-3-09
E-mail: hira@it.ecei.tohoku.ac.jp

あらまし　高精度な類似文字識別を実現するためには，2 つの類似クラスの差異が良く現れるような特徴空間へ特徴量ベクトルを射影することが有効である．類似文字識別に有効な識別関数として，混合マハラノビス関数 (CMF) が提案されている．これは特徴量を固有値の小さな固有ベクトルが張る部分空間に射影してマハラノビス距離の補正を行う手法である．しかし，CMF は類似クラスの差異情報として，2 つの類似クラスの平均特徴量ベクトルの差分を用いているが，一般に手書き文字の特徴量の分布は複雑な多次元分布をしていると考えられるため，1 本の差分ベクトルだけでクラスの差異を捉えるには限界がある．本稿では，2 つの部分空間の差を表す差分部分空間を導入した新しい CMF を提案し，手書き文字データベース ETL9B を用いた実験によりその有効性を示す．

キーワード　手書き文字認識，類似文字，混合マハラノビス関数，差分部分空間

A Study of Compound Mahalanobis Function Using Difference Subspace

Junichi HIRAYAMA†, Hidehisa NAKAYAMA†, and Nei KATO†

† Graduate School of Information Sciences, Tohoku University, aza-Aoba 6-3-09, Aramaki, Aoba-ku, Sendai-shi, Miyagi, 980-8579 Japan
E-mail: hira@it.ecei.tohoku.ac.jp

Abstract　In order to distingish similar characters, it is necessary to project feature vector to a feature space which includes difference between two similar categories. Compound Mahalanobis Function (CMF) has been proposed for high accuracy similar character identification, and modification of the Mahalanobis Distance projecting feature vector to eigenvectors with smaller eigenvalues. However, CMF uses single difference vector of two mean vectors as difference information between two similar categories, and handwritten character distribution is complicated multidimensional distribution. As a result, it is hard to recognize similar character with only this difference vector. In this paper, we propose a new CMF method using difference subspace, and verify effectiveness of the proposed method through experiment using ETL9B.

Key words　Handwritten Recognition,Similar Character,Compound Mahalanobis Function,Difference Subspace

1. はじめに

現在まで，手書き文字認識に関する多くの研究が行われており，高い認識精度が得られている．例えば，個別文字認識においては，手書き文字データベース ETL9B [1] を用いた実験で，99% を超える認識結果も報告されている [2], [3]．しかしながら，類似文字に対する認識率は依然として低く，手書き類似文字認識の高精度化が課題となっている．

類似文字認識の高精度化に関する研究では，混合部分空間法，混合投影距離法，混合マハラノビス関数など，2 つの類似クラス間の差異情報を利用する，混合識別関数が提案されている [4], [5]．混合識別関数は，類似文字間の差異情報として，認識対象字種と対立類似字種との平均特徴量ベクトルの差分を認識に用いる．そして，類似クラスの違いが固有値の小さい固有ベクトル方向に顕著に現れるという性質を利用して，学習用の特徴量ベクトルから得られる共分散行列の固有ベクトルのうち，その固有値の小さなものから適当な数を選択し，部分空間を構成する．例えば混合マハラノビス関数は，差分ベクトルをこの部分空間へ射影したベクトルを用いて，マハラノビス距離に補正を行う識別関数である．

実際の手書き文字の特徴量は複雑な多次元分布をしていると考えられるため，類似クラス間の差異情報として平均特徴量ベクトルの差分ベクトルを用いるだけで，類似文字の違いを正確に捉えるには限界がある．より高精度な類似文字認識を行うた

めには，類似クラスの差異を多次元の射影空間として捉えることが必要であると考えられる．

制約相互部分空間法 (CMSM : Constraint Mutual Subspace Method) は，制約部分空間と呼ばれる識別に有効なクラス間変動のみを含む空間への射影を相互部分空間法に付加した手法である．制約部分空間への射影は識別に有効な特徴抽出となり，照明変動などを含んだ顔画像認識において，その有効性が示されている [9], [11]．CMSM の適用に際して，具体的な制約部分空間をいかに生成するかが重要な課題であり，その 1 つとして，二つの部分空間の差異を表す差分部分空間が提案されている [9], [10]．各部分空間を差分部分空間に射影することで，部分空間同士のなす角が広がり，相互部分空間法などの角度に基づいた手法でその有効性が確認されている．

この差分部分空間を構成する基底ベクトルは，マハラノビス距離のような距離に基づいた手法においても，識別に有効であると考えられるが，これらについての検討はまだ行われていない．本稿では，未知の特徴量と標準特徴量との距離を差分部分空間に射影した項を混合マハラノビス関数の補正項に適用することで，類似文字認識の高精度化を行う．そして，ETL9B を用いた実験により本手法の有効性を示す．

以下，2. では従来の混合マハラノビス関数について述べ，3. では差分部分空間を用いてこれを改良する手法を提案する．4. で提案手法の有効性を確認するため，類似文字種を対象にした実験と結果，5. で全字種を対象にした実験と結果について述べる．6. は結論である．

2. 混合マハラノビス関数

類似文字認識の高精度化を行うために提案された混合マハラノビス関数 (CMF : Compound Mahalanobis Function) [4] は，2 つの類似クラスの平均特徴量の差分ベクトルを，2 つのクラスが良く分離できる部分空間に射影し，射影された差分ベクトルを用いてマハラノビス距離に補正を行う関数である．

認識対象クラス Ω_1 と対立類似クラス Ω_2 の平均特徴量ベクトルをそれぞれ $\boldsymbol{u}$，$\boldsymbol{v}$ とする．また，Ω_1 の d 次元学習用サンプルの共分散行列の固有値を

$$\begin{aligned} &\lambda_1, \lambda_2, \cdots, \lambda_d \\ &(\lambda_i \geqq \lambda_{i+1} \quad i = 1, 2, \cdots, d-1) \end{aligned} \tag{1}$$

とし，λ_i に対する固有ベクトルを $\boldsymbol{\phi}_i$ とする．

ここで，Ω_1 と Ω_2 の違いが $\boldsymbol{\phi}_k, \boldsymbol{\phi}_{k+1}, \cdots, \boldsymbol{\phi}_d$ により構成される部分空間に顕著に現れると仮定するとき，CMF は次式で定義される．

$$D_{cmf}(\boldsymbol{x}) = \sum_{i=0}^{d} \frac{\{\boldsymbol{\phi}_i^T(\boldsymbol{x}-\boldsymbol{u})\}^2}{\lambda_i} + \mu \sum_{i=k}^{d} \frac{(\boldsymbol{\phi}_i^T \boldsymbol{\delta})^2}{\lambda_i} \tag{2}$$

ただし，$\boldsymbol{x}$ は未知の特徴量ベクトル，μ は結合重み定数，$\boldsymbol{\delta}$ は，以下の式で与えられるベクトルである．

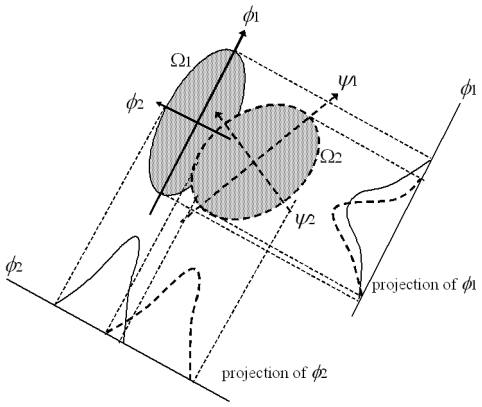


図 1　2 つのクラスの差

$$\begin{aligned} \boldsymbol{\delta} &= (\boldsymbol{\xi}^T(\boldsymbol{x}-\boldsymbol{u}))\boldsymbol{\xi} \\ \boldsymbol{\xi} &= \frac{\sum_{i=k}^{d} \boldsymbol{\phi}_i^T(\boldsymbol{u}-\boldsymbol{v})\boldsymbol{\phi}_i}{\sqrt{\sum_{i=k}^{d} (\boldsymbol{\phi}_i^T(\boldsymbol{u}-\boldsymbol{v}))^2}} \end{aligned} \tag{3}$$

$\boldsymbol{\xi}$ は，Ω_1 と Ω_2 の平均特徴量ベクトルの差分ベクトル $\boldsymbol{u}-\boldsymbol{v}$ を $\boldsymbol{\phi}_k, \cdots, \boldsymbol{\phi}_d$ で構成される部分空間に射影して，大きさを 1 に正規化したベクトルであり，$\boldsymbol{\delta}$ は未知の特徴量と平均特徴量の距離 $\boldsymbol{x}-\boldsymbol{u}$ を $\boldsymbol{\xi}$ 方向に射影して得られるベクトルである．すなわち CMF は，$\boldsymbol{\delta}$ の長さをマハラノビス距離に換算し，結合重み係数 μ によりマハラノビス距離との線形結合をとったものである．

ここで，手書き文字認識においてはマハラノビス距離よりも，固有値にバイアスを加えた改良型マハラノビス距離 [8] の方が識別に有効であることがわかっている．よって，式 (2) において固有値にバイアスを加えた次式を CMF と再定義する．

$$D_{cmf}(\boldsymbol{x}) = \sum_{i=0}^{d} \frac{\{\boldsymbol{\phi}_i^T(\boldsymbol{x}-\boldsymbol{u})\}^2}{\lambda_i + b} + \mu \sum_{i=k}^{d} \frac{(\boldsymbol{\phi}_i^T \boldsymbol{\delta})^2}{\lambda_i + b} \tag{4}$$

本稿では，式 (4) のバイアス b として全クラス全固有値の平均を使用する．

3. 混合マハラノビス関数の差分部分空間による拡張

類似文字識別を行う場合，類似クラスの差異成分を強調するような特徴空間への射影が有効である．CMF では 2 つのクラスの差異情報として平均特徴量ベクトルの差分ベクトルを利用しているが，一般に手書き文字の特徴量は複雑な多次元分布をしていると考えられるため，差分ベクトルだけで類似クラスの差異を捉えるには限界がある．より高精度な類似文字認識を行うためには，クラスの差異を射影空間全体で捉えることが必要である．本章では，部分空間同士の差異を表す射影空間として差分部分空間 [10] を導入し，これを用いて混合マハラノビス関数を拡張する．

3.1 類似クラス識別に関する考察

図 1 に示すように，ある類似クラス Ω_1,Ω_2 が存在し，それぞ

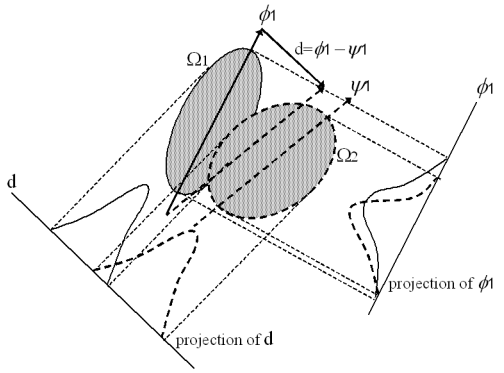
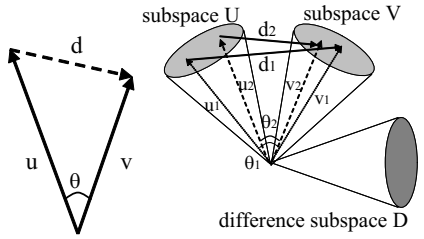


図 2　差分部分空間への射影



(a) 1 次元の場合　　(b) 多次元の場合

図 3　差分部分空間の概念

れの部分空間を張る固有ベクトルが $\boldsymbol{\phi}_1, \boldsymbol{\phi}_2$ 及び $\boldsymbol{\psi}_1, \boldsymbol{\psi}_2$ であるとする．このとき，固有値の大きな固有ベクトルである $\boldsymbol{\phi}_1$(楕円の長軸方向) への射影では，分布の重なりが大きく有効な識別ができない．$\boldsymbol{\psi}_1$ についても同様である．一方，固有値の小さな固有ベクトル $\boldsymbol{\phi}_2$(楕円の短軸方向) への射影により有効な識別が可能になる．CMF は，このように類似クラスの違いが固有値の小さな固有ベクトル方向に顕著に現れるという性質を利用した手法である．

次に，図 2 のように $\boldsymbol{\phi}_1$,$\boldsymbol{\psi}_1$ の差分ベクトル $\boldsymbol{d}$ の方向へ射影することを考える．$\boldsymbol{d}$ 方向への射影によっても分布が分離し，有効な識別が可能である．固有値の大きな固有ベクトルはクラスの分布に沿った軸であると考えることができ，それらの差分ベクトルは部分空間の分布の差を考慮した有効な識別方向になっている．このことは，一般に d 次元特徴量についても同様に考えることができる．

3.2　差分部分空間の概念

差分部分空間 $\boldsymbol{D}$ は，図 3 のように 2 つのベクトルの差分ベクトルを多次元に拡張したものになっている．

2 つの N 次元部分空間 $\boldsymbol{U}, \boldsymbol{V}$ に対して，N 個の正準角 $\theta_i (i = 1 \sim N)$ が定義できる．ここで i 番目に小さな正準角 θ_i を形成する 2 つのベクトル $\boldsymbol{u}_i \in \boldsymbol{U}$ と $\boldsymbol{v}_i \in \boldsymbol{V}$ の差分ベクトルを $\boldsymbol{d}_i = \boldsymbol{u}_i - \boldsymbol{v}_i$ とする．差分部分空間は差分ベクトル $\boldsymbol{d}_i$ を基底とする空間であると定義される [9]．この差分部分空間はクラスの差に基づいた識別に有効な射影空間となっている．

3.3　射影行列による差分部分空間の生成

福井らは文献 [10] において，差分部分空間を 2 つの部分空間に対する射影行列の和の固有ベクトルから生成する手法を提案した．

類似クラス Ω_1，Ω_2 に対する射影行列を $\boldsymbol{P}$,$\boldsymbol{Q}$, その和を $\boldsymbol{G}$ とする．射影行列は以下の式で定義される．

$$\begin{aligned} \boldsymbol{P} &= \sum_{i=1}^{N_b} \boldsymbol{\phi}_i \boldsymbol{\phi}_i^T \\ \boldsymbol{Q} &= \sum_{i=1}^{N_b} \boldsymbol{\psi}_i \boldsymbol{\psi}_i^T \end{aligned} \tag{5}$$

$$\boldsymbol{G} = \boldsymbol{P} + \boldsymbol{Q} \tag{6}$$

ここで $\boldsymbol{\phi}_i$,$\boldsymbol{\psi}_i$ は，それぞれ Ω_1,Ω_2 の部分空間を張る固有ベクトルである．差分部分空間は射影行列の和 $\boldsymbol{G}$ について，式 (7) の固有値問題を解くことで得られる．

$$\boldsymbol{G}\boldsymbol{a} = \lambda \boldsymbol{a} \tag{7}$$

これら一連の流れは，類似クラスの固有ベクトルをデータとした主成分分析であると解釈できる．式 (7) の $\boldsymbol{G}$ の $(N_b \times 2)$ 本の固有ベクトルのうち，固有値の大きな N_p 本が 2 つのクラスの共通成分を表す空間，残りの固有値の小さな N_c 本が差異成分を表す空間と考えることできる $(N_p + N_c = N_b \times 2)$．すなわち，クラス Ω_1 と Ω_2 の差分部分空間 $\boldsymbol{D}$ は $\boldsymbol{G}$ の固有ベクトルのうち，固有値の小さな方から選んだ N_c 本を基底とする空間となる．

$$\boldsymbol{D} = (\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_{N_c}) \tag{8}$$

なお，N_b,N_c の値は実験的に定めるものとする．

3.4　差分部分空間による混合マハラノビス関数の拡張

3.3. で求めた差分部分空間上でのマハラノビス距離を次式で計算する．

$$\sum_{i=0}^{N_c} \frac{\{\boldsymbol{d}_i^T (\boldsymbol{x} - \boldsymbol{u})\}^2}{\lambda_i^d} \tag{9}$$

但し，差分部分空間を張る基底ベクトル $\boldsymbol{d}_i$ 上でのデータの分散 λ_i^d は本来，主成分分析により固有ベクトルに対する固有値として与えられるが，この場合，式 (7) の固有値問題からは直接得られない．そこで，学習用サンプル $\{\boldsymbol{x}_1^l, \boldsymbol{x}_2^l, \cdots, \boldsymbol{x}_N^l\}$ を差分部分空間をなす各固有ベクトルに射影することによって，データの分散を計算する．

$$\begin{aligned} u_i^l &= \frac{1}{N} \sum_{k=0}^{N} \boldsymbol{d}_i^T \boldsymbol{x}_k^l \\ \lambda_i^d &= \frac{1}{N} \sum_{k=0}^{N} (\boldsymbol{d}_i^T \boldsymbol{x}_k^l - u_i^l)^2 \end{aligned} \tag{10}$$

式 (4) と同様に補正項として，結合重み係数 ν を用いて改良型マハラノビス距離との線形結合をとった式 (11) が本稿で提案する差分部分空間を用いた混合マハラノビス関数 (DS-CMF:Difference Subspace based CMF) である．

$$D_{dscmf}(\boldsymbol{x}) = \sum_{i=0}^{d} \frac{\{\boldsymbol{\phi}_i^T (\boldsymbol{x} - \boldsymbol{u})\}^2}{\lambda_i + b} + \nu \sum_{i=0}^{N_c} \frac{\{\boldsymbol{d}_i^T (\boldsymbol{x} - \boldsymbol{u})\}^2}{\lambda_i^d + b^d} \tag{11}$$

表 1　類似字種セット

1	え	之	6	熊	態	11	伸	仲
2	ば	ぱ	7	采	釆	12	推	椎
3	び	ぴ	8	土	土	13	磨	麿
4	穎	頴	9	師	帥	14	末	末
5	干	千	10	情	楕	15	肋	助

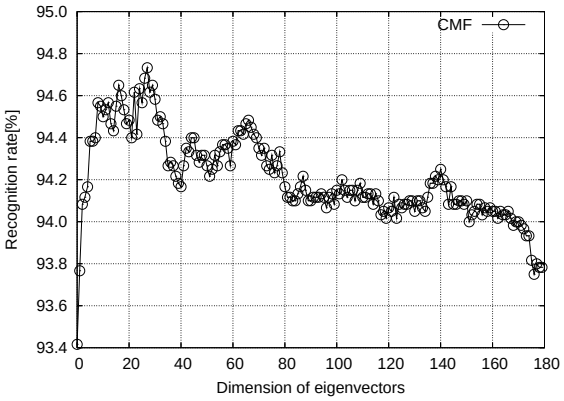


図 4　CMF

4. 類似文字実験

提案手法である DS-CMF が類似文字識別に有効であることを確認するために，CMF との比較実験を行った．認識対象は手書き文字データベース ETL9B に含まれる字種のうち，表 1 に示す 15 組 30 字種である．これらは文献 [6], [7] で用いられている誤認識が多く発生する類似文字対である．

前処理として，各サンプル文字画像に対してノイズ除去，スムージング，非線形正規化を行った．特徴抽出には 196 次元改良型方向線素特徴量 [12]，学習・未知データの分割には 10 分割ローテーション法を用いた．これは，ETL9B 各字種 200 サンプルのうち，180 個を学習データ，残りの 20 個を未知データとするセットを 10 パターン (A–J とする) 作成し，10 セットの結果の平均を最終的な認識結果とする方法である．また，学習データが 180 個であるため，共分散行列の固有ベクトルは 179 本しか求まらない．そのため，本実験では式 (4),(11) の計算を $d = 179$ として行う．

図 4 は CMF のパラメータ k を変化させたときの，図 5 は提案手法のパラメータ N_c を変化させたときの認識率の変動を表したグラフである．その他のパラメータは，実験により認識率が最も高かった $\mu = 0.7$，$\nu = 0.7$，$N_b = 31$ を使用した．図 5 において認識率がある次元数を境に急激に低下していることが確認できる．このことは，式 (7) の固有値問題で得られる固有ベクトルのうち，固有値の小さなものがクラスの差を表すベクトル，固有値の大きなものがクラスの共通成分を表すベクトルになっていることを表している．

図 6 は表 1 に示した 1 ～ 15 番の類似文字対に対して，各々の認識手法を用いて実験を行い，認識率をグラフにしたものである．参考のために改良型マハラノビス距離 (MMD:Modified Mahalanobis Distance) [8] を用いて認識を行った結果も一緒に掲載している．この結果より，CMF と提案手法である DS-CMF

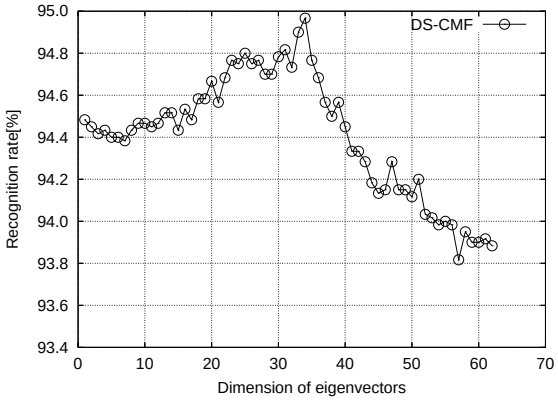


図 5　DS-CMF

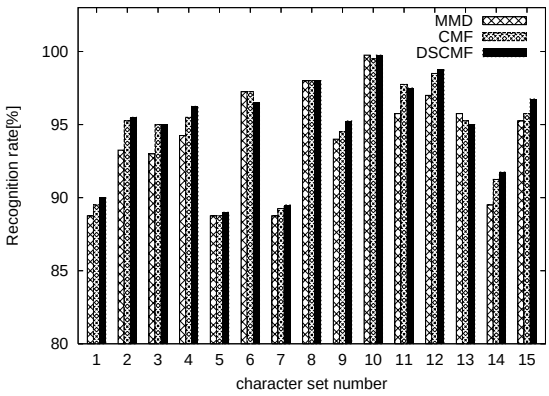


図 6　字種対ごとの認識率

表 2　最大認識率

CMF	94.73 %
DS-CMF	94.96 %

ともに MMD に比べて全ての字種対で認識率の向上が確認される．つまり補正項の付加により，認識精度が向上することを示している．さらに多くの類似字種対において，CMF よりも DS-CMF の方が認識率が高くなり，CMF で構成される射影空間よりも DS-CMF で構成される差分部分空間の方が識別に有効な特徴空間であると言える．

それぞれの最大認識率は表 2 のようになり，提案手法により認識率が改善されることが確認できた．

5. 全字種実験

ETL9B に含まれる 3036 字全字種を対象にした認識実験を行った．計算時間を考慮し，識別を以下に説明する大分類，細分類，再細分類の 3 段階に分けて行った．これらは文献 [4], [6] で使用されている識別方法である．

a) 大　分　類

シティブロック距離を用いて全字種を対象に識別を行い，各未知データごとに候補を上位 M 字種に絞りこむ．

b) 細　分　類

大分類で得られた候補 M 字種に対して改良型マハラノビス距離を計算する．

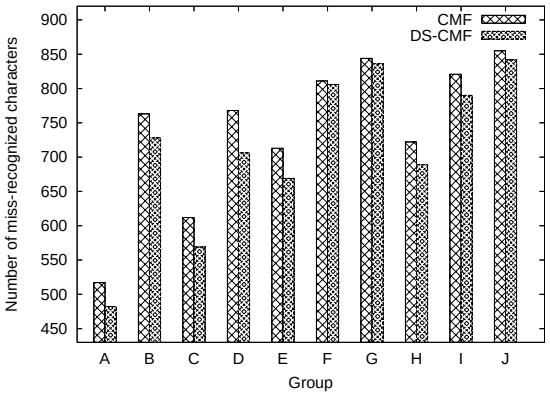


図 7　ETL9B 全字種に対する認識結果

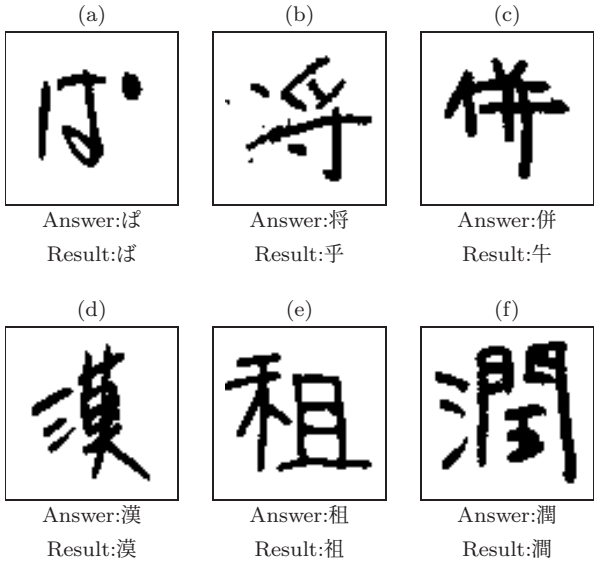


図 8　誤認識された文字の例

c) 再細分類

細分類の結果得られた未知データと n 位候補辞書データの距離を d_n としたとき，

$$d_n - d_1 \leqq \Delta \tag{12}$$

となる全ての候補に対して，CMF 及び提案手法である DS-CMF を用いて，2 クラス識別を行う．本実験では $M = 30$，$\Delta = 10.0$ とした．また，CMF のパラメータは $k = 98$，$\mu = 3.0$，提案手法のパラメータは $N_b = 26$，$N_c = 25$，$\nu = 1.0$ とした．

図 7 は，各字種ごとに用意した 10 セットの学習・未知データ分割パターン (A–J) において，誤認識された文字の個数をグラフにしたものである．横軸にセット名，縦軸に誤認識文字数を示した．この結果によると，全てのセットにおいて，CMF よりも提案手法である DS-CMF の誤認識文字数が減少していることが確認される．再細分類に CMF を用いた場合は誤字数が 7426 字だったのに対し，提案手法を用いた場合は 7113 字となり，

$$\frac{7426 - 7113}{7426} \times 100 = 4.21[\%]$$

の改善率が得られたことになる．このことから提案手法は CMF に比べて，再細分類に有効な識別方法であると言える．

最後に，提案手法により誤認識された文字の例を図 8 に示す．上段の (a)〜(c) ような文字は人間が見ても判別できないような文字であったため，誤認識されたと考えられる．しかし，下段の (d)〜(f) の文字のように人間が見て判断できるにもかかわらず，誤認識された文字もあり，今後も手法の改善が必要である．

6. おわりに

本稿では類似文字識別に有効である混合マハラノビス関数の補正項で用いる部分空間に，2 つの部分空間の差を表す差分部分空間を導入することで，類似クラスの差異を強調し，高精度な識別を行う手法を提案した．

提案手法では，類似クラスの差異が部分空間の差という多次元情報から構成されるため，従来の混合マハラノビス関数に比べて，より良くクラスの差異を捉えることが可能となる．

手書き文字データベース ETL9B を用いた実験を行った結果，認識率が従来の混合マハラノビス関数では 94.73% であったのに対し，提案手法では 94.96% であり，従来手法よりも高い認識精度が得られた．今後，より詳細な誤認識に関する検討を行い，手法の改善を行っていく予定である．

文　献

[1] 斉藤泰一，山田博三，山本和彦，“JIS 第 1 水準手書き漢字データベース ETL9B とその解析，” 信学論 (D), vol.J68-D, no.4, pp.757-764, Apl. 1985

[2] N. Kato, M. Suzuki, S. Omachi, H. Aso, and Y. Nemoto, “A handwritten character recognition system using directional element feature and asymmetric mahalanobis distance,” IEEE Trans. on Pattern Analysis & Machine Intelligence, vol.21, no.3, pp.258-262, March 1999.

[3] 澤和弘，若林哲史，鶴岡信治，木村文隆，三宅康二，“こう配特徴ベクトルと変動吸収共分散行列による手書き漢字認識の高精度化，” 信学論 (D-II), vol.J84-D-II, no.11, pp.2387-2397, Nov. 2001

[4] 鈴木雅人，大町真一郎，加藤寧，阿曽弘具，根元義章，“混合マハラノビス識別関数による高精度な類似文字識別手法，” 信学論 (D-II), vol.J80-D-II, no.10, pp.2752-2760, Oct. 1997

[5] 中嶋孝，若林哲史，木村文隆，三宅康二，“混合識別関数による類似文字認識の高精度化，” 信学論 (D-II), vol.J83-D-II, no.2, pp.623-633, Feb. 2000

[6] 鈴木雅人，加藤寧，根元義章，市村洋，“2 次混合マハラノビス関数を用いた類似文字識別手法，” 信学論 (D-II), vol.J84-D-II, no.4, pp.659-667, Apl. 2001

[7] 鈴木仁士，和泉勇治，加藤寧，根元義章，“非線形混合識別関数を用いた類似文字識別手法，” 信学論 (D-II), vol.J88-D-II, no.4, pp.704-715, Apl. 2005

[8] 加藤寧，安倍正人，根元義章，“改良型マハラノビス距離を用いた高精度な手書き文字認識，” 信学論 (D-II), vol.J79-D-II, no.1, pp.45-52, Jan. 1996

[9] 福井和広，山口修，鈴木薫，前田賢一，“制約相互部分空間法を用いた環境変動にロバストな顔画像認識 －照明変動の影響を抑える制約部分空間の学習－，” 信学論 (D-II), vol.J82-D-II, no.4, pp.613-620, Apl. 1999

[10] 福井和広，山口修，“一般化差分部分空間に基づく制約相互部分空間法，” 信学論 (D-II), vol.J87-D-II, no.8, pp.1622-1631, Aug. 2004

[11] 福井和広，山口修，“部分空間法の理論拡張と物体認識への応用，” 情処論, vol.46, no.sig15, pp.21-34, Oct. 2005

[12] 孫 寧，安倍正人，根元義章，“改良型方向線素特徴量および部分空間法を用いた高精度な手書き文字認識システム，” 信学論 (D-II), vol.J78-D-II, no.6, pp.922-930, Jun. 1995

多重直交相互部分空間法を用いた顔認識

河原　智一†　　西山　正志†　　山口　　修†

† (株) 東芝 研究開発センター　〒 212-8582 川崎市幸区小向東芝町 1
E-mail: †{tomokazu.kawahara,masashi.nishiyama,osamu1.yamaguchi}@toshiba.co.jp

あらまし　我々は制約相互部分空間法の特徴抽出を発展させた直交相互部分空間法を提案した．直交相互部分空間法では，識別に有効な特徴抽出を行うために直交化行列による変換を行い，参照部分空間同士の間の角度を広げることで識別性能を向上させている．高い識別性能を得るためには直交化行列の学習が重要であり，その生成法の確立が問題となる．一方，制約相互部分空間法における特徴抽出の学習では，アンサンブル学習に着目し，特徴抽出を複数回行い，各結果を統合することで認識性能を向上させる多重制約相互部分空間法を提案している．本稿では直交相互部分空間法に対して同様のアプローチを用いた多重直交相互部分空間法を提案する．顔画像実験により識別性能が向上することを確かめる．

キーワード　直交相互部分空間法，アンサンブル学習，顔認識

Face Recognition by Multiple Orthogonal Mutual Subspace Method

Tomokazu KAWAHARA†, Masashi NISHIYAMA†, and Osamu YAMAGUCHI†

† Corporate Research and Development Center TOSHIBA Corporation　1, Komukai Toshiba-cho, Saiwai-ku, Kawasaki-shi, 212-8582, Japan
E-mail: †{tomokazu.kawahara,masashi.nishiyama,osamu1.yamaguchi}@toshiba.co.jp

Abstract　We previously proposed the Orthogonal Mutual Subspace Method (OMSM). OMSM can extract effective features for identification by means of an orthogonalization matrix. To achieve high performance the matrix extraction method is critical. Additionally we previously proposed another method, the Multiple Constrained Mutual Subspace Method (MCMSM). MCMSM is a Constrained Mutual Subspace Method applied on the framework provided by ensemble learning. In this paper, we propose a new method based on these, the Multiple Orthogonal Mutual Subspace Method (MOMSM), for learning multiple orthogonalization matrices using ensemble learning. Through experimental results, we show the effectiveness of this method for face identification.

Key words　Orthogonal Mutual Subspace Method, Ensemble Learning, Face Recognition

1. はじめに

相互部分空間法 [3] に基づいたパターン認識法として，制約相互部分空間法 [6]〜[8] をはじめ，図 A·1 に示した手法が提案されてきた．最近では，制約相互部分空間法の特徴抽出を発展させた直交相互部分空間法 [13] を提案した．我々はこれらの手法を用いて，顔画像による個人認識 [14], [15] をターゲットとして，実環境で動作する個人認証システムを実現してきた [16], [17].

制約相互部分空間法および直交相互部分空間法では，カテゴリ間の関係を考慮し，部分空間同士の間の角度を広げる特徴抽出を行うことで，識別に有効なカテゴリ間の差異を強調し，認識性能を向上させている．この特徴抽出を制約相互部分空間法では制約部分空間への射影によって，直交相互部分空間法では直交化行列による変換によって実現している．制約部分空間および直交化行列の生成法については，高い識別性能を得るための学習方法の確立が重要である．

制約部分空間の学習については，アンサンブル学習 [18]〜[20] を取り入れ，学習によって複数の制約部分空間を生成し，各制約部分空間ごとに特徴抽出を行う多重制約相互部分空間法 [12] を提案した．アンサンブル学習を導入した方法は他にも [21], [22] などがあり，本稿ではこの考え方を直交化行列の学習に取り入れ，複数の直交化行列を作成し，各直交化行列ごとに特徴抽出を行う多重直交相互部分空間法 (Multiple Orthogonal Mutual Subspace Method: MOMSM) を提案する．多重直交相互部分空間法では，入力部分空間と参照部分空間を複数の直交化行列でそれぞれ変換し，各直交化行列で変換された入力部分空間と参照部分空間のなす角度を類似度として求める．得られた複数の類似度を結合することで最終的に類似度を決定する．複数の

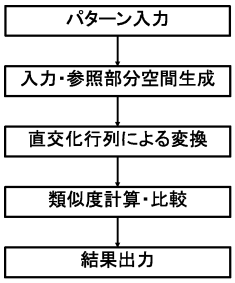


図 1　直交相互部分空間法の処理の流れ
Fig. 1　Flow chart of OMSM

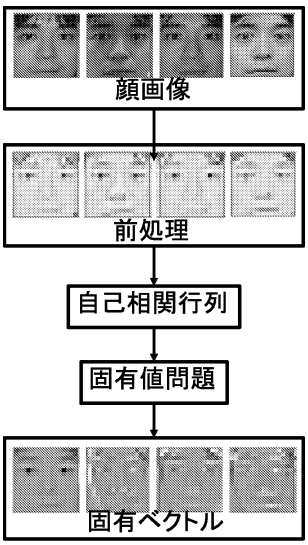


図 2　部分空間の生成の例 (顔パターンの場合)
Fig. 2　A Example of generation of a subspace from face patterns

直交化行列の生成する方法は多重制約相互部分空間法を踏襲し，Bagging [19] に代表される並列的な学習の考え方を利用する．まず，ランダムにいくつかのカテゴリをサンプリングを行い，そのカテゴリに属する学習パターンを用いて直交化行列を生成し，この操作を必要な数だけ繰り返すことで複数の直交化行列を生成する．

以下，2. 章で従来法である直交相互部分空間法について説明し，3. 章で提案手法である多重直交相互部分空間法について述べる．提案手法の有効性を確認するため，4. 章で識別実験を行う．その結果を踏まえ，5. 章で考察を行う．

2.　直交相互部分空間法

直交相互部分空間法は複数の入力・参照パターンを用いて識別を行う手法である．簡単な処理の流れを図 1 に示す．認識対象の複数の入力パターンの自己相関行列 [23] に対して固有値問題を解き，その固有ベクトルを基底とすることで，入力部分空間を生成する (図 2)．識別の際に参照する参照パターンについても，同様の方法で参照部分空間を生成する．次に，入力・参照部分空間に対して直交化行列による変換を施した後，それらの部分空間同士の間の角度から類似度を算出する．

直交化行列は，複数ある部分空間に対して部分空間同士の間の角度を広げる線形変換を表す行列である．特に，全部分空間の基底の集合が 1 次独立なら，直交化行列による変換で各部分空間同士は直交する [13]．一般には，カテゴリ数と学習用部分空間の次元の積が空間の次元以下なら，学習用部分空間の基底の集合は 1 次独立になる．

2.1　直交化行列の生成

まず，直交化行列生成の際に使用する学習用部分空間について述べる．同一カテゴリ内のパターン変動を学習するため，各カテゴリごとに変動を含んだパターンを複数用意し，先と同様の方法で部分空間を生成する (図 2)．それぞれのカテゴリについて生成した部分空間を学習用部分空間と呼ぶ．学習用部分空間の固有値が大きい基底ベクトルは，各カテゴリの主要なパターン変動を表していると考えられる．直交化行列生成の際に使用する学習用部分空間は，認識対象のカテゴリのパターンから生成するのが理想であるが，認識対象とは異なるカテゴリの学習用部分空間から生成することも可能である．

次に，カテゴリ数を L，各カテゴリの学習用部分空間の次元を N_B，学習用部分空間へ射影する射影行列を $\mathbf{P}_j (j = 1 \ldots L)$ とする．ただし，カテゴリ毎の学習用部分空間の正規直交基底を $\psi_{jk} (k = 1 \ldots N_B)$ とすると，射影行列 $\mathbf{P}_j$ は次のようになる．

$$\mathbf{P}_j = \sum_{k=1}^{N_B} \psi_{jk} \psi_{jk}^T \tag{1}$$

その L 個の射影行列の平均を $\mathbf{P}$ (式 (2)) とすると，$\mathbf{P}$ は L 個の学習用部分空間の分布を表現する行列となる．

$$\mathbf{P} = \frac{1}{L}(\mathbf{P}_1 + \mathbf{P}_2 + \ldots + \mathbf{P}_L) \tag{2}$$

$$\mathbf{P}v = \lambda v \tag{3}$$

具体的には，式 (4) のように，$\mathbf{P}$ の固有値 λ は，$\mathbf{P}$ の固有ベクトルでノルムを 1 に正規化した v をすべての学習用部分空間へ射影し，その射影長の 2 乗の平均と一致する．

$$\lambda = \frac{1}{L} \sum_{i=1}^{L} |\mathbf{P}_i v|^2 \tag{4}$$

ただし，$|\cdot|$ はベクトルのノルムとする．式 (4) から，L 個の学習用部分空間の分布は，固有値が大きい固有ベクトルの方向ほど密集している．そのため，図 3 のように，行列 $\mathbf{P}$ の固有値をすべて 1 にする白色化変換によって学習部分空間の間の角度が広がると考えられる．

直交化行列 $\mathbf{O}$ は $\mathbf{P}$ の固有値をすべて 1 にする白色化変換を表す行列として，式 (5) で与える．

$$\mathbf{O} = \mathbf{\Lambda}^{-1/2} \mathbf{B}^t \tag{5}$$

ただし，$\mathbf{\Lambda}^{-1/2}$ は $\mathbf{P}$ の固有値の平方根の逆数を並べた対角行列，$\mathbf{B}$ は $\mathbf{P}$ の固有ベクトルを並べた行列，右上の t はその転置である．式 (6) により，$\mathbf{O}$ による変換で，固有値がすべて 1 になることが分かる．ただし，$\mathbf{I}$ は単位行列とする．

$$\mathbf{OPO}^t = \mathbf{\Lambda}^{-1/2} \mathbf{B}^t \mathbf{B} \mathbf{\Lambda} \mathbf{B}^t \mathbf{B} \mathbf{\Lambda}^{-1/2} = \mathbf{I} \tag{6}$$

なお，N_B は実験的に定める．

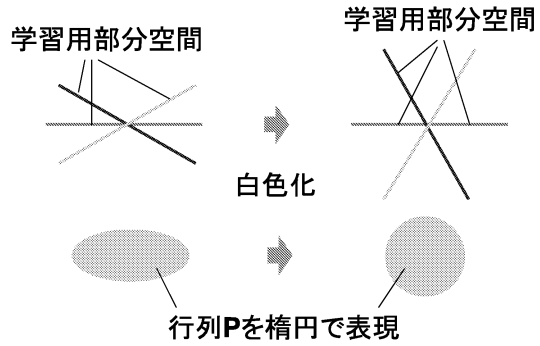


図 3　白色化変換により部分空間同士の角度を広げる
Fig.3　Whitening makes angles between subspaces larger

2.2　直交化行列による特徴抽出

複数のパターンから生成した入力・参照部分空間 (図 2) に対して，部分空間を張る N 本の基底を直交化行列 $\mathbf{O}$ で変換し，変換後の N 本のベクトルに Gram-Schmidt の直交化を施す．Gram-Schmidt 直交化された N 本のベクトルをそれぞれ直交化行列で変換した入力・参照部分空間の基底とする．

2.3　部分空間同士の類似度計算

直交化行列による変換後の入力部分空間と参照部分空間の間の角度から類似度を計算し，類似度が最大の参照部分空間に対応するカテゴリを入力パターンの属するカテゴリと判定する．また，類似度がしきい値を下まわる場合は判定せず棄却する．類似度の計算は次の通りである．

直交化行列による変換後の入力部分空間を P，参照部分空間を Q とする．P と Q の類似度 s は，相互部分空間法 [3] により得られる正準角と呼ばれる部分空間同士の間の角度 θ から式 (7) で決定される．

$$s = \cos^2\theta \tag{7}$$

部分空間同士が完全に一致していれば $\theta = 0$ である．$\cos^2\theta$ は，以下の行列 $\mathbf{X}$ の最大固有値となる．

$$\mathbf{X} = (x_{mn}) \quad (m, n = 1 \ldots N) \tag{8}$$

$$x_{mn} = \sum_{l=1}^{N} (\psi_m, \phi_l)(\phi_l, \psi_n) \tag{9}$$

ここで，ψ_m, ϕ_l は部分空間 P，Q の m,l 番目の基底ベクトル，(ψ_m, ϕ_l) は ψ_m と ϕ_l の内積，N は部分空間の基底ベクトルの数を表す．

3.　多重直交相互部分空間法

本稿で提案する多重直交相互部分空間法は，Bagging [19] の考え方を直交相互部分空間法に導入した手法である．Bagging は複数の識別器を用意し，各識別器から得られた結果を統合することで認識を行う．それぞれの識別器はすべての学習パターンからランダムにサンプリングしたパターンを用いて生成する．学習パターンの選択にランダム性があるため，生成された識別器は異なるものになる．本手法では複数の直交化行列を用意し，各直交化行列ごとに直交相互部分空間法を行った結果を統合することで識別を行う．簡単な処理の流れを図 4 に示す．

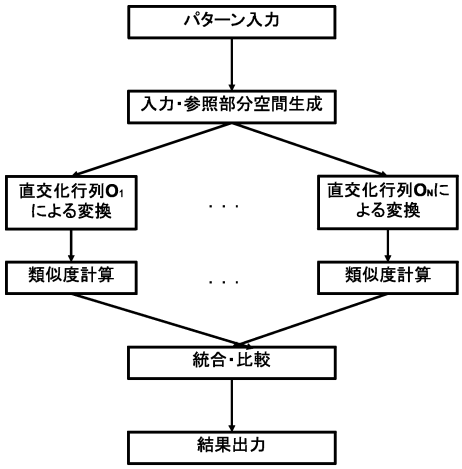


図 4　多重直交相互部分空間法の処理の流れ
Fig.4　Flow chart of MOMSM

3.1　複数の直交化行列の生成

複数の直交化行列を作成するため，ここでは L 個の学習用部分空間からランダムに L' 個をサンプリングし，直交化行列を生成する．サンプリング後の直交化行列の作成法は 2.1 と同じである．M 個の直交化行列 O_i $(i = 1, \ldots M)$) を生成する手続きを以下に示す．

（1）　L 個の学習用部分空間から重複しないようランダムに L' 個の学習用部分空間を選択する．

（2）　L' 個の学習用部分空間の基底ベクトルを用いて 2.1 の手続きで 1 個の直交化行列を生成する．

（3）　M 個の直交化行列が生成されるまで (1) に戻る．

なお，L' は実験的に定める．

3.2　複数の直交化行列による特徴抽出

入力部分空間 V に対して，各直交化行列 O_i $(i = 1, \ldots M)$ ごとに 2.2 の処理を行うことで，入力部分空間 V_i $(i = 1, \ldots M)$ を得る．参照部分空間 W に対しても同様の処理を行うことで，参照部分空間 W_i $(i = 1, \ldots M)$ を得る．

3.3　複数の類似度計算

直交化行列 O_i による変換で得られた入力部分空間 V_i と参照部分空間 W_i の類似度 s_i を 2.3 の方法で計算する．

3.4　複数類似度の統合

各直交化行列に特徴抽出で得られた類似度を統合し，多重直交相互部分空間法の類似度 s を以下の式で求める．ここでは平均値を用いる．

$$s = \frac{1}{M} \sum_{i=1}^{M} s_i \tag{10}$$

4.　認 識 実 験

提案手法と従来法の認識性能を比較するため，動画データによる 2 種類の認識実験を行った．実験 1 では認識対象の様々な変動パターンから学習した特徴抽出を用いて実験した．実験 2 は，実用に近い形での認識性能を比較するために認識対象とは異なるカテゴリの様々な変動パターンを用いて特徴抽出を行った．

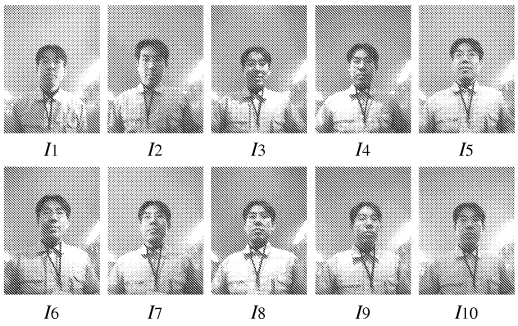


図 5　各照明条件における撮影画像の例
Fig. 5　Examples of captured image in each lighting condition.

4.1　実験の仕様

認識実験に使用したデータは独自に収集した動画像である．50 人の人物について，異なる 10 種類の照明条件下 (以下，$I_1, \ldots I_{10}$ と呼ぶ) で撮影した．図 5 に I_1 から I_{10} の照明条件で取得された画像を示す．各画像から瞳と鼻孔の位置を基準として顔領域のパターンを抽出し，パターンをラスタースキャンによりベクトルに変換する．前処理を行い，ベクトル長の正規化された 210 次元に変換した．撮影条件や前処理等の詳細は [12] を参照していただきたい．

認識実験は，50 人を 25 人ずつの 2 グループ (A,B) に分け，グループ A で認識実験を行い，グループ B は特徴抽出の学習のみに使用した．各グループについて，各照明環境下で取得された画像列を前半と後半に分け，前半を参照用，後半を入力用とし，実験の試行回数を増やすため，入力用の画像列をさらに分割し，1 人の人物に対して試行を複数回の入力とした．認識実験に使用しないグループ B は参照部分のみを使用した．以下，照明環境 I_i で撮影されたグループ A，B の画像列の参照用をそれぞれ A_{I_i}，B_{I_i}，入力用を A'_{I_i}，B'_{I_i} と呼ぶ．

認識実験は全部で 2 種類行った．特徴抽出の学習を除き，2 種類とも A_{I_i} を参照し，$A'_{I_1}, \ldots, A'_{I_{10}}$ の入力に対して人物の識別を行った．結果は，$A_{I_1}, \ldots, A_{I_{10}}$ それぞれを参照とした実験結果の平均とした．

特徴抽出の学習に使用した学習用部分空間は，実験 1 では $A_{I_1}, \ldots, A_{I_{10}}$ から，実験 2 では $B_{I_1}, \ldots, B_{I_{10}}$ から生成した．つまり，学習用部分空間はそれぞれ，実験 1 では参照する人物の様々な変動を含む理想的なパターン，実験 2 では参照する人物とは異なる人物の様々な変動を含む実用的なパターンから生成した．これらの学習用部分空間から直交化行列および制約部分空間を生成した．以上をまとめたのが表 1 である．

表 1　参照，入力および特徴抽出の学習に用いたパターン
Table 1　Reference, input and training patterns

	参照	入力	学習
実験 1	A_{I_i}	$A'_{I_1}, \ldots, A'_{I_{10}}$	$A_{I_1}, \ldots, A_{I_{10}}$
実験 2	A_{I_i}	$A'_{I_1}, \ldots, A'_{I_{10}}$	$B_{I_1}, \ldots, B_{I_{10}}$

4.2　識別精度の評価基準

識別精度の評価は次の 2 つの基準を用いた．

1 位正解率 (CMR: Correct Matching Rate):
本人類似度が他人類似度より高くなる割合．本人類似度は入力部分空間と参照部分空間に対応する人物が同じ場合に算出された類似度，他人類似度は異なる場合に算出された類似度である．

等価エラー率 (EER:Equal Error Rate):
FAR(他人受理誤り率) と FRR(本人排除誤り率) が等しい時の割合．FAR は以下の式で求まる．

$$FAR = A/B$$

ただし，A =(他人類似度がしきい値以上の試行数)，B =(全試行数 − 本人の試行数) である．登録数は参照するカテゴリ数を表す．一方，FRR は以下の式で求まる．

$$FRR = C/D$$

ただし，C =(本人類似度がしきい値以下の試行数)，D =(本人の試行数) である．顔認識の場合，登録されていない未知の人物に対応する必要があり，EER が低いとその手法の認識性能は高い．

4.3　実験に用いた手法

提案手法の有効性を確認するために，識別性能について従来手法との比較実験を行った．各手法のパラメータ等は以下のように設定した．

相互部分空間法 (MSM)
入力部分空間と参照部分空間の間の角度 θ の $\cos^2\theta$ を類似度とした．入力および参照部分空間の次元は 7 次元とした．

制約相互部分空間法 (CMSM)
制約部分空間への射影により特徴抽出を行い，相互部分空間法により類似度を計算した．入力および参照部分空間の次元は 7 とした．学習用部分空間の次元 $N_B = 35$ 次元とし，制約部分空間の次元 $N_C = 188$ 次元とした．

多重制約相互部分空間法 (MCMSM)
制約部分空間への射影により特徴抽出を行い，相互部分空間法により類似度を計算した．入力および参照部分空間の次元は 7 とした．学習用部分空間の次元 $N_B = 35$ 次元とし，制約部分空間の次元 $N_C = 190$ 次元とした．また制約部分空間の数 M を 5 とし，各制約部分空間は 25 個の学習用部分空間から 15 個をランダムにサンプリングして生成した．この学習方法は M 個の選択にランダム性があるため，CMR と EER は一連の手続きを 10 回行い，平均値とした．

直交相互部分空間法 (OMSM)
直交化行列による線形変換により特徴抽出を行い，相互部分空間法により類似度を計算した．入力および参照部分空間の次元は 7 とし，学習用部分空間の次元 $N_B = 35$ 次元とした．

多重直交相互部分空間法 (OMSM)
直交化行列による線形変換により特徴抽出を行い，相互部分空間法により類似度を計算した．入力および参照部分空間の次元は 7 とした．学習用部分空間の次元 $N_B = 35$ 次元とした．また直交化行列の数 M を 5 とし，各直交化行列は 25 個の学習用部分空間から 15 個をランダムにサンプリングして生成した．

この学習方法は M 個の選択にランダム性があるため，CMR と EER は一連の手続きを 10 回行い，平均値とした．

4.4 実験 1(参照する人物の様々な変動を学習)

認識対象の様々な変動パターンから学習用部分空間を生成し，特徴抽出を行った実験の結果を表 2 に示す．

表 2　実験 1 の結果
Table 2　Results of experiment 1

	平均 CMR(%)	平均 EER(%)
MSM	95.51	5.48
CMSM	96.48	2.99
MCMSM	97.30	2.63
OMSM	99.17	1.69
MOMSM	99.08	1.46

CMR では OMSM，EER では MOMSM が最も良く，OMSM と MOMSM の結果が近いものとなった．今回の実験では，学習パターンが参照する人物自身の様々な変動を含んでいるため，OMSM の直交化行列の学習が十分に行われていると言える．一方，MOMSM の個々の直交化行列は 25 人中 15 人のデータのみで学習しているため，人数の観点からは不利な状況であるが，それぞれの結果を統合することで OMSM と同等の認識性能が得られることがわかる．これから，制約部分空間だけでなく直交化行列についても，十分に学習が行えない場合に直交化行列を複数生成し，それぞれの結果を統合することで，十分に学習が行えた場合と同等の認識性能が得られる可能性があることが分かった．

4.5 実験 2(参照する人物とは異なる人物の様々な変動を学習)

次に、学習パターンの違いによる影響をみるために，参照する人物を全く含まない，異なる人物グループ B で特徴抽出の学習を行った，認識実験の結果を表 3 に示す．パラメータについては実験 1 と同様である．なお，特徴抽出のための射影や線形変換を行わないため MSM の結果は実験 1 と同じである．

表 3　実験 2 の結果
Table 3　Results of experiment 2

	平均 CMR(%)	平均 EER(%)
MSM	95.51	5.48
CMSM	95.97	3.47
MCMSM	96.87	3.01
OMSM	98.35	2.15
MOMSM	98.55	1.84

ここでは，CMR，EER 共に MOMSM が最も良い結果を得られた．今回の実験では，学習パターンに含まれる人物の変動が参照する人物のものとは異なるため，あらかじめ参照人物が特定できないような状況における認識性能を調べることが目的となる．このように学習が不十分な場合にも，直交化行列を複数にすることで認識性能の向上する可能性があることが分かった．さらに，OMSM との結果を比較すると，学習用に多くの人物のパターンを集められない場合に全人物から 1 つの直交化行列を作るより，少ない人数から複数の直交化行列を作る方が有効であると考えられる．

5. 特徴抽出の数と認識性能

特徴抽出の数と認識性能の関係を調べるため，直交化行列の数および制約部分空間の数を 1 から 10 まで変化させ 4. 章の実験 1，2 を行った．図 6 は実験 1，図 7 は実験 2 の結果で，横軸は特徴抽出の数，縦軸は EER である．各図内の横線は OMSM および CMSM の結果である．

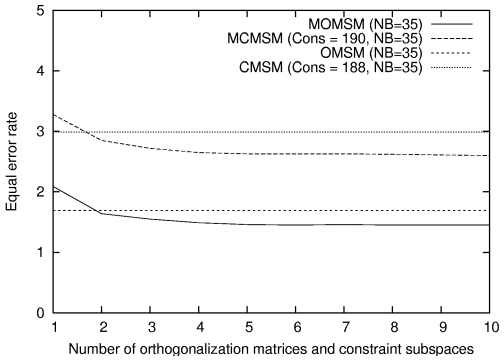


図 6　特徴抽出の数と認識性能 (EER) の関係 (実験 1)
Fig. 6　Relation between number of feature extraction and equal error rate in experiment 1.

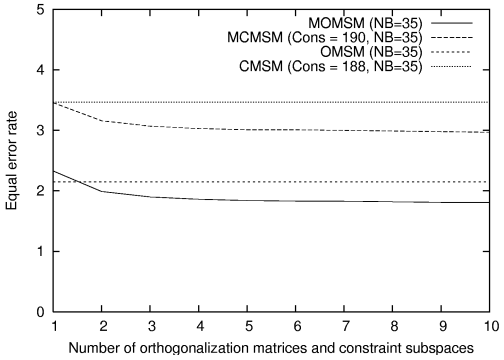


図 7　特徴抽出の数と認識性能 (EER) の関係 (実験 2)
Fig. 7　Relation between number of feature extraction and equal error rate in experiment 2.

まず，両図から MOMSM および MCMSM のグラフの傾向が似ていることが分かる．OMSM，CMSM 共に参照部分空間の間の角度を広げる特徴抽出を行うが，その効果が似ているため，複数回行った場合の挙動も似ていると考えられる．

次に，特徴抽出の数が 1 の時，つまり 15 人で特徴抽出を学習した場合，25 人で学習した場合に比べ，OMSM および CMSM 共に認識性能が落ちていることが分かる．これらの傾向が実験 1，2 両方に見られるとことから，参照する人物の変動を学習できる場合だけでなく，参照する人物とは異なる人物の変動を学習する場合にも学習する人数が両手法で認識性能に重要であると考えられる．学習する人数と認識性能の関係の考察については今後の課題とする．

また，特徴抽出数が 1 と 2 の場合を比べると，図 6，7 共に 2 の場合に大きく認識性能が大きく向上している．一方，特徴抽出数が 2 と 3 以上を比較した場合，特徴抽出数が増えるほど認識性能が向上するものの，上昇分は微小であることが分かる．今回は 25 個の学習用部分空間から 15 個をランダムに選んで特徴抽出を学習していたため，特徴抽出の数が増えてもその変化が少ないという原因が考えられるが，これらの検証も含め，特徴抽出の数と認識性能の関係の考察についても今後の課題とする．

本稿では学習に用いる人物のバリエーションを変化させることで，直交化空間を生成したが，小規模な実験であったため直交化空間のバリエーションも少ない．認識対象とする人物を増やした実験を今後行うとともに，学習に用いる人数が多い場合には，AdaBoost による学習アルゴリズムの効果が期待できると考えている．

6. おわりに

本稿では，複数の直交化行列の生成に対してアンサンブル学習を適用し，複数の直交化行列を用いて特徴抽出を多重化した多重直交相互部分空間法について提案した．提案手法では，入力部分空間と参照部分空間を複数の直交化行列による変換を施すことで特徴抽出を行い，直交化行列によって変換された入力部分空間と参照部分空間の類似度を算出し，直交化行列の数だけ求まった類似度を結合することで最終的に類似度を決定する．複数の直交化行列を生成するために，アンサンブル学習の代表的な手法である Bagging の考え方を導入した．提案手法により従来の認識手法に比べて識別精度が向上することを，照明変動が生じる環境で撮影されたデータベースを用いた識別実験で確認した．

文　献

[1] S. Watanabe, and N. Pakvasa, “Subspace method of pattern recognition,” Proc. 1st Int. J. Conf. on Pattern Recognition (1973).

[2] E. Oja, Subspace Methods of Pattern Recognition, Research Studies Press, (1983).

[3] 前田賢一，渡辺貞一，“局所的構造を導入したパターンマッチング法，” 信学論，D Vol. J68-D，No.3，pp.345-352 (1985).

[4] 津田宏治，“ヒルベルト空間における部分空間法，” 信学論 D-II Vol. J82-D-II，No.4，pp.592-599，(1999)

[5] 前田英作，村瀬洋，“カーネル非線形部分空間法によるパターン認識，” 信学論 D-II Vol. J82-D-II，No.4，pp.600-612，(1999)

[6] 福井和広，山口修，鈴木薫，前田賢一，“制約相互部分空間法を用いた環境変動にロバストな顔画像認識 –照明変動の影響を抑える制約相互部分空間の学習–，” 信学論，D-II Vol. J82-D-II, No. 4, pp. 613-620 (1999).

[7] K. Fukui, and O. Yamaguchi, “Face Recognition Using Multi-viewpoint Patterns for Robot Vision,” 11th International Symposium of Robotics Research, pp. 192-201 (2003).

[8] 福井和広，山口修，“一般化差分部分空間に基づく制約相互部分空間法，“ 信学論，D-II Vol. J87-D-II, No. 8, pp. 1622-1631 (2004).

[9] 坂野鋭，武川直樹，中村太一，“核非線形相互部分空間法による物体認識，” 信学論 D-II Vol. J84-D-II，No.8，pp.1549-1556 (2001)

[10] 福井和広，“カーネル差分部分空間に基づく非線形特徴抽出，” 第 6 回情報論的学習理論ワークショップ (IBIS2003)，pp.265-270 (2003)

[11] 福井和広，山口修，“カーネル非線形制約相互部分空間法による物体認識，” 信学論 D-II Vol. J88-D-II，No.8，pp.1349-1356 (2005)

[12] 西山正志，山口修，福井和広，“多重制約相互部分空間法を用いた顔画像認識，“ 信学論，D-II Vol. J88-D-II, No. 8, pp. 1339-1348 (2005).

[13] 河原 智一，西山正志，山口修，“直交相互部分空間法を用いた顔認識，“ 情処研報，CVIM-151，pp17-24(2005).

[14] 赤松茂，“コンピュータによる顔の認識 –サーベイ–，“ 信学論，D-II Vol. J80-D-II，No. 8，pp. 2031-2046 (1997).

[15] 岩井儀雄，労世竑，山口修，平山高嗣，“画像処理による顔検出と顔認識，“ 情処研報，CVIM-149, pp. 343-368 (2005).

[16] 佐藤俊雄，助川寛，横井謙太朗，土橋浩慶，緒方淳，岡崎彰夫，“立ち位置変動を考慮した顔照合セキュリティシステム「FacePass」の開発，“ 映像情報メディア学会誌，Vol. 56，No.7，pp.1111-1117 (2002).

[17] 小坂谷達夫，山口修，福井和広，“制約相互部分空間法を用いた顔認識システムの開発と評価，“ 情報処理学会論文誌，Vol, 45. No.3, pp.951-959 (2004).

[18] 麻生英樹，津田宏治，村田昇，“パターン認識と学習の統計学 新しい概念と手法，“ 岩波書店，(2003).

[19] L. Breiman, “Bagging Predictors,” Machine Learning, Vol.24, No.2, pp.123-140, (1996).

[20] Y. Freund and R.E. Schapire, “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting,” J. Comput. Syst. Sci., Vol.55, No.1, pp.119-139 (1997).

[21] G.-D. Guo and H.-J. Zhang, “Boosting for Fast Face Recognition, Second International Workshop on Recognition,” Analysis and Tracking of Faces and Gestures in Real-time Systems(2001).

[22] X. Wang and X. Tang, “Random Sampling LDA for Face Recognition,” IEEE Proc. Conf. on Computer Vision and Pattern Recognition, vol.2, pp.259-265 (2004).

[23] 石井健一郎，上田修功，前田英作，村瀬洋，“わかりやすいパターン認識，” オーム社 (1998).

[24] 福井和広，山口修，“部分空間法の理論拡張と物体認識への応用，“ 情報処理学会論文誌 Vol, 46. No. SIG 15(CVIM 12), pp.21-34 Oct. (2005).

付　録

部分空間法の拡張の流れの中における，直交相互部分空間法および多重直交相互部分空間法の位置関係を整理するため，部分空間法の機能拡張の系譜 [24] に，これらの手法を入れたものが図 A·1 である．

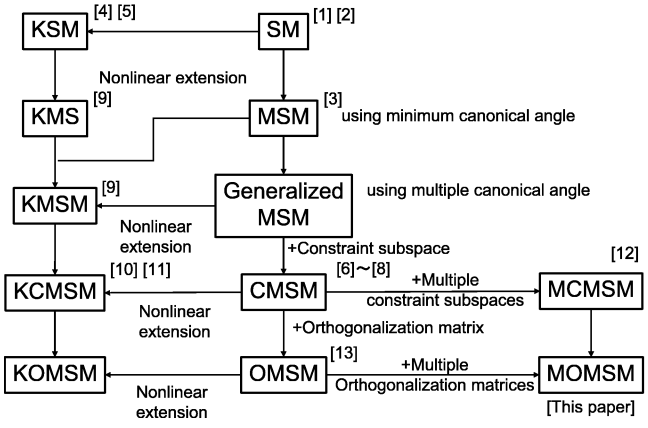


図 A·1　部分空間法に対する機能拡張の系譜

Fig. A·1　Genealogy of function enhancement of subspace method

制約相互部分空間法と直交相互部分空間法の比較

– クラス部分空間の直交化の観点から –

福井　和広†　　山口　修††

† 筑波大学大学院システム情報工学研究科　〒 305–0031 つくば市天王台 1–1–1
† 株式会社東芝研究開発センター　〒 212–8582 川崎市幸区東芝町 1
E-mail: †kfukui@cs.tsukuba.ac.jp, ††osamu1.yamaguchi@toshiba.co.jp

あらまし　本稿では，相互部分空間法の拡張である制約相互部分空間法と直交相互部分空間法の識別性能を比較する．これらの方法は，入力部分空間と各クラス部分空間の成す角度を測る前に，クラス部分空間同士をできるだけ直交化しておくことで相互部分空間法の識別性能を高めている．制約相互部分空間法では，各クラス部分空間を制約部分空間へ射影し，共通部分空間を取り除くことで，結果的に直交状態に近づける．一方，直交相互部空間法では，各クラス部分空間の基底ベクトルを直交化変換することで明示的に直交化を実現する．このように両者の直交化の原理は異なるので，両者の優劣は対象パターン分布や学習データの質などに依存して変化する．総じて，線形識別では直交相互部分空間法が優位であり，カーネル関数を用いた非線形識別では同程度の識別性能を実現する．このため，非線形識別のクラス数が多い問題においては，計算量の少ない制約相互部分空間法がやや有利となると考えられる．
キーワード　制約，直交相互部分空間法，カーネル非線形制約相互部分空間法，カーネル非線形直交相互部分空間法

Comparison between constrained mutual subspace method and orthogonal mutual subspace method

Kazuhiro FUKUI† and Osamu YAMAGUCHI††

† System Information Engineering, Tsukuba University　Tennoudai 1–1–1, Tsukuba,Japan
† Toshiba Corporate R&D Center　Komukai-Toshiba-chou 1, Saiwai-ku, Kawasaki, Japan
E-mail: †kfukui@cs.tsukuba.ac.jp, ††osamu1.yamaguchi@toshiba.co.jp

Abstract　In this paper, we compare the performances of the constrained mutual subspace method (CMSM) and the orthogonal mutual subspace method (OMSM), which were proposed to improve the performance of the mutual subspace method (MSM). Since the recognition performances of them vary depending on a shape of pattern distribution and a quality of learning patterns, they should be used properly according to a problem, considering the computing times.
Key words　Constrained mutual subspace method, orthogonal mutual subspace method

1. はじめに

本稿では，制約相互部分空間法 [1]〜[3] と直交相互部分空間法 [4] の特性を線形識別とカーネル関数を用いた非線形識別の両方の枠組みにおいて比較する．

これらの方法は相互部分空間法 [5] の識別能力を向上させるために，相互部分空間法にそれぞれ異なる特徴抽出を前処理として加えた方法である．相互部分空間法 (MSM) は部分空間法 [6]〜[8] の自然な拡張として提案された．部分空間法では入力ベクトルと辞書部分空間の成す角度，あるいは射影長を基に識別を行うのに対して，相互部分空間法では入力側も部分空間として 2 つの部分空間の成す角度を類似度とする．動画像系列や複数視点画像セットなど，そのパターン分布を線形部分空間で表せる様々な問題に適用可能である．

相互部分空間法は部分空間法に比べてパターン変形に対する吸収能力が大きく向上するが，識別性能については十分とは言えない．これは KL 展開（主成分分析）を適用して得られるクラス部分空間は，各クラスのパターン分布を近似するという点では最適な部分空間になっているが，識別の観点からは必ずしも最適であるとは限らないからである．

そこで相互部分空間法の識別能力を改善する目的で，制約相互部分空間法と直交相互部分空間法が提案された．両者は異なるアルゴリズムで特徴抽出を行うが，それらの有効性は各クラス部分空間を事前にできるだけ直交化するという観点から共通的に説明できる．制約相互部分空間法 (CMSM) では，制約部分空間と名付けられた特徴空間へ各クラス部分空間を射影する．これにより各クラス部分空間から共通的な部分空間を取り取り除き，結果的に各クラス部分空間同士の角度を拡げ，各クラス部分空間の関係を直交関係に近づける．一方，直交相互部分空間法 (OMSM) では，全クラス部分空間を張る基底ベクトルに直交化変換を施し，基底ベクトルの分布を白色化することで，各クラス部分空間を直交化する．

両者の直交化の原理は異なるので，対象パターン分布や学習パターンの再現性などに依存して識別性能の優劣が変わる．正面顔を用いた評価実験では，OMSM の方が高い識別性能を達成できるという報告がなされている．総じて，直交化行列を用いて明示的に直交化を実現する OMSM の方が高い識別性能を達成できる場合が多い．しかしながら，汎化能力を考えた場合，学習パターンに対して最大限の直交化を実現する OMSM が必ずしも最適とは言えないかも知れない．また各クラス部分空間に共通部分空間が存在する場合，OMSM では本来識別に寄与しない共通部分空間も平等に直交化されるために，識別性能が大きく低下する可能性がある．これに対して，CMSM では共通部分空間は削除されるので，その影響は小さいと考えられる．

パターン分布の非線形性が強く，各クラスのパターン分布を重なりの無い線形部分空間で表せない場合には，CMSM, OMSM ともに性能低下する．これを解決するためには，カーネル関数を用いた非線形化が有効である．部分空間法，相互部分空間法のそれぞれの非線形拡張であるカーネル非線形部分空間法 [9], [10]，非線形核相互部分空間法 [14] は，線形識別に比べて識別性能が向上することが確認されている．同様に CMSM の非線形拡張であるカーネル非線形制約相互部分空間法 (KCMSM) [11]〜[13] も CMSM に比べて高い識別性能を実現する．

直交相互部分空間法の非線形化も同様に可能である．CMSM と OMSM の計算過程は固有値を考慮するか否かの違いのみで基本的には同じなので，KCMSM の計算過程において，固有値を考慮することで OMSM を非線形化できる．本稿ではその非線形化された直交相互部分空間法をカーネル非線形直交相互部分空間法 (Kernel nonlinear orthogonal mutual subspace method (KOMSM)) と名付ける．線形識別の OMSM では，クラス数と各クラスの次元数の積が，特徴空間の次元数より大きい場合には，完全な直交化はできない．これに対して，KOMSM では特徴空間の次元に上限がないために，ほぼ完全な直交度を達成できるという特性を持つ．興味深いことに，線形識別の枠組みで観察された CMSM と OMSM の識別性能の優劣は，非線形識別においてはほぼ見られなくなり，同程度の高い識別性能を示す．実際に，各クラス部分空間の平均的な直交度を計算すると，KOMSM のほぼ完全な直交化に対して，KCMSM も極めて高い直交化を実現し，両者の差は小さい．

カーネル関数を用いた非線形識別は高い識別性能を実現できる反面，学習データ数，クラス数に依存した計算量が大きな問題になる．計算量の観点から KCMSM と KOMSM を比較すると，KCMSM の方が有利と思われる．KOMSM では識別すべきクラス数に比例したサイズの行列の固有値と固有ベクトルを全て求める必要とがあるが，KCMSM は固有値の大きい側の一部の固有ベクトルを求めるだけで済む．

以下，第 2 章では先ずベースとなる相互部分空間法を説明し，その拡張として制約相互部分空間法と直交相互部分空間法のアルゴリズムを説明する．第 3 章ではカーネル非線形制約相互部分空間法について説明し，これに基づいて直交相互部分空間法をカーネ非線形直交相互部分空間法へと非線形拡張する．第 4 章では比較実験の結果に基づいて両者の特性を比較する．第 5 章はまとめである．

2. 線形識別アルゴリズム

本章では先ず部分空間の成す正準角の概念を導入して，これに基づいて各方法のアルゴリズムについて説明する．

2.1 相互部分空間法

相互部分空間法 (Mutual Subspace Method(MSM)) [5] は部分空間法 (Subspace Method) [6] の拡張であり, 図 1 に示すように，入力と辞書の 2 つの部分空間のなす最小角度 θ_1 に基づいて識別を行う．この最小角度を基準とした識別は正準角の概念 [15] を用いて一般化できる．m 次元部分空間 $\mathcal{P}$ と n 次元部分空間 $\mathcal{Q}$(便宜上, m≧n と仮定) の間には n 個の正準角が定義でき, 第 1 正準角 θ_1 は 2 つの部分空間のなす最小角となる．第 2 正準角 θ_2 は最小正準角 θ_1 に直交する方向において計った最小角, 同様に第 3 正準角 θ_3 は第 2 正準角 θ_2 に直交する方向で計った最小角である．以下同様に n 個の正準角 θ_i $(i = 1, \ldots, n)$ が順次求まる．これを式で書くと次のようになる．

$$cos^2\theta_i = \max_{\substack{\mathbf{u}_i \perp \mathbf{u}_{j(=1,\ldots,i-1)} \\ \mathbf{v}_i \perp \mathbf{v}_{j(=1,\ldots,i-1)}}} \frac{|(\mathbf{u}_i \cdot \mathbf{v}_i)|^2}{||\mathbf{u}_i||^2 ||\mathbf{v}_i||^2} \tag{1}$$

ここで，$\mathbf{u}_i \in P, \mathbf{v}_i \in Q, ||\mathbf{u}_i|| \neq 0, ||\mathbf{v}_i|| \neq 0$，また $(\cdot)$ と $||\cdot||$ はそれぞれ内積とノルムを表す．

本論文では，相互部分空間法とはこれら複数の正準角により一般化された方法を指すものとして議論を進める．

2.2 正準角の計算

正準角は以下のように計算できる．部分空間 $\mathcal{P}$ と $\mathcal{Q}$ の各正規直交基底ベクトル $\mathbf{\Phi}, \mathbf{\Psi}$ から求まる射影行列を，$\mathbf{P} = \sum_{i=1}^{m} \mathbf{\Phi}_i \mathbf{\Phi}_i^T$ と $\mathbf{Q} = \sum_{i=1}^{n} \mathbf{\Psi}_i \mathbf{\Psi}_i^T$ とすると，$\mathbf{PQP}$(あるいは $\mathbf{QPQ}$) の第 i 番目に大きい固有値 λ_i が第 i 正準角 θ_i に対する $cos^2\theta_i$ となる．更に，この行列の固有値問題は 以下の小さい行列 $\mathbf{S}$ の固有値問題(注1)に変換され，最終的に $cos^2\theta_i$ はこの行列の第 i 固有値 λ_i として求まる [5].

$$\mathbf{S} = (s_{ij}), \ (i, j = 1 \sim n) \tag{2}$$

(注1)：統計学では，この行列 $\mathbf{S}$ の特異値が正準角の余弦となることが古くから知られていた [15].

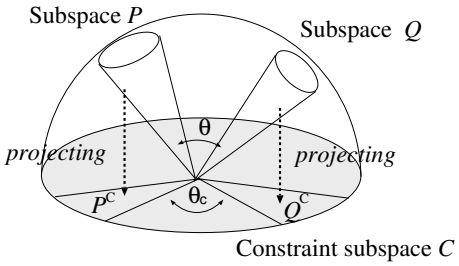


図 1　相互部分空間と制約相互部分空間法の概念図
Fig. 1　Concept of CMSM

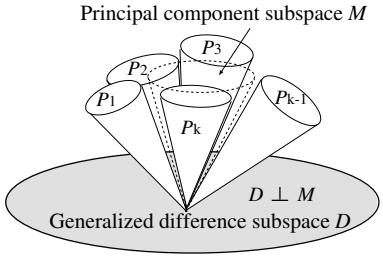


図 2　一般化差分部分空間の概念図
Fig. 2　Concept of generalized difference subspace

$$s_{ij} = \sum_{m=1}^{n} (\mathbf{\Psi}_i \cdot \mathbf{\Phi}_m)(\mathbf{\Phi}_m \cdot \mathbf{\Psi}_j)$$

2.3　部分空間の構造的な類似度

部分空間の構造的な類似性を測る尺度として, 第 n 正準角 (固有値) まで考慮した類似度 $S[n]$ を以下に定義する.

$$S[n] = \sum_{i=1}^{n} \frac{\lambda_i}{n} \tag{3}$$

ここで固有値 λ_i は, 行列 $\mathbf{S}$ の i 番目に大きい固有値である. 類似度 $S[n]$ は, 2 つの部分空間が完全に一致する時に最大値 1.0 となり, 両者が離れるにつれて低下してゆき, 両者が直交する時に最小値 0.0 となる. 上記以外でも固有値の積で定義される類似度も有効である [16].

2.4　制約相互部分空間法

制約部分空間の生成

識別性能を高めるために, 図 1 に示すように, 入力部分空間 $\mathcal{P}$ と辞書部分空間 $\mathcal{Q}$ を識別に有効な成分から成る制約部分空間 $\mathcal{C}$ へ射影し, 射影 $\mathcal{P}^c$ と $\mathcal{Q}^c$ に対して正準角を測る. このように相互部分空間法に制約部分空間への射影を付加した方法を制約相互部分空間法 (Constrained Mutual Subspace Method (CMSM)) [2] と呼ぶ. 制約部分空間とは識別性能を高めるための特徴空間の総称であり, 本稿では一般化差分部分空間を用いる. 一般化差分部分空間 $\mathcal{D}$ は k 個の n 次元クラス部分空間 $\mathcal{P}_i (i = 1 \sim k)$ に対する射影行列 $\mathbf{P}_i$ の総和 $\mathbf{G} (= \sum_{\mathbf{i=1}}^{\mathbf{k}} \mathbf{P_i})$ の固有ベクトルで, 固有値 λ が小さい方から順に選んだ n_c 個のベクトル $\mathbf{d}_{n\times k}, \mathbf{d}_{n\times k-1}, \ldots, \mathbf{d}_{n\times k-n_c+1}$ により張られる.

$$\mathbf{Gd} = \lambda \mathbf{d} \tag{4}$$

一般化差分部分空間 $\mathcal{D}$ の有効性に関しては幾つかの解釈が可能である. 定性的に言えば, 一般化差分部分空間は 2 つのクラス部分空間の差異を表す差分部分空間を, k 個のクラス部分空間に対して一般化した空間で, k 個のクラス部分空間の差異を表す空間となっている [2]. したがって一般化差分部分空間への射影は, 各クラス部分空間からクラス部分空間の違いを表す成分を選択的に抽出することに相当する.

別の見方をすれば, 図 2 に示すように, 一般化差分部分空間は全クラス部分空間の和空間から識別に貢献しない全クラス部分空間の主成分空間 $\mathcal{M}$ を取り除いた空間になっている. 幾何学的には, 空間 $\mathcal{M}$ を取り除くことにより, 一般化差分部分空間へ射影された各クラス部分空間同士の角度が広がり, 射影前に比べて各部分空間の関係は直交状態に近くなる. これは角度基準の識別において好ましい特性である [8]. 射影後の直交度合いは一般化差分部分空間の次元数 n_c に依存しており, 基本的には n_c を予備実験などで得られた最適値に設定するが, これらの最適値は式 (4) における固有値 λ の分布からある程度推定することが可能と考えられる. これについては考察において議論する.

制約部分空間への射影と認識

制約相互部分空間法による物体認識の流れは以下のようになる. まずマルチカメラ [17] や動画像などから得られる見え方パターン分布に対して KL 展開 (平均二乗誤差最小基準) やグラム・シュミットの直交化を適用して入力部分空間 $\mathcal{P}_{in}$ を生成する. 次に入力部分空間 $\mathcal{P}_{in}$ の一般化差分部分空間への射影 $\mathcal{P}_{in}^D$ を求める. ここで m 次元入力部分空間 $\mathcal{P}_{in}$ の一般化差分部分空間 $\mathcal{D}$ への射影 $\mathcal{P}_{in}^D$ は以下の手順で求まる.

(1)　部分空間 $\mathcal{P}_{in}$ を張る m 個の基底ベクトルの制約部分空間上への射影ベクトルを求める.

(2)　各射影ベクトルの長さを 1.0 に正規化する.

(3)　m 個の正規化ベクトルに対してグラム・シュミットの直交化を施す.

(4)　直交化された m 個の正規化ベクトルが射影部分空間 $\mathcal{P}_{in}^D$ の基底ベクトルとなる.

全ての辞書部分空間 $\mathcal{P}_k$ の一般化差分部分空間 $\mathcal{D}$ への射影 $\mathcal{P}_k^D$ も同様の手順で求めておく. 識別では, 入力部分空間 $\mathcal{P}_{in}^D$ と全ての辞書部分空間 $\mathcal{P}_k^D$ との類似度 S(式 (3)) を求め, しきい値以上で最も高い類似度を有する辞書部分空間のクラスとして入力部分空間を識別する.

なお, 上記の処理は, 先に学習パターンを制約部分空間へ射影し, 射影パターンから部分空間を生成する処理に置き換えてもよい.

2.5　直交相互部分空間法

直交相互部分空間法 (OMSM) は, 直交化行列を用いて明示的に各クラス部分空間の関係を直交化したうえで, 相互部分空間法を適用する方法である [4]. 基本的な計算手順は直交部分空間法 [8] と同じであるが, 直交化行列を求めるための学習パターンが異なる. 直交部分空間法では, 各クラスの学習パターンから計算される自己相関行列から直交化行列を計算する. これに対して, OMSM では各クラス部分空間を張る基底ベクト

ルから計算される射影行列を用いる．つまり直交部分空間法では全学習パターンに対して直交化を行うに対して，直交相互部分空間法では，各クラス部分空間を張る基底ベクトルに対して直交化を施すのである．

直交化行列 $\mathbf{O}$ は行列 $\mathbf{G}$ の固有値をすべて 1 にする白色化変換を表す行列として, 次式で与えられる．

$$\mathbf{O} = \mathbf{\Lambda}^{-1/2}\mathbf{B}^{\top} \tag{5}$$

ただし, $\mathbf{\Lambda}^{-1/2}$ は行列 $\mathbf{G}$ の固有値の平方根の逆数を並べた対角行列, $\mathbf{B}$ は行列 $\mathbf{G}$ の固有ベクトルを行に並べた行列, 右上の $\top$ はその転置である．式 (6) により，$\mathbf{O}$ による変換で, 固有値がすべて 1 になることが分かる．ただし，$\mathbf{I}$ は単位行列とする．

$$\mathbf{OGO}^{\top} = \mathbf{\Lambda}^{-1/2}\mathbf{B}^{\top}\mathbf{B\Lambda B}^{\top}\mathbf{B\Lambda}^{-1/2} = \mathbf{I} \tag{6}$$

複数のパターンから生成した入力と辞書部分空間に対して, 部分空間を張る N 本の基底ベクトルを直交化行列 $\mathbf{O}$ により変換し, 更に変換後の N 本のベクトルに Gram-Schmidt の直交化を施す．最終的にこれらの直交化された N 本のベクトルをそれぞれ直交化行列で変換した入力と辞書部分空間の基底ベクトルとする．

3. 非線形識別アルゴリズム

本章ではカーネル関数による非線形写像を用いて CMSM と OMSM の非線形化について説明する．

3.1 カーネル非線形主成分分析の導入

非線形特徴空間 $\mathcal{F}$ 上でカーネル差分部分空間あるいはカーネル直交化行列を形成するための準備として, カーネル非線形主成分分析 [18] について述べる．

f 次元のパターン $\mathbf{x} = (x_1, x_2, \ldots, x_f)^{\top}$ を非線形変換 ϕ により, 原空間に比べて遥かに高い次元 f_ϕ の非線形特徴空間 $\mathcal{F}$ に写像する．

$$\phi : R^f \rightarrow R^{f_\phi},\ \mathbf{x} \rightarrow \phi(\mathbf{x}) \tag{7}$$

これにより線形識別不可の問題を線形識別可能な問題に変換する．空間 $\mathcal{F}$ 上の写像に対して主成分分析や非線形部分空間への射影を行うためには, 写像 $\phi(\mathbf{x})$ と写像 $\phi(\mathbf{y})$ の内積 $(\phi(\mathbf{x})\cdot\phi(\mathbf{y}))$ を計算する必要がある．しかし, 空間 $\mathcal{F}$ 上において, この内積を直接計算することは, 対象とするベクトルの次元が極めて高いために計算困難 (無限次元空間では不可能) となる．ところが非線形変換 ϕ をカーネル関数 $k(\mathbf{x},\mathbf{y})$ を介して定義すると, 内積 $(\phi(\mathbf{x})\cdot\phi(\mathbf{y}))$ は原パターン $\mathbf{x}$ と $\mathbf{y}$ から計算することができる．これが “カーネルトリック” と呼ばれる計算技法である．具体的な非線形変換 ϕ が存在するためには, カーネル関数 $k(\mathbf{x},\mathbf{y})$ が Mercer の条件を満足する必要があり, 例えば以下のような関数が存在する [18].

$$k(\mathbf{x},\mathbf{y}) = \exp(-\frac{||\mathbf{x}-\mathbf{y}||^2}{\sigma^2}) \tag{8}$$

このガウシアン関数を適用した場合には, パターンは無限次元空間へ写像されることになる．

カーネルトリックを用いた非線形特徴空間 $\mathcal{F}$ 上の主成分分析がカーネル非線形主成分分析法である．m 個のパターン $\mathbf{x}_i (i = 1{\sim}m)$ に対する非線形主成分分析は, カーネル関数を介して得られる以下の $m{\times}m$ 行列 $\mathbf{K}$(カーネル行列) の固有値問題に帰着される(注2)．

$$\begin{aligned} \mathbf{Ka} &= \lambda\mathbf{a} \qquad (9)\\ k_{ij} &= (\phi(\mathbf{x}_i)\cdot\phi(\mathbf{x}_j))\\ &= k(\mathbf{x}_i,\mathbf{x}_j) \end{aligned}$$

写像されたベクトル $\phi(\mathbf{x})$ の第 i 非線形主成分ベクトル $\mathbf{e}_i$ への射影成分は次式で計算される．

$$(\phi(\mathbf{x}),\mathbf{e}_i) = \sum_{j=1}^{m} a_{ij}k(\mathbf{x},\mathbf{x}_j) \tag{10}$$

ここで a_{ij} は行列 $\mathbf{K}$ の第 i 番目に大きい固有値 λ_i に対応する固有ベクトル $\mathbf{a}_i$ の第 j 成分である．ただし $\mathbf{a}_i$ は $\lambda_i(\mathbf{a}_i\cdot\mathbf{a}_i)$=1.0 を満足するように基準化されている．

3.2 カーネル差分部分空間の生成

カーネル差分部分空間 $\mathcal{D}_\phi$ の生成, および $\mathcal{D}_\phi$ への射影はベクトル内積で構成されているので, カーネルトリックを用いた非線形化が可能である．基本的な生成の流れは線形一般化差分部分空間 $\mathcal{D}$ と同様であり, 線形クラス部分空間を非線形クラス部分空間に置き換えれば良い．以下では, r 個の d 次元非線形クラス部分空間 $\mathcal{V}_k (k = 1 \sim r)$ から n_k 次元のカーネル差分部分空間 $\mathcal{D}_\phi$ を求める手順を述べる．

クラス k の非線形クラス部分空間 $\mathcal{V}_k$ が m 個の学習パターン $\mathbf{x}_i^k (i = 1 \sim m)$ から生成されているとすると，空間 $\mathcal{V}_k$ を張る d 個の基底ベクトル $\mathbf{e}_i^k (i = 1{\sim}d)$ は次式で表される．

$$\mathbf{e}_i^k = \sum_{j=1}^{m} a_{ij}^k \phi(\mathbf{x}_j^k) \tag{11}$$

ここで a_{ij}^k は式 (10) で示した係数である．同様に他のクラスに対しても非線形クラス部分空間を生成する．

次に r 個の d 次元非線形クラス部分空間の全ての基底ベクトル, つまり合計 $r{\times}d$ 個の基底ベクトルからカーネル差分部分空間の基底ベクトルを求める．これは全ての基底ベクトルをデータと見なして主成分分析を行なうことに相当する．ここで全ての基底ベクトルを列として並べた行列を $\mathbf{E}$ とする．

$$\mathbf{E} = [\mathbf{e}_1^1, \ldots, \mathbf{e}_d^1, \ldots, \mathbf{e}_1^r, \ldots, \mathbf{e}_d^r] \tag{12}$$

以下で定義される行列 $\mathbf{D}$ の固有値問題を解く．

$$\mathbf{Db} = \beta\mathbf{b} \tag{13}$$

$$D_{ij} = (\mathbf{E}[i]\cdot\mathbf{E}[j]), \quad (i,j = 1{\sim}r{\times}d) \tag{14}$$

ここで $\mathbf{E}[i]$ は行列 $\mathbf{E}$ の第 i 列成分を意味する．上式におけるクラス k の第 i 基底ベクトル $\mathbf{e}_i^k$ とクラス k^* の第 j 基底ベクト

(注2)：以下は各パターン $\mathbf{x}$ からそのクラス平均を引いていないので，厳密には平均二乗誤差最小基準による KL 展開の非線形化に相当する．

ル $\mathbf{e}_j^{k^*}$ の内積は，以下のように変形することで，$\mathbf{x}^k$ と $\mathbf{x}^{k^*}$ のカーネル関数値 $k(\mathbf{x}^k, \mathbf{x}^{k^*})$ の線形和として実際に計算できる．

$$(\mathbf{e}_i^k \cdot \mathbf{e}_j^{k^*}) = (\sum_{s=1}^{m} a_{is}^k \phi(\mathbf{x}_s^k) \cdot \sum_{t=1}^{m} a_{jt}^{k^*} \phi(\mathbf{x}_t^{k^*})) \quad (15)$$

$$= \sum_{s=1}^{m}\sum_{t=1}^{m} a_{is}^k a_{jt}^{k^*} (\phi(\mathbf{x}_s^k) \cdot \phi(\mathbf{x}_t^{k^*})) \quad (16)$$

$$= \sum_{s=1}^{m}\sum_{t=1}^{m} a_{is}^k a_{jt}^{k^*} k(\mathbf{x}_s^k, \mathbf{x}_t^{k^*}) \quad (17)$$

カーネル差分部分空間 $\mathcal{D}_\phi$ の第 i 基底ベクトル $\mathbf{d}_i$ は行列 $\mathbf{D}$ の i 番目に小さい固有値 β_i に対応する固有ベクトル $\mathbf{b}_i$ を重み係数として, 以下のように基底ベクトル $\mathbf{E}[j]$ $(j = 1 \sim r{\times}d)$ の線形和で表される．

$$\mathbf{d}_i = \sum_{j=1}^{r\times d} b_{ij} \mathbf{E}[j] \quad (18)$$

ここでベクトル $\mathbf{b}_i$ は $\beta_i(\mathbf{b}_i \cdot \mathbf{b}_i)$=1.0 を満足するように基準化されている．更に $\mathbf{E}[j]$ はクラス $\zeta(j)$ の第 $\eta(j)$ 基底ベクトルとすると，上式は以下のように変形できる．

$$\sum_{j=1}^{r\times d} b_{ij}\mathbf{E}[j] = \sum_{j=1}^{r\times d} b_{ij} \sum_{s=1}^{m} a_{\eta(j)s}^{\zeta(j)} \phi(\mathbf{x}_s^{\zeta(j)}) \quad (19)$$

$$= \sum_{j=1}^{r\times d}\sum_{s=1}^{m} b_{ij} a_{\eta(j)s}^{\zeta(j)} \phi(\mathbf{x}_s^{\zeta(j)}) \quad (20)$$

この基底ベクトル $\mathbf{d}_i$ は実際には計算できないが，次節で述べるように写像ベクトル $\phi(\mathbf{x})$ のこの基底ベクトルに対する射影成分は計算できる．

以下では学習パターンを先にカーネル差分部分空間に射影して，射影された学習パターンから辞書部分空間を生成するという方法について説明するが，先に学習パターンから各非線形クラス部分空間の基底ベクトル (式 (11)) を求めておいて，これらの基底ベクトルをカーネル差分部分空間へ射影し，さらに Gram-Schmidt を適用して辞書部分空間を生成することもできる．

3.3 カーネル差分部分空間への射影計算

前節の準備により, 写像ベクトル $\phi(\mathbf{x})$ のカーネル差分部分空間 $\mathcal{D}_\phi$ の第 i 基底ベクトル $\mathbf{d}_i$ への射影成分は, 入力ベクトル $\mathbf{x}$ と $r{\times}m$ 個の全学習ベクトル $\mathbf{x}_s^k (s = 1{\sim}m, k = 1{\sim}r)$ を用いて次式で計算できる．

$$(\phi(\mathbf{x}) \cdot \mathbf{d}_i) = \sum_{j=1}^{r\times d}\sum_{s=1}^{m} b_{ij} a_{\eta(j)s}^{\zeta(j)} (\phi(\mathbf{x}) \cdot \phi(\mathbf{x}_s^{\zeta(j)})) \quad (21)$$

$$= \sum_{j=1}^{r\times d}\sum_{s=1}^{m} b_{ij} a_{\eta(j)s}^{\zeta(j)} k(\mathbf{x}, \mathbf{x}_s^{\zeta(j)}) \quad (22)$$

したがって, 写像 $\phi(\mathbf{x})$ の $n_k (< r{\times}d)$ 次元のカーネル差分部分空間上への射影 $\tau(\phi(\mathbf{x}))$ の各成分は以下で表される．

$$\tau(\phi(\mathbf{x})) = (z_1,\ z_2, \ldots,\ z_{n_k})^\top \quad (23)$$

$$z_i = (\phi(\mathbf{x}) \cdot \mathbf{d}_i)$$

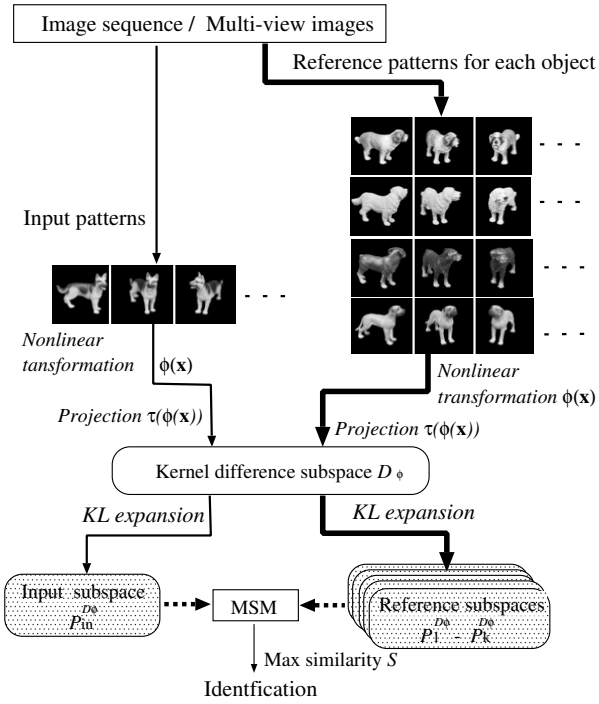


図 3　KCMSM に基づく物体認識の流れ
Fig. 3　Flow of object recognition using KCMSM

3.4 カーネル非線形制約相互部分空間法のアルゴリズム

射影パターン $\tau(\phi(\mathbf{x}))$ に相互部分空間法を適用して, カーネル非線形制約相互部分空間法 (KCMSM) を構築する．

以下では, 図 3 に沿って KCMSM の流れを示す．

（1） まずクラス k に属するパターンのセット $\{\mathbf{x}_1^k, \ldots, \mathbf{x}_m^k\}$ の非線形写像 $\phi(\mathbf{x}_i^k)$ をカーネル差分部分空間 $\mathcal{D}_\phi$ へ射影する．

（2） 次に射影パターンのセット $\{\tau(\phi(\mathbf{x}_1^k)), \ldots, \tau(\phi(\mathbf{x}_m^k))\}$ に KL 展開を適用して線形クラス部分空間 $\mathcal{P}_k^{D_\phi}$ を生成する．具体的には射影パターンのセットから自己相関行列を求め，この固有ベクトルで固有値が大きい方から n 個を，n 次元線形クラス k 部分空間 $\mathcal{P}_k^{D_\phi}$ の基底とする．同様に他のクラス部分空間も求めておく．

（3） 識別時には, 入力パターンのセット $\{\mathbf{x}_1^{in}, \ldots, \mathbf{x}_m^{in}\}$ から同様にカーネル差分部分空間 $\mathcal{D}_\phi$ 上における線形入力部分空間 $\mathcal{P}_{in}^{D_\phi}$ を求める．

（4） $\mathcal{P}_{in}^{D_\phi}$ と各クラス部分空間 $\mathcal{P}_k^{D_\phi}$ の類似度 S(式 (3)) を計算する．全ての類似度の中で, しきい値以上で最も高い類似度に該当するクラス部分空間のクラスを入力パターン分布のクラスとする．

3.5 カーネル非線形直交相互部分空間法

直交相互部分空間法 (OMSM) も CMSM と同様にカーネルトリックを用いて，カーネル非線形直交相互部分空間法 (KOMSM) へと非線形拡張できる．

特徴空間 $\mathcal{F}$ 上で定義される直交化行列をカーネル直交化行列 $\mathbf{O}_\phi$ と名付ける．カーネル直交化行列 $\mathbf{O}_\phi$ の第 i 行ベクトル $\mathbf{O}_{\phi_i}$ は，式 (18) に行列 $\mathbf{D}$ の固有値 β を考慮して次のように表される．

$$\mathbf{O}_\phi i = \frac{\mathbf{d}_i}{\sqrt{\beta_i}} \tag{24}$$

$$= \sum_{j=1}^{r\times d} \frac{b_{ij}}{\sqrt{\beta_i}} \mathbf{E}[j] \tag{25}$$

このカーネル直交化行列 $\mathbf{O}_\phi$ により写像 $\phi(\mathbf{x})$ の直交化されたパターン $\chi(\phi(\mathbf{x}))$ は以下で表される．

$$\chi(\phi(\mathbf{x})) = (\chi_1,\ \chi_2, \ldots,\ \chi_{r\times d})^\top \tag{26}$$

$$\chi_i = (\phi(\mathbf{x}) \cdot \mathbf{O}_{i\phi})$$

KOMSM の大まかな流れは以下のようになる．

（1） クラス k に属するパターンセット $\{\mathbf{x}_1^k, \ldots, \mathbf{x}_m^k\}$ の非線形写像 $\phi(\mathbf{x}_i^k)$ をカーネル直交化行列 $\mathbf{O}$ を用いて変換する．

（2） 次に (1) で変換されたパターンのセット $\{\chi(\phi(\mathbf{x}_1^k)), \ldots, \chi(\phi(\mathbf{x}_m^k))\}$ に KL 展開を適用して線形クラス部分空間 $\mathcal{P}_k^{O_\phi}$ を生成する．具体的には変換されたパターンのセットから自己相関行列を求め，この固有ベクトルで固有値が大きい方から n 個を，n 次元の線形クラス k 部分空間 $\mathcal{P}_k^{O_\phi}$ の基底とする．同様に他のクラス部分空間も求めておく．

（3） 識別時には, 入力パターンのセット $\{\mathbf{x}_1^{in}, \ldots, \mathbf{x}_m^{in}\}$ を直交変換したパターンから線形入力部分空間 $\mathcal{P}_{in}^{O_\phi}$ を求める．

（4） $\mathcal{P}_{in}^{O_\phi}$ と各クラス部分空間 $\mathcal{P}_k^{O_\phi}$ の成す正準角を類似度として求める．全ての類似度の中で, しきい値以上で最も高い類似度に該当するクラス部分空間のクラスを入力パターン分布のクラスとする．

なお，上記の処理 (1)-(3) は，KCMSM と同様に，先に学習パターンから各非線形クラス部分空間の基底ベクトル (式 (11)) を求めておいて，$\phi(\mathbf{x})$ の替わりにこれらの基底ベクトルを直交化行列を用いて変換する処理に置き換えてもよい．

4. 比較実験

相互部分空間法 (MSM)，制約相互部分空間法 (CMSM)，直交相互部分空間法 (OMSM)，核非線形相互部分空間法 (KMSM) [14]，カーネル非線形制約相互部分空間法 (KCMSM)，カーネル非線形直交相互部分空間法 (KOMSM) の性能を多視点画像と正面顔画像の 2 種類のデータベースを用いて比較した．

4.1 3次元物体識別 (実験 1)

3 次元モデルを複数視点から撮影した公開データベース (ETH-80 [20]) から，図 4 に示す形状が極めて類似している 30 モデルを抜き出して用いた． 各モデル毎に図 5 に示す 41 視点から撮影された画像セットが用意されている．視点は全てのモデルに対して同じである．これらの内, 奇数枚目 (21 枚) を各クラス部分空間，差分部分空間 $\mathcal{D}$, カーネル差分部分空間 $\mathcal{D}_\phi$, 直交化行列 $\mathbf{O}$, カーネル直交化行列 $\mathbf{O}_\phi$ の学習用データ, 偶数枚目 (20 枚) を評価データとした．よって学習視点と評価視点は異なる．評価データについては，20 枚の中から i から $i+9$ までの 10 枚を取り出して一つのデータセットとし, これを開始フレーム i を 1 から 10 まで変化させて合計 10 個の評価セットを用意した．したがって全試行回数は 9000(=10×30×30) 回である．

評価尺度は入力モデルが正しいモデルとして識別される識別率 (%), 正解モデルに対する類似度と他モデルに対する類似度の分離度，および等価エラー率 (EER:Equal Error Rate) を用いた．分離度は 1.0 に正規化された量で, 1.0 に近い程, クラス識別能力が高いことを示している [1]．EER は FAR(他人受理誤り率) と FRR(本人排除誤り率) が一致する際の誤り率である．

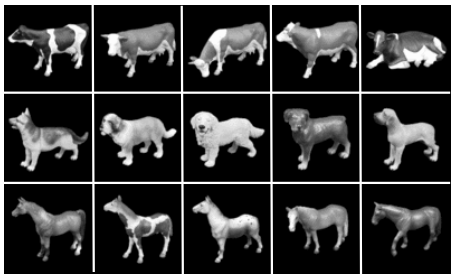


図 4 評価データの一部：上段:cow, 中段:dog, 下段:horse, 各クラスの 10 モデルのうち 5 モデルを表示
Fig. 4 Evaluation data, Top: cow, Middle: dog, Low: horse

実験パラメータは以下の通りである．評価には 180 ピクセル ×180 ピクセルのカラー画像を 15 ピクセル ×15 ピクセルのモノクロ画像に変換したものを用いた．よってデータの次元数 f は 225(=15×15) である．入力部分空間 $\mathcal{P}_{in}^D$, $\mathcal{P}_{in}^{D_\phi}$, $\mathcal{P}_{in}^O$, $\mathcal{P}_{in}^{O_\phi}$, およびクラス部分空間 $\mathcal{P}_k^D$, $\mathcal{P}_k^{D_\phi}$, $\mathcal{P}_k^O$, $\mathcal{P}_k^{O_\phi}$ の次元数は全て 7 とした．ただし, $\mathcal{P}_{in}^D$ と $\mathcal{P}_k^D$ は, 非線形識別の生成手順と合わせるために, 一般化差分部分空間へ射影されたパターンに対して KL 展開を適用して生成した．$\mathcal{P}_{in}^O$ と $\mathcal{P}_k^O$ も同様に先にパターンを直交行列を用いて変換してから生成した．MSM,KMSM の入力およびクラス部分空間の次元数も 7 とした．

一般化差分部分空間 $\mathcal{D}$ および直交化行列 $\mathbf{O}$ は全モデル (30 個) の 20 次元クラス部分空間から生成した. 一般化差分部分空間の 次元数 n_c は 190 から 215 の範囲で変化させた．カーネル差分部分空間 $\mathcal{D}_\phi$ およびカーネル直交化行列 $\mathbf{O}_\phi$ は全モデル (30 個) の 20 次元非線形クラス部分空間から生成した．カーネル差分部分空間の次元数 n_k は 50 から 550 の範囲で変化させた．非線形識別の全方法において，カーネル関数として式 (8) で定義されたガウシアン関数 ($\sigma^2 = 0.05$) を用いた．

4.2 実験結果

(1) 識別率の比較

表 1 と表 2 に各識別方式の識別率と分離度, 表 3 に EER を示す.

表中の表示は, 方式-差分部分空間 (カーネル差分部分空間) の次元数を表している．例えば KCMSM-550 は 550 次元のカーネル差分部分空間を用いたカーネル非線形制約相互部分空間法を意味する．また S[1]〜S[4] は式 (3) で定義した類似度であり, 括弧内の数字は用いる正準角の個数を示している．

まず全般的な知見として, 線形識別法 (MSM, CMSM,OMSM) に対して非線形識別法 (KMSM, KCMSM, KOMSM) の識別性能が大きく向上しており，多視点画像識別において線形識別法では最良の性能を得ることが難しいことが分かる．ここで OMSM の識別性能が極端に悪いのは，各クラス部分空間に重なりが生じて，共通部分空間が存在することに一因があると思

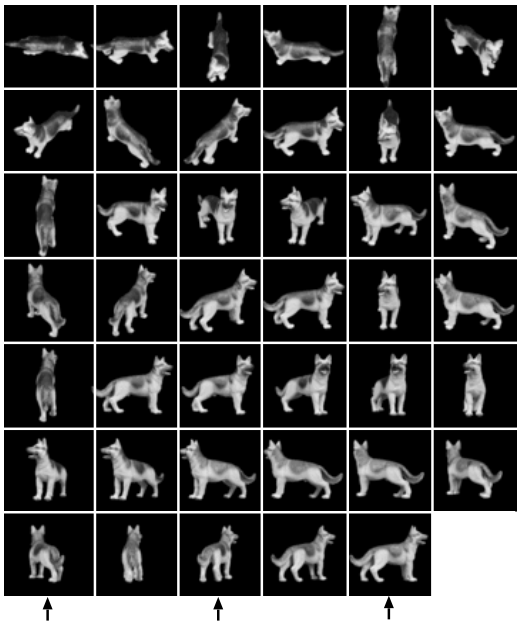


図 5　dog1 の全データ: 矢印で指した列が学習用画像
Fig. 5　All view-patterns of dog1

表 1　各方式の識別率 (%):下線の値は各方式の最良値 (実験 1)
Table 1　Recognition rate of each method (%)

	S[1]	S[2]	S[3]	S[4]
MSM	72.7	73.7	**76.3**	74.3
CMSM-215	75.7	**81.3**	76.3	73.7
CMSM-200	73.3	81.0	79.3	77.7
CMSM-190	71.0	73.0	73.0	75.0
OMSM	51.3	54.0	**56.0**	54.0
KOMSM	85.3	87.3	88.0	**88.0**
KMSM	84.7	**87.0**	82.0	81.7
KCMSM-550	83.0	85.3	85.7	86.3
KCMSM-500	79.3	85.0	87.0	87.0
KCMSM-450	82.0	88.0	89.3	**89.7**
KCMSM-400	83.3	87.7	88.3	89.7
KCMSM-300	81.0	87.7	88.7	89.0
KCMSM-200	81.7	81.7	83.3	83.3
KCMSM-100	57.7	62.7	68.0	65.3
KCMSM-50	36.0	35.7	29.0	29.0

われる．重なりが無い場合には，行列 **G**(式 (4)) はデータ次元と同じ 225 個の正の固有値を持つが，実際には 211 個の正の固有値しか求まらなかった．

各方式の最良識別性能（表中の下線を引いた値）を見ると，MSM から KMSM へ非線形化することで識別率が 76.3%から 87.0%, 分離度が 0.082 から 0.429 と大きく改善された．また CMSM から KCMSM への非線形化でも, 識別率が 81.3%から 89.7% へ, 分離度が 0.257 から 0.621 へと大きく改善された．更に KMSM と KCMSM を比較すると, KCMSM では識別率で約 3%, 分離度で約 0.2 の改善が見られた．特に分離度が高いということは, 対象物の数が多くなった場合でも, KCMSM の方が高い識別性能を維持できる可能性を示唆している．KOMSM についても同様で OMSM の識別率を大幅に改善して，KCMSM

表 2　各方式の分離度:下線の値は各最良値 (実験 1)
Table 2　Separability of each method

	S[1]	S[2]	S[3]	S[4]
MSM	0.055	0.074	**0.082**	0.080
CMSM-215	0.203	0.236	0.242	0.236
CMSM-200	0.215	**0.257**	0.254	0.245
CMSM-190	0.229	0.255	0.249	0.244
OMSM	0.039	0.095	0.116	**0.116**
KOMSM	0.537	0.612	0.620	**0.621**
KMSM	0.375	0.420	0.420	**0.429**
KCMSM-550	0.538	0.581	0.584	0.538
KCMSM-500	0.556	0.607	0.616	0.612
KCMSM-450	0.549	0.618	0.621	**0.621**
KCMSM-400	0.529	0.601	0.607	0.609
KCMSM-300	0.483	0.536	0.545	0.545
KCMSM-200	0.340	0.385	0.403	0.408
KCMSM-100	0.141	0.194	0.212	0.213
KCMSM-50	0.057	0.080	0.085	0.089

表 3　各方式の ERR (実験 1)
Table 3　Recognition rate of each method (%)

Linear methods	MSM	CMSM	OMSM
EER(%)	12.0	9.5	25.0
Kernel methods	KMSM	KCMSM	KOMSM
EER(%)	6.0	4.0	4.0

と同程度の高い識別性能を実現した．これらの傾向は表 3 の EER にも同様に現れている．直交化を実現する原理が異なる KCMSM と KOMSM が，同程度の識別性能を実現しているのは大変興味深い．

次に複数の正準角を用いる有効性を見る． KCMSM と KOMSM では正準角の個数が増えるに従って識別性能が向上した．これは, ある視点から見たパターンが似ていても, 他の視点から見たパターンは異なるという特性をうまく利用した結果である．ただし, 同じ非線形識別の KMSM では, 分離度は向上するものの, 識別率は逆に低下してしまった．直交化が, 類似度 $S[n](n\geqq 2)$ の有効性を引き出したと言える．

(2) クラス部分空間の成す平均角度

各方法の識別性能は各クラス部分空間同士の成す角度と密接に関係している．そこで，各クラス部分空間の関係を表す指標として，平均直交度を $1.0-cos^2\hat{\theta}$ と定義する．ここで $\hat{\theta}$ は各クラス部分空間の成す平均正準角である．平均直交度が 1.0 に近い程，各カテゴリ部分空間同士が直交関係に近いことを意味する．

図 6 の CMSM は，一般化差分部分空間へ射影された 30 個の 7 次元線形クラス部分空間の平均直交度の変化を示している．横軸は一般化差分部分空間およびカーネル差分部分空間の次元数，縦軸は平均直交度を示している．図中の KCMSM は，カーネル差分部分空間への射影された 30 個の非線形クラス部分空間の平均直交度の変化を示している．

先ず CMSM についてみる．射影を伴わない原空間 (225 次元) における平均直交度は 0.160 である．これに対して 55 次元の主成分部分空間を取り除いた 170 次元では 0.725 まで改善さ

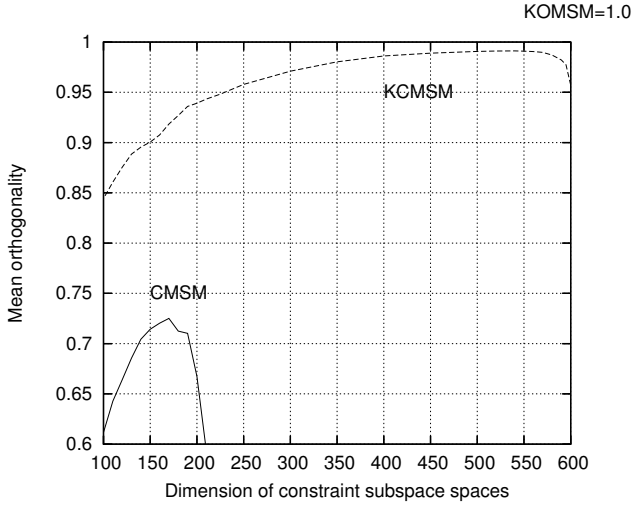


図 6　制約部分空間の次元に対する各クラス部分空間の平均直交度：直交相互部分空間とカーネル直交相互部分空間法の平均直交度はそれぞれ 0.611 と 1.0 である．
Fig. 6　Change of orthogonality between class subspaces

れた．ここれに対して OMSM の直交度は 0.611 と CMSM より低くなっている．原理的には CMSM より高い直交度を実現できるはずであるが，これは先に述べたように，クラス部分空間に共通部分空間が存在することが影響していると考えられる．

次に KCMSM をみる．カーネル差分部分空間は 30 個の 20 次元非線形部分空間から生成したので，カーネル差分部分空間の次元は 600 次元まで設定可能である．ただし，600 次元の場合には射影を行わない核非線形相互部分空間法 (KMSM) に一致する．

450 次元のカーネル差分部分空間へ射影することで直交度は最大 0.991(角度では 84.5 度) まで向上した．これに対して KOMSM では完全な直交化 (1.0，角度では 90.0 度) を実現したが，両者の差異は小さい．この事は非線形識別においては，両者の識別性能が同程度であるという実験結果ともうまく対応している．この対応の中で，平均直交度が最大となる次元数と識別性能が最大となる次元数がほぼ一致していることは，平均直交度が識別性能を予測する一つの指標となるということを示唆している．

4.3　正面顔の識別 (実験 2)

照明条件を 10 種類に変化させて撮影した 50 人分の顔画像を用いて評価実験を行った．被験者には顔向きを軽く上下左右に振るように指示した．入力画像のサイズは 320 ピクセル ×240 ピクセルであり，これから目鼻の位置情報を用いて，15 ピクセル ×15 ピクセルの正規化顔パターンを抽出した [19]．50 人の内，25 人分のデータを，各クラス部分空間，差分部分空間 $\mathcal{D}$, カーネル差分部分空間 $\mathcal{D}_\phi$, 直交化行列 $\mathbf{O}$, カーネル直交化行列 $\mathbf{O}_\phi$ の生成に使用した．各人物のデータ数は 500 枚 (=照明 10 種類× 50 枚) とした．残りの 25 人分のデータは評価として用いて，異なる照明条件間で総当りの識別を行った．総試行回数は 18468 回である．評価データについては，各人物の各照明条件のデータ (130〜170 枚) から i から $i+9$ までの 10 枚を取り出して一つのデータセットとし, これを開始フレーム i を 1 から 10 まで変化させて複数の評価セットを用意した．

表 4　各方式の識別率 (%) と ERR (実験 2)
Table 4　Recognition rate of each method (%)

Linear	MSM	CMSM-200	OMSM
Recognition rate (%)	91.74	91.3	97.09
EER(%)	12.0	7.5	6.3
Nonlinear	KMSM	KCMSM-450	KOMSM
Recognition rate (%)	91.15	97.40	97.42
EER(%)	11.0	4.3	3.5

全ての方法において，入力部分空間とクラス部分空間の次元数を 7 とした．$\mathcal{P}^D_{in}$，$\mathcal{P}^D_k$，$\mathcal{P}^O_{in}$，および $\mathcal{P}^O_k$ は非線形識別の生成手順と合わせるために, 一般化差分部分空間へ射影あるいは直交行列を用いて変換されたパターンから生成した．

一般化差分部分空間 $\mathcal{D}$ および直交化行列 $\mathbf{O}$ は 25 個 (25 人分) の 60 次元クラス部分空間から生成した．カーネル差分部分空間 $\mathcal{D}_\phi$ およびカーネル直交化行列 $\mathbf{O}\phi$ は 25 個の 60 次元非線形クラス部分空間から生成した．カーネル差分部分空間の次元数 n_k は 200 から 1500 の範囲で変化させた．非線形識別の全方法において，カーネル関数はガウシアン関数 ($\sigma^2 = 1.0$) を用いた．

表 4 は各方式の第 1 位識別率と EER を示している．非線形識別法の有効性は先の実験結果と同じであるが，この問題では線形直交相互部分空間法の高い識別率が目立つ．これは，先の 3 次元物体識別の問題に比べると，顔画像パターン分布の非線形性はそれほど強くなく，線形識別の枠組みが十分に有効であるからと思われる．

5.　考　　察

5.1　計算量の比較

計算量に関しては，制約相互部分空間法の方が有利であり，特にカーネル関数を用いた非線形識別においては，その優位性は高くなる．カーネル非線形直交相互部分空間法では，行列 $\mathbf{K}$ の全ての固有ベクトルと固有値を求める必要がある．ところが，$\mathbf{K}$ のサイズはクラスと学習パターンの数に比例して大きくなり, その固有値ベクトルと固有値を計算することが困難になる．これに対して，制約相互部分空間法では主成分部分空間を張る固有値が大きい側の固有ベクトルを求めればよい．一般化差分部分空間への射影はこれらの固有ベクトルの射影から間接的に求まる．

5.2　一般化差分部分空間の最適次元の推定

OMSM と KOMSM はパラメータ選定が不要であるが，CMSM と KCMSM は一般化差分部分空間およびカーネル差分部分空間の次元数を指定する必要がある．両者の識別性能はこれらの次元設定にそれほど敏感ではないが，最大性能を引き出すためには最適次元の予測が重要な課題である．

これまでの議論から，各クラス部分空間の平均直交度が最大となる次元数を最適値とするのは自然であろう．直交相互部分空間法においては，各クラス部分空間の基底ベクトル集合を白色化，つまり全方向の分散を均一にする操作により直交化を実

現している．この事に注目すれば，和空間から主成分部分空間を取り除いた残りの一般化差分部分空間の全基底ベクトル集合において，その分散がきるだけ均一であることが望ましい．一般化差分部分空間へ射影されたクラス部分空間の基底ベクトル集合の分散は，行列 **G** の固有値の分布から計算できるので，これらの固有値を用いて分散の均一性を測り，最適次元を推定できる可能性がある．

6. ま と め

制約相互部分空間法と直交相互部分空間法をクラス部分空間の直交化という観点から比較した．

線形識別の枠組みにおいては，直交相互部分空間法の優位性が見られた．しかしながら，パターン分布の非線形性が強く，各クラス部分空間が共通部分空間を持つ場合には，直交化行列による変換は有効に働かず，むしろ識別性能の低下を引き起こす場合もあることに注意が必要である．これに対して制約相互部分空間法では，共通部分空間は自動的に削除されるので，この影響は小さい．

カーネル関数を用いた非線形識別では，線形識別では大きかった両者の差異が小さくなり，直交化の原理が異なるにも拘わらず，同程度の極めて高い識別性能を実現した．ただし，計算量については制約相互部分空間法の方が少ないために，クラス数や各クラス部分空間の次元数が大きな問題に対しては，制約相互部分空間法がやや有利になると思われる．

文　献

[1] 福井 和広, 山口 修, 鈴木 薫, 前田 賢一, “制約相互部分空間法を用いた環境変動にロバストな顔画像認識 - 照明変動を抑える制約部分空間の学習 -,” 信学論 (D-II), vol.J82-D-II, no.4,pp.613-620,1999.

[2] 福井 和広, 山口 修, “一般化差分部分空間に基づく制約相互部分空間法,” 信学論 (D-II), vol.J87-D-II, no.8, pp.1622-1631, 2004.

[3] K. Fukui, 0. Yamaguchi, “ Face recognition using multi-viewpoint patterns for robot vision, ”11th International Symposium of Robotics Research (ISRR'03), pp.192-201, springer, 2005.

[4] 河原 智一, 西山 正志, 山口 修, ”直交相互部分空間法を用いた顔認識,”, 情処研報 CVIM-151,pp.17-24,2005.

[5] 前田 賢一, 渡辺 貞一, “局所的構造を導入したパターン・マッチング法,” 信学論 (D), vol.J68-D, no.3, pp.345-352, 1985.

[6] S. Watanabe, N. Pakvasa, “Subspace method of pattern recognition,” Proc. 1st Int. J. Conf. on Pattern Recognition, 1973.

[7] 飯島 泰蔵, “パターン認識,” 電気・電子工学大系 43, コロナ社, 1973.

[8] エルッキ・オヤ著, 小川 英光, 佐藤 誠 訳, “パターン認識と部分空間法,” 産業図書,1986.

[9] 前田 英作, 村瀬 洋, “カーネル非線形部分空間法によるパターン認識,” 信学論 (D-II), vol.J82-D-II, no.4, pp.600-612, 1999.

[10] 津田 宏治, “ヒルベルト空間における部分空間法,” 電子信学論 (D-II), vol.J82-D-II, no.4, pp.592-599, 1999.

[11] 福井 和広, 山口 修, ”カーネル非線形制約相互部分空間法による物体認識,”, 電子情報通信学会論文誌 (D-II), vol.J88-D-II, no.8, pp.1349-1356, 2005.

[12] K. Fukui, B. Stenger, O. Yamaguchi, ”A framework for 3D object recognition using the kernel constrained mutual subspace method”, ACCV06, part-I, pp.315-324, 2006.

[13] 福井 和広, 山口 修, ”部分空間法の理論拡張と物体認識への応用,”, 情報処理学会論文誌コンピュータビジョンとイメージメディア, vol.46, No.SIG 15 (CVIM 12), pp.21-34, 2005.

[14] 坂野 鋭, 武川 直樹 , 中村 太一, “核非線形相互部分空間法による物体認識”, 信学論 (D-II),vol.J84-D,No.8,pp.1549-1556,2001.

[15] F. Chatelin, “行列の固有値,” 伊理 正夫, 伊理 由実訳, シュプリンガー・フェアラーク東京, 1993.

[16] 前田 賢一, 山口 修, 福井 和広, ”部分空間の正準角を利用した3次元パターンマッチングの一基礎検討ー顔と写真との区別を目的として,”, 電子情報通信学会論文誌 (D-II), vol.J89-D-II, no.6, pp.1288-1289, 2006.

[17] 小坂谷 達夫, 山口 修, 福井 和広, “マルチカメラを用た顔画像認識システム,” 画像センシングシンポジウム, 2002.

[18] B. Schölkopf, A. Smoa, and K.-R. Müller,“Nonlinear principal component analysis as a kernel eigenvalue problem, ” Neural Computation, Vol. 10, pp.1299-1319, 1998.

[19] 福井 和広, 山口 修, “形状抽出とパターン照合の組合せによる顔特徴点抽出,” 信学論 (D-II), vol.J80-D-II, no.8,pp.2170-2177, 1997.

[20] B. Leibe and B. Schiele, “Analyzing Appearance and Contour Based Methods for Object Categorization,”, CVPR2003, pp.409-415, 2003.

Approximation of spherical PCA by Euclideanization

Jun FUJIKI† and Shotaro AKAHO†

† National Institute of Advanced Industrial Science and Technology, Tsukuba-Central 2, 1–1–1 Umezono, Tsukuba-shi, Ibaraki 305–0035 Japan

Abstract To measure the similarity between two high dimensional vector data, correlation coefficient is often used instead of Euclidean distance. In this case, the high dimensional vectors are normalized as hyperspherical points and the distance between two hyperspherical data is measured as the length along geodesic. On hypersphere, the Pythagorean theorem does not hold and it makes difficult to apply data analysis methods defined in Euclidean space, such as principal component analysis, to hyperspherical data. In this paper, we propose spherical principal component analysis to analyze hyperspherical data on the criterion of spherical least squares. Spherical principal component analysis is defined as low dimensional great hypersphere fitting to the high dimensional hyperspherical data. We also propose the approximation of spherical principal component analysis because spherical principal component analysis is a kind of non-linear minimization.

Key words spherical data, hypersphere fitting, least squares, Euclideanization, principal component analysis, equi-directional projection

1. Introduction

Recently, data analysis on hypersphere is getting of great significance.

In computer vision, omnidirectional camera such as catadioptric camera and fish-eye camera is widely used for robot navigation, surveillance system and so on. To understand and to unify omnidirectional cameras, spherical camera is defined [2], [4], [8], [10], [13], [15], [17], [18]. On spherical camera, feature points are represented as the points on 2-dimensional sphere, and feasure lines are represented as the great circle of 2-dimensional sphere, which are "lines" in spherical geometry.

In computer vision, camera motions are also identified as a sequence of spherical points which are corresponding to the front directions of cameras. Since we do not need to consider the horizontal directions of cameras and the distances from cameras to an object by rotating and scaling of the size of images. Therefore, smoothing the camera motion is realized by fitting a small or great circle to an sequence of spherical points [6], [7], [14].

In finance engineering, the BGM model [3] plays important roles in interest model. The model is a kind of LIBOR market model, which applies the London interbank offered rate (LIBOR) in reality. In the BGM model uses the multi-dimensional Brownian motion derived from multi-dimensional normal distribution, and the correlation matrix plays important role to describe Brownian motion. We regard each component of correlation matrix as the inner product of two hyperspherical data. To reduce the computational cost of simulating the multi-dimensional Brownian motion, dimension reduction of the correlation matrix is an important process [12]. Dimension reduction of correlation matrix is equivalent to the fitting a great hypersphere to the data on the unit hypersphere.

In the fields of bioinformatics and data mining, when analyzing gene expression profile data and processing text data, we can regard the data as hyperspherical points [16]. These examples show that data analysis on hypersphere is getting of great significance recently.

In this paper, we propose spherical principal component analysis to analyze hyperspherical data on the criterion of spherical least squares. Spherical principal component analysis is defined as low dimensional great hypersphere fitting to the high dimensional hyperspherical data. We also propose the approximation of spherical principal component analysis because spherical principal component analysis is a kind of non-linear minimization.

We also refer two methods to reduce the dimension of data space, that is, extracting a low-dimensional small hyperspherical structure embedded in high-dimensional hypersphere. The former method is called Spherical Least Squares by Euclideanization for StereoGraphic projection [9], and the latter method is called Sequential Dimension Reduction [9].

2. Spherical PCA (SPCA)

One of the purpose of **Principal component analysis (PCA)** is compressing of data dimension to remove the noise on the data. Then **spherical PCA (SPCA)** should be defined as the data compression method for the hyperspherical data. From this point of view, SPCA is the low dimensional fitting to the high dimensional hyperspherical data.

2.1 Spherical Least Squares (SLS)

Because the data are on the hypersphere, the fitting error should be defined by the function of not the Euclidean distance but the distance along geodesic, which is equivalent to the angle, between the data and fitting space.

The distance between the hyperspherical data $\boldsymbol{x}_i$ and some specific subset of hypersphere S is defined as

$$\mathrm{dist}(\boldsymbol{x}_i, S) = \min_{\boldsymbol{y} \in S} \cos^{-1}(\boldsymbol{x}_i^\top \boldsymbol{y}). \tag{1}$$

When the best fitting is defined as the minimizer of the sum of square distance, the fitting error is defined as

$$\sum \{\mathrm{dist}(\boldsymbol{x}_i, S)\}^2. \tag{2}$$

We call the minimization of sum of square of not the Euclidean distance but the distance along geodesic of hypersphere as **Spherical Least Squares**(**SLS**).

2.2 Spherical PCA (SPCA)

In spherical geometry, a line on the sphere is equivalent to a great circle of the sphere. Then **Spherical PCA (SPCA)** is defined as *low dimensional great hypersphere fitting* to the hyperspherical data.

In the low dimensional great hypersphere fitting, we should pay attention to the fact as follows:

Fact: *Let S^i and S^j $(i < j)$ be the best i-dimensional great hypersphere fitting and j-dimensional great hypersphere fitting, respectively. Generally,*

$$S^i \not\subset S^j \tag{3}$$

holds.

Figure 1 is the example of SPCA, that is great hypersphere fitting of the data and the blue circle is the result of 1-dimensional great hypersphere fitting. Red point is the result of 0-dimensional great hypersphere fitting, that is, the center of the data which minimizes the square distance along geodesic between data.

It is easy to find that the center is not on the best S^1 fitting, that is

$$S^0 \not\subset S^1. \tag{4}$$

On usual PCA in the Euclidean space, The Pythagorean theorem ensures the best i-dimensional hyperplane fitting is always belongs to the best j-dimensional hyperplane fitting when $i < j$ by using the criterion of least squares. However, the Pythagorean theorem does not holds on the sphere by using the distance along geodesic, then the fact holds.

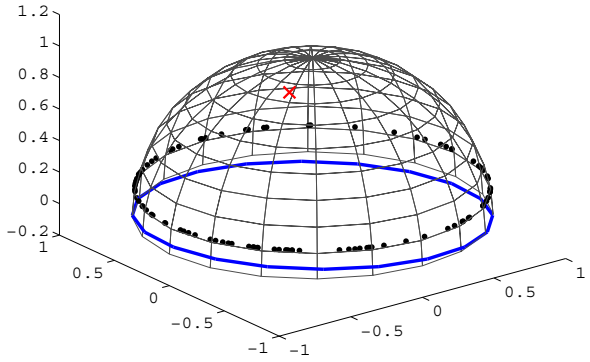


Fig. 1 S_0 fitting (red) does not belongs to S_1 fitting (blue).

3. Hypersphere fitting

For SPCA, it is enough to consider the great hypersphere fitting, but also consider the small hypersphere fitting in the paper. Of course the great hypersphere fitting is the special case of the small hypersphere fitting.

3.1 Representation of small hypersphere

Let $\boldsymbol{R}^{n+1}$ be an $n+1$-dimensional Euclidean space, and O be its origin. We define the n-dimensional unit hypersphere as

$$S^n = \{\mathrm{X} \,|\, \mathrm{OX} = \|\boldsymbol{x}\| = 1,\ \boldsymbol{x} \in \boldsymbol{R}^{n+1}\}. \tag{5}$$

We consider a d-dimensional small hypersphere on the unit hypersphere S^n. Generally, hypersphere is represented as the intersection between S^n and a $d+1$-dimensional Euclidean space $\boldsymbol{E}^{d+1}$.

Let C be the center of the small hypersphere, OC is perpendicular to $\boldsymbol{E}^{d+1}$ (when $\mathrm{O} \in \boldsymbol{E}^{d+1}$, $\mathrm{C} = \mathrm{O}$).

Let $\overrightarrow{\mathrm{OC}} = \boldsymbol{c}$, the radius of the small hypersphere is computed as

$$\alpha = \sqrt{1 - ||\boldsymbol{c}||^2}. \tag{6}$$

Let one of the orthonormal basis of associated linear space $\boldsymbol{E}^{d+1}$ be $\langle \boldsymbol{r}_1, \ldots, \boldsymbol{r}_{d+1} \rangle$, and

$$\underset{(n+1 \times d+1)}{R} = (\, \boldsymbol{r}_1 \quad \cdots \quad \boldsymbol{r}_{d+1} \,) \tag{7}$$

be the matrix arraying the orthonormal basis (Generally, orthonormal coordinate matrix R is not unique to represent the same Euclidean space).

Euclidean space $\boldsymbol{E}^{d+1}$ is parameterized as

$$\boldsymbol{E}^{d+1} = \left\{ \mathrm{X} \,\middle|\, \mathrm{X} = \mathrm{C} + R\boldsymbol{t},\ \boldsymbol{t} \in \boldsymbol{R}^{d+1} \right\}. \tag{8}$$

Therefore, the d-dimensional small hypersphere, which is the

intersection between the unit hypersphere S^n and $\boldsymbol{E}^{d+1}$ is parameterized as

$$\left\{ \mathrm{X} \,\middle|\, \mathrm{X} = \mathrm{C} + R\boldsymbol{t}, \boldsymbol{t} \in \boldsymbol{R}^{d+1}, ||\boldsymbol{t}|| = \alpha \right\}. \tag{9}$$

The small hypersphere is described only by the C and R, but we also use radius α to represent the small hypersphere as $\alpha S^d(\mathrm{C}, R)$ for convenience. When we consider only its radius α, we abbreviate the representation to αS^d.

Since $\alpha S^d(\mathrm{C}, R)$ is a d-dimensional small hypersphere with radius α and its center is $\sqrt{1-\alpha^2}$ far from the origin O, $\alpha S^d(\mathrm{C}, R)$ is represented as

$$\begin{cases} x_1^2 + x_2^2 + \cdots + x_{d+1}^2 = \alpha^2, \\ x_{d+2} = \cdots = x_n = 0, \\ x_{n+1} = \sqrt{1-\alpha^2} \end{cases} \tag{10}$$

when we choose an appropriate coordinate system. We call this representation as the **normal form of small hypersphere**.

3.2 Representation of great hypersphere

We consider a d-dimensional great hypersphere $1S^d = S^d$ on the unit hypersphere S^n.

The same as previous subsection, the d-dimensional great hypersphere is parameterized as

$$\left\{ \mathrm{X} \,\middle|\, \mathrm{X} = \mathrm{O} + R\boldsymbol{t}, \boldsymbol{t} \in \boldsymbol{R}^{d+1}, ||\boldsymbol{t}|| = 1 \right\}, \tag{11}$$

which is described only by the R.

The **normal form of great hypersphere** is

$$\begin{cases} x_1^2 + x_2^2 + \cdots + x_{d+1}^2 = 1 \\ x_{d+2} = \cdots = x_n = x_{n+1} = 0 \end{cases} \tag{12}$$

when we choose an appropriate coordinate system.

3.3 Hypersphere fitting under SLS

The aim of the paper is dimension reduction of the data distributed on the unit hypersphere S^n on $\boldsymbol{R}^{n+1}$, and compress the data to the data on the great and/or small hypersphere $\alpha S^d(\mathrm{C}, R)$.

On small hypersphere fitting, C $(,\alpha)$ and R are chosen so as to minimize the square distance along geodesic. And on great hypersphere fitting, only R is chosen so as to minimize the square distance along geodesic.

Let $\boldsymbol{z}_f$ $(f = 1, \ldots, F)$ be data on S^n, there hold $||\boldsymbol{z}_f|| = 1$ $(f = 1, \ldots, F)$.

Let r_f be the distance from $\boldsymbol{z}_f$ to $\alpha S^d(\mathrm{C}, R)$ along the geodesic on S^n, there holds

$$r_f = \cos^{-1}(\boldsymbol{z}_f^\top \boldsymbol{c} + \alpha ||R^\top \boldsymbol{z}_f||), \tag{13}$$

because

$$\begin{aligned} \cos r_f &= \max_{\mathrm{X} \in \alpha S^d(\mathrm{C},R)} \boldsymbol{z}_f^\top \boldsymbol{x} = \max_{||\boldsymbol{t}||=\alpha} (\boldsymbol{z}_f^\top \boldsymbol{c} + \boldsymbol{z}_f^\top R\boldsymbol{t}) \\ &= \boldsymbol{z}_f^\top \boldsymbol{c} + \alpha ||R^\top \boldsymbol{z}_f|| \end{aligned} \tag{14}$$

(The maximum holds when $\boldsymbol{t} = \dfrac{\alpha}{||R^\top \boldsymbol{z}_f||} R^\top \boldsymbol{z}_f$).

To estimate the best small and/or great hypersphere under least squares along geodesic is to minimize

$$J = \frac{1}{2} \sum_{f=1}^{F} r_f^2, \tag{15}$$

which is a function of C $(,\alpha)$ and R. This is the hypersphere fitting under SLS.

However, it is not easy to solve the hypersphere fitting under SLS because the differentiation of $||R^\top \boldsymbol{z}_f||$ with respect to R is needed to apply the Newton's method for minimizing the equation (15) or some other minimizing technique computing gradient of the cost function. Therefore, we propose the approximations of SLS.

4. Euclideanization

In this paper, we propose a method to compute an approximation of SLS, called **Euclideanizatoin** of hypersphere. Euclideanization is the adjustment of the metric of projected space to keep the metric of the original space as well as possible.

4.1 Example to understand Euclideanization

In this subsection, we present an example to understand the general case of Euclideanization, we discuss the case $n = 2$ and $d = 1$, that is, fitting small circle to the data on the sphere. We use stereographic projection to map the surface of the sphere to Euclidean space.

The characteristics of the data are as follows: we regard the true small circle as latitude of the globe. We determine the arc of the small circle by setting some longitude range (the angle unit is radian) on the latitude. We generate 100-data from uniform distribution on the arc of the small circle and adding the Gauss noise of 0.01-radian standard deviation along meridional direction. We determine $\boldsymbol{z}_f$ as rotating the 100-data by three-dimensional rotation matrix. We estimate the best small circle from the 100-data under some criterion.

The results are shown as the true small circle as an latitude for convenience. the dotted line represents the true small circle and the solid line represents the estimated small circle.

4.2 Hyperplane fitting

The easiest way to estimate small hypersphere is as follows: Fitting $d+1$-dimensional hyperplane to $\boldsymbol{z}_f$ by least square criterion, and the estimated small hypersphere is the intersection between the hyperplane and the unit hypersphere (Gray et al. [11] uses this estimation as the initial value of their algorithm).

Figure 2 shows the estimation of the small circle by plane fitting. When a longitude range is small, the estimation tends toward the tangent plane of the unit sphere because

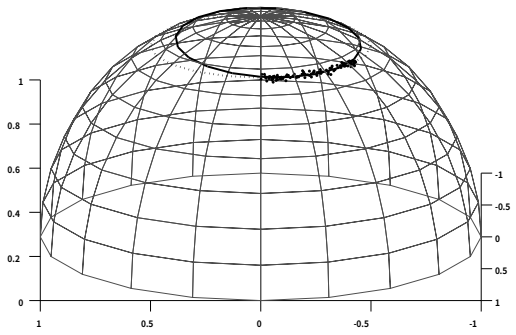


Fig. 2 Hyperplane fitting: latitude $\pi/3$, longitude range $\pi/3$.

the sphere itself is approximated by the tangent plane.

4. 3 Estimation by stereographic projection

To overcome the problem that an estimation tends toward a tangent hyperplane, we try to estimate a small circle by stereographic projection. Stereographic projection has a property that the small circle on the unit sphere is projected to the circle on the projective plane. Therefore, we estimate the small circle by fitting the circle on the projective plane under least square criterion and project the circle to the unit sphere by stereographic projection. Figure 3 shows the es-

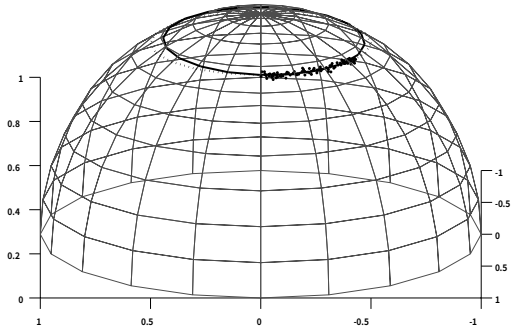


Fig. 3 Estimation by stereographic projection: latitude $\pi/3$, longitude range $\pi/3$

timation by stereographic projection for the same data as Figure 2. By stereographic projection, the tendency toward a tangent hyperplane is reduced.

Let N($\boldsymbol{n}$) be a fixed point chosen from n-dimensional hypersphere S^n and S($-\boldsymbol{n}$) be its antipode. Stereographic projection is a mapping from n-dimensional hypersphere S^n onto $\Pi^n : \boldsymbol{n}^\top \boldsymbol{x} = -1$ which is a tangent n-dimensional hyperplane at S($-\boldsymbol{n}$). By Stereographic projection, a point Z($\boldsymbol{z}$) $\in S^n$ on the n-dimensional hypersphere S^n is projected to a point Z$'$($\boldsymbol{z}'$) which is the intersection between line NZ and Π^n.

Let $\boldsymbol{y} = \boldsymbol{z} - \boldsymbol{n}$ and $\boldsymbol{y}' = \boldsymbol{z}' - \boldsymbol{n}$, the stereographic projection is simply represented as $\boldsymbol{y}' = 4\boldsymbol{y}/||\boldsymbol{y}||^2$, which is the inversion with respect to inversion hypersphere with center N and radius 2 in the $n+1$-dimensional Euclidean space. By this inversion, d-dimensional hypersphere on the unit hypersphere S^n is mapped to d-dimensional hypersphere or d-dimensional hyperplane (see **appendix A**) on the tangent hyperplane Π^n.

The estimation referred in this subsection is as follows: first, fitting d-dimensional small hypersphere or d-dimensional hyperplane on the n-dimensional projective hyperplane under least square criterion. Next, mapping the fitted d-dimensional small hypersphere or d-dimensional hyperplane onto the n-dimensional hypersphere S^n by the inversion.

4. 4 Euclideanization for Stereographic projection(ESG)

The estimation referred in the previous subsection sometimes gives a poor estimation. Then, we propose the weighted least squares where the weights are determined by the changing of metric under stereographic projection. Simply speaking, the enlargement rate of some point on the hypersphere by stereographic projection is k, the weights are determined as k^{-2}. Like this, the geodesic distances are approximately replaced with weighted Euclidean distance. And the weights are determined by changing of metric[1]. We call the replacing as **Euclideanization**.

In this subsection, we propose a method to estimate the approximation of SLS by **Euclideanization for Stereographic projection** (**ESG**). We call the method the **SLS-ESG**.

First, Figure 4 shows the estimation by ESG for the same data as Figure 2. For the data, the estimation by ESG is almost the same as the least squares without weights on the projective plane (Figure 3).

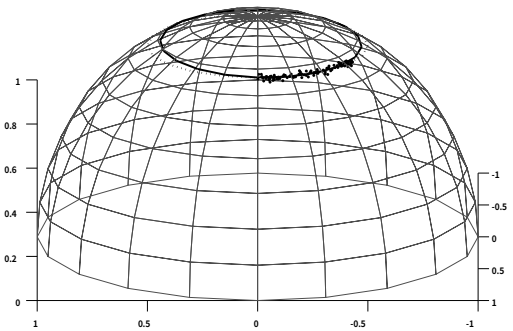


Fig. 4 Estimation by ESG: latitude $\pi/3$, longitude range $\pi/3$

However, Figure 5 shows a poor estimation by least squares on the projective plane without weights. The reason of such a poor estimation is that the existence of data around the north pole. The enlargement rate of the area by stereographic projection is getting higher when the area is getting closer to

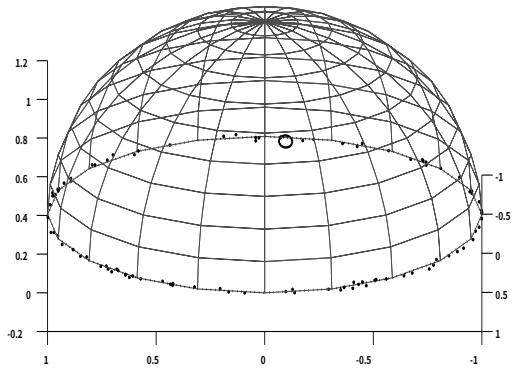


Fig. 5 Estimation by stereographic projection and usual least squares, latitude 0, longitude range 2π

the north pole from the south pole. When we apply ordinary least squares to the data on the projective plane, the estimation tends to the data far from the origin of the projective plane (the image of the south pole).

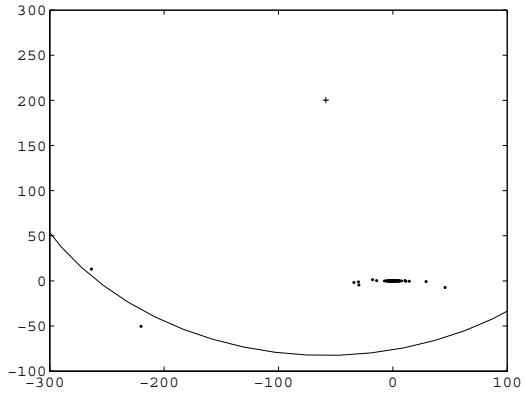


Fig. 6 Representation of ordinary least squares estimation on the projective plane, latitude 0, longitude range 2π

Figure 6 shows the data on the projective plane and estimated circle by ordinary least squares. we can see that the estimation is tends to the data far from the origin of the projective plane.

To overcome this problem, we set weights to revise the metric (enlargement by stereographic projection).

Let r be the distance from the origin of the projective plane to the datum, the weight of the datum are determined by $(1 + r^2/4)^{-2}$ because there holds $dr = (1 + r^2/4)d\theta$ from figure 7.

Figure 8 and 9 show that the estimation by SLS-ESG is improved.

This is why we propose the Euclideanization to adjust the metric.

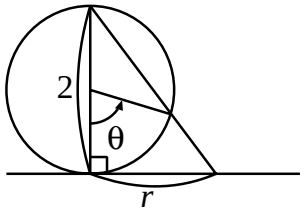


Fig. 7 Changing of metric by stereographic projection

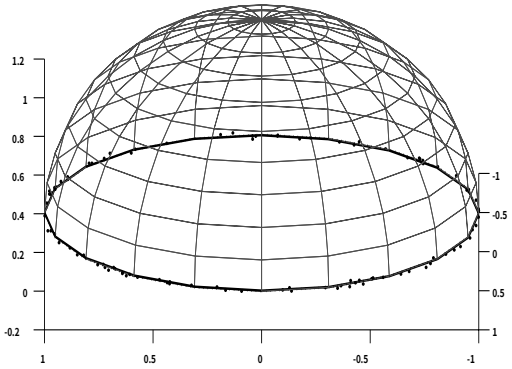


Fig. 8 Estimation by SLS-ESG, latitude 0, longitude range 2π

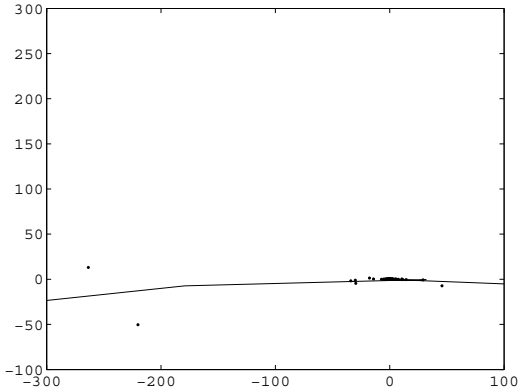


Fig. 9 Representation of SLS-ESG estimation on the projective plane, latitude 0, longitude range 2π

5. Equi-directional projection

Many of the projection from hypersphere surface to its Euclidean tangent space are unified by **equi-directional projection** of hypersphere.

Let $\widetilde{\boldsymbol{o}}_{\pm}$ be defined as $\widetilde{\boldsymbol{o}}_{\pm} = \begin{pmatrix} \mathbf{0}_n \\ \pm 1 \end{pmatrix}$, that is north and/or south pole of S^n, and let $\widetilde{\boldsymbol{x}} = \begin{pmatrix} \boldsymbol{x} \\ x_{n+1} \end{pmatrix}$.

Let $\boldsymbol{E}^n_-$ be the n dimensional Euclidean tangent space at $\widetilde{\boldsymbol{o}}_-$.

The map from $S^n - \{\widetilde{\boldsymbol{o}}_+\}$ to $\boldsymbol{E}^n_-$ is called equi-directional projection of hypersphere if and only if the map is represented as

$$\widetilde{\boldsymbol{x}} \mapsto f(x_{n+1})\boldsymbol{x} = f(x_{n+1})\sqrt{1-x_{n+1}^2}\cdot\frac{\boldsymbol{x}}{||\boldsymbol{x}||} \tag{16}$$

where $f(x)$ satisfies

$$f(-1) = 1, \quad f(x) > 0 \tag{17}$$

and

$$\frac{d}{dx}\{f(x)\sqrt{1-x^2}\} = \frac{f'(x)(1-x^2) - xf(x)}{\sqrt{1-x^2}} > 0, \tag{18}$$

that is,

$$\frac{f'(x)(1-x^2)}{f(x)} > x. \tag{19}$$

Let r be the distance between south Pole and the projection point, there holds

$$r = f(x_{n+1})\sqrt{1-x_{n+1}^2} \tag{20}$$

and the condition Eq.19 is the monotonous increasing of r.

5.1 Volume element

To compute the weights of Euclideanization, we should pay attention to the changing of volume element by the equi-directional projection of the hypersphere. The ratio of volume element is equivalent to the Jaccobian of the mapping.

On the hypersphere, the n-dimensional volume element V_1 is computed as the n-dimensional volume of parallel pipe organized by $\left\{\frac{\partial\widetilde{\boldsymbol{x}}}{\partial x_i}\right\}_{i=1}^{n}$.

There hold

$$\frac{\partial\widetilde{\boldsymbol{x}}}{\partial\boldsymbol{x}} = \begin{pmatrix} \mathrm{I}_n \\ -x_{n+1}^{-1}\boldsymbol{x}^\top \end{pmatrix} \tag{21}$$

and

$$\det\begin{pmatrix} \frac{\partial\widetilde{\boldsymbol{x}}}{\partial\boldsymbol{x}} & \boldsymbol{p} \end{pmatrix} = \begin{pmatrix} x_{n+1}^{-1}\boldsymbol{x} \\ 1 \end{pmatrix}^\top \boldsymbol{p}, \tag{22}$$

for any vector $\boldsymbol{p}$, V_1 is computed as

$$V_1 = \left\| \begin{pmatrix} x_{n+1}^{-1}\boldsymbol{x} \\ 1 \end{pmatrix} \right\| = |x_{n+1}|^{-1}. \tag{23}$$

On the projection space by the equi-directional projection, the n-dimensional volume element V_2 is computed as the n-dimensional volume of parallel pipe organized by $\left\{\frac{\partial f(x_{n+1})\widetilde{\boldsymbol{x}}}{\partial x_i}\right\}_{i=1}^{n}$.

There holds

$$\frac{\partial f(x_{n+1})\boldsymbol{x}}{\partial\boldsymbol{x}} = f(x_{n+1})\mathrm{I}_n - \frac{f'(x_{n+1})}{x_{n+1}}\boldsymbol{x}^\top\boldsymbol{x}, \tag{24}$$

then the volume element V_2 is computed as

$$\begin{aligned} V_2 &= \left|\det\frac{\partial f(x_{n+1})\boldsymbol{x}}{\partial\boldsymbol{x}}\right| \\ &= \{f(x_{n+1})\}^n\left|1 - \frac{f'(x_{n+1})}{x_{n+1}f(x_{n+1})}||\boldsymbol{x}||^2\right| \\ &= \{f(x_{n+1})\}^n\left|1 - \frac{f'(x_{n+1})(1-x_{n+1}^2)}{x_{n+1}f(x_{n+1})}\right|. \end{aligned} \tag{25}$$

By considering the V_1 and V_2, the Jaccobian of the projection is computed as

$$\begin{aligned} J &= \frac{V_2}{V_1} \\ &= \{f(x_{n+1})\}^n\left|x_{n+1} - \frac{f'(x_{n+1})(1-x_{n+1}^2)}{f(x_{n+1})}\right| \\ &= \{f(x_{n+1})\}^n\left\{\frac{f'(x_{n+1})(1-x_{n+1}^2)}{f(x_{n+1})} - x_{n+1}\right\} \end{aligned} \tag{26}$$

(because of Eq.19). Then the length is expected to enlarged

$$J^{\frac{1}{n}} = f(x_{n+1})\left\{\frac{f'(x_{n+1})(1-x_{n+1}^2)}{f(x_{n+1})} - x_{n+1}\right\}^{\frac{1}{n}} \tag{27}$$

times by the equi-directional projection. This means the weight of Euclideanization for the equi-directional projection is

$$J^{-\frac{1}{n}} = \frac{1}{f(x_{n+1})}\left\{\frac{f'(x_{n+1})(1-x_{n+1}^2)}{f(x_{n+1})} - x_{n+1}\right\}^{-\frac{1}{n}}. \tag{28}$$

5.2 Orthographic projection

By orthographic projection, there hold

$$f(x) = 1, \quad f'(x) = 0, \tag{29}$$

then the range to satisfy Eq.17 and Eq.19 is $x_{n+1} < 0$ and the weights are computed by

$$J^{-\frac{1}{n}} = (-x_{n+1})^{-\frac{1}{n}}. \tag{30}$$

Rewriting $J^{-\frac{1}{n}}$ by

$$\begin{aligned} r &= f(x_{n+1})\sqrt{1-x_{n+1}^2} = \sqrt{1-x_{n+1}^2} \\ &\Longleftrightarrow x_{n+1} = -(1-r^2)^{\frac{1}{2}}, \end{aligned} \tag{31}$$

the representation of weights by r are

$$J^{-\frac{1}{n}} = (1-r^2)^{-\frac{1}{2n}}. \tag{32}$$

5.3 Stereographic projection

By stereographic projection, there hold

$$f(x) = \frac{2}{1-x}, \quad f'(x) = \frac{2}{(1-x)^2} \tag{33}$$

then the range to satisfy Eq.17 and Eq.19 hold any x_{n+1}, and the weights are computed by

$$J^{-\frac{1}{n}} = \frac{1-x}{2}. \tag{34}$$

Rewriting $J^{-\frac{1}{n}}$ by

$$\begin{aligned} r &= f(x_{n+1})\sqrt{1-x_{n+1}^2} = 2\sqrt{\frac{1+x_{n+1}}{1-x_{n+1}}} \\ &\Longleftrightarrow x_{n+1} = \frac{1+r^2/4}{1-r^2/4} \end{aligned} \tag{35}$$

the representation of weights by r are

$$J^{-\frac{1}{n}} = \left(\frac{r^2}{4}+1\right)^{-1}. \tag{36}$$

5.4 Azimuthal equidistant projection

By azimuthal equidistant projection, there hold

$$f(x) = \frac{\pi - \cos^{-1} x}{\sqrt{1-x^2}}, \tag{37}$$

$$f'(x) = \frac{1}{1-x^2} + \frac{x(\pi - \cos^{-1} x)}{\sqrt{1-x^2}^3} \tag{38}$$

then the range to satisfy Eq.17 and Eq.19 hold any x_{n+1}, and the weights are computed by

$$J^{-\frac{1}{n}} = \left(\frac{\pi - \cos^{-1} x}{\sqrt{1-x^2}} \right)^{-\frac{n-1}{n}} \tag{39}$$

Rewriting $J^{-\frac{1}{n}}$ by

$$r = f(x_{n+1})\sqrt{1-x_{n+1}^2} = \pi - \cos^{-1} x$$
$$\Longleftrightarrow x_{n+1} = -\cos r \tag{40}$$

the representation of weights by r are

$$J^{-\frac{1}{n}} = \left(\frac{\sin r}{r} \right)^{\frac{n-1}{n}}. \tag{41}$$

5.5 Gnomic projection

By gnomic projection, there hold

$$f(x) = -\frac{1}{x}, \quad f'(x) = \frac{1}{x^2} \tag{42}$$

then the range to satisfy Eq.17 and Eq.19 is $x_{n+1} < 0$ and the weights are computed by

$$J^{-\frac{1}{n}} = (-x_{n+1})^{\frac{n+1}{n}} \tag{43}$$

Rewriting $J^{-\frac{1}{n}}$ by

$$r = f(x_{n+1})\sqrt{1-x_{n+1}^2} = \frac{\sqrt{1-x_{n+1}^2}}{-x_{n+1}}$$
$$\Longleftrightarrow x_{n+1} = -(1+r^2)^{-\frac{1}{2}} \tag{44}$$

the representation of weights by r are

$$J^{-\frac{1}{n}} = (r^2+1)^{-\frac{n+1}{2n}}. \tag{45}$$

5.6 Equisolidangle projection

Equisolidangle projection does not change the volume element, there holds $J = 1$, then weights are constant 1. The representation of equisolidangle projection is computed in Appendix B.

6. Euclideanization for equi-directional projection

In this subsection, we compare the estimation of SPCA with Euclideanization to without Euclideanization. we discuss the case $n = 2$ and $d = 1$, that is, fitting great circle to the data on the sphere.

The characteristics of the data are as follows: we regard the true great circle as equator of the globe. We generate 100-data from the great circle as their longitudes are distributed from Gauss distribution of its mean and standard deviation are 0 and ω-radian, respectively. We add the Gauss noise of σ-radian standard deviation along x, y and z-axes, and normalize the data as their norm are equal to 1. We determine $\boldsymbol{z}_f$ as rotating the 100-data by three-dimensional rotation matrix. We estimate the best great circle fitting (S^1 fitting) and the best point fitting (S^0 fitting) from the 100-data under some criterion.

Figure 10∼17 show the S^0 and S^1 fitting error. Fitting error is the average of 100-trial. In the graphs, "1" denotes orthographic projection, "2" denotes orthographic projection cut of at $\pi/3$, "3" denotes stereographic projection, "4" denotes azimuthal equidistant projection, "5" denotes gnomic projection and "6" denotes equisolidangle projection. Orthographic projection cut of at $\pi/3$ means using only each datum of its colatitude is less than $\pi/3$, that is, using only each datum of its latitude is greater than $\pi/6$.

Blue bar in the graphs represents error of estimation with Euclideanization and red bar in the graphs represents error of estimation without Euclideanization.

When the data are distributed short range of equator (Fig. 10∼ Fig. 13), Euclideanization yields worse S^0 fitting than without Euclideanization (Fig. 10 and Fig. 11).

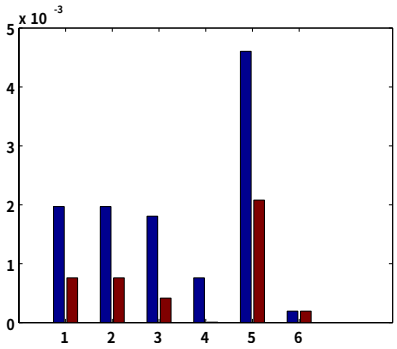


Fig. 10 S^0 fitting error; $\omega = 0.25$, $\sigma = 0.05$.

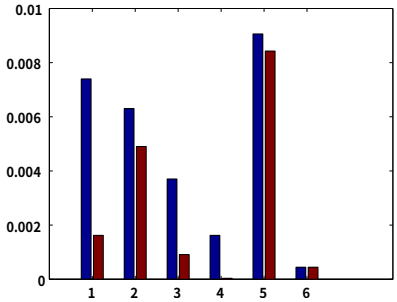


Fig. 11 S^0 fitting error; $\omega = 0.25$, $\sigma = 0.2$.

However, the S^0 fitting errors are sufficiently small and we can say the estimation with Euclideanization and without

Euclideanization are almost the same.

Figure 12 and Fig. 13 show that it makes little difference between the estimation with Euclideanization and without Euclideanization in S^1 fitting.

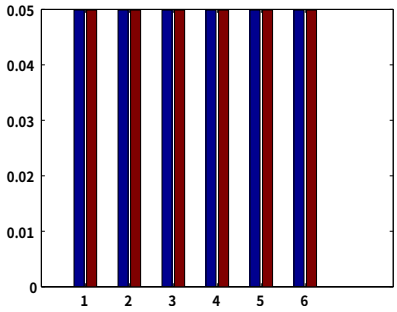


Fig. 12 S^1 fitting error; $\omega = 0.25$, $\sigma = 0.05$.

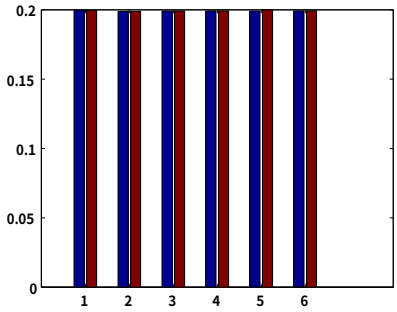


Fig. 13 S^1 fitting error; $\omega = 0.25$, $\sigma = 0.2$.

When the data are distributed wide range of equator, (Fig. 14$\sim$ Fig. 17),

Euclideanization yields better S^0 fitting than without Euclideanization except orthogonal projection (Fig. 14 and Fig. 15).

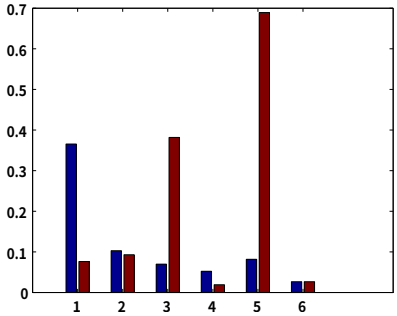


Fig. 14 S^0 fitting error; $\omega = 1$, $\sigma = 0.05$.

Especially, Euclideanization for stereographic projection and for gnomic projection yield good performance. The feature of these two projection is the enlargement rate $J^{\frac{1}{n}}$ is getting to infinity when the point is getting far from the south pole. This means the slight noise of the points far from the south pole is enlarged to large noise and it yields

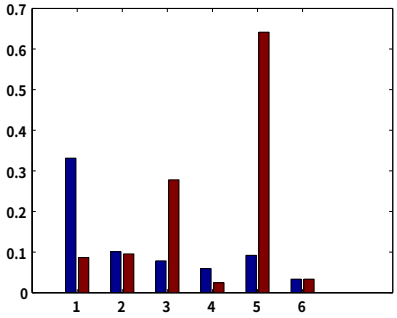


Fig. 15 S^0 fitting error; $\omega = 1$, $\sigma = 0.2$.

worse estimation without Euclideanization. In this case, Euclideanization is very important for estimation.

For orthogonal projection, Euclideanization yields worse S^0 fitting than without Euclideanization. The feature of the orthogonal projection is the enlargement rate $J^{\frac{1}{n}}$ is getting to 0 when the point is getting far from the south pole. This means the large noise of the points far from the south pole is neglected and it yields worse estimation with Euclideanization. In this case, Euclideanization is not suitable for estimation.

Figure 16 and Fig. 17 show that it makes little difference between the estimation with Euclideanization and without Euclideanization in S^1 fitting except without Euclideanization for stereographic projection. The reason of the worse estimation is not sure, but Euclideanization gives better performance for stereographic projection.

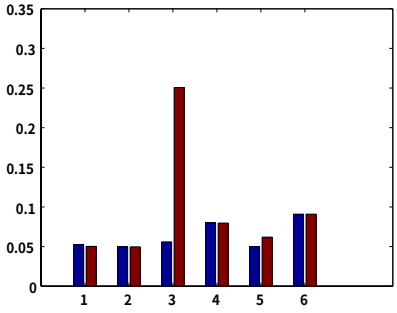


Fig. 16 S^1 fitting error; $\omega = 1$, $\sigma = 0.05$.

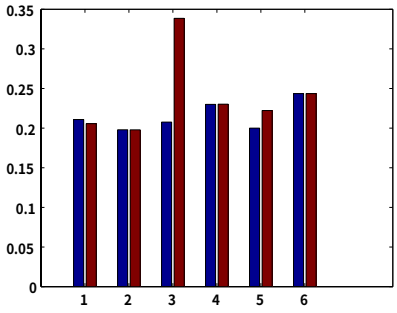


Fig. 17 S^1 fitting error; $\omega = 1$, $\sigma = 0.2$.

From Fig. 18 to Fig. 23 show the S^0 estimation and S^1 estimation of the data when the data are distributed wide range of equator. In these figure, red lines denote estimation of SLS by Newton's method, green lines denote estimation of equi-directional projection without Euclideanization, and blue lines denote estimation of equi-directional projection with Euclideanization. From these figures, Euclideanization derives stable estimation for all equi-directional projection.

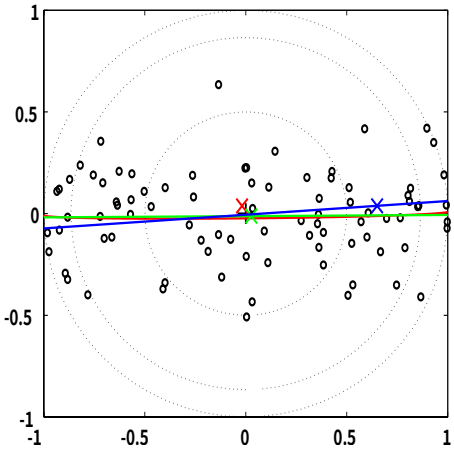


Fig. 18 Orthographic; $\omega = 1$, $\sigma = 0.2$.

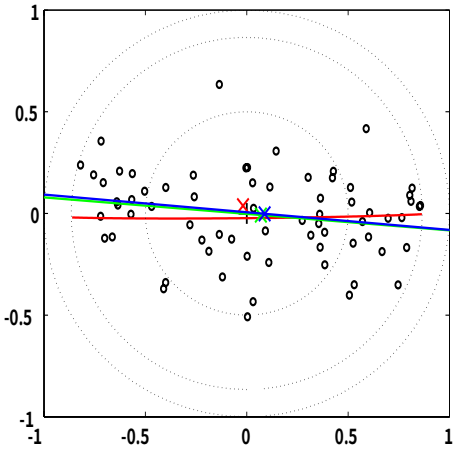


Fig. 19 Orthographic with cutoff at $\pi/3$; $\omega = 1$, $\sigma = 0.2$.

7. Euclideanization for Stereographic projection (ESG)

In this section, we highlight to **Euclideanization for Stereographic projection (ESG)** because stereographic projection has a property that the small hypersphere on the unit hypersphere is projected to the hypersphere on the projective hyperplane. Euclidean space is regarded as hypersphere of its radius is infinity, then many data analysis method for Euclidean space is applicable for hypersphere by applying these data analysis method to the projective hyperplane.

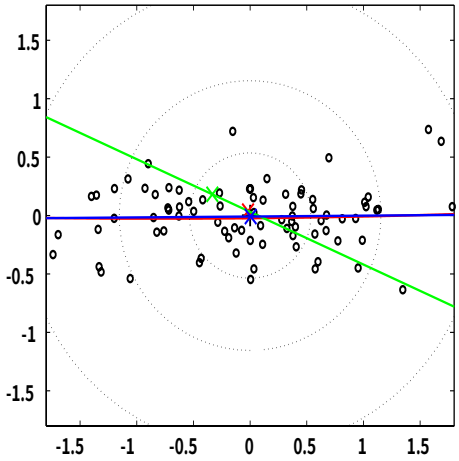


Fig. 20 Stereographic; $\omega = 1$, $\sigma = 0.2$.

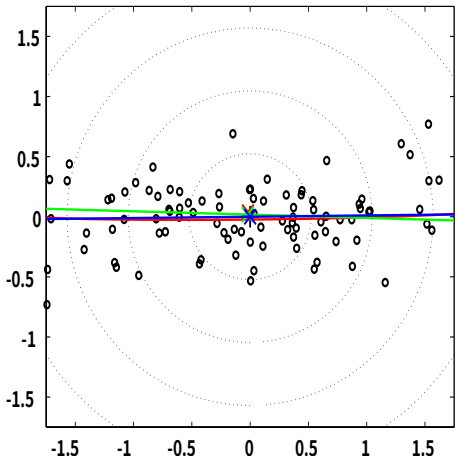


Fig. 21 Azimuthal equidistant; $\omega = 1$, $\sigma = 0.2$.

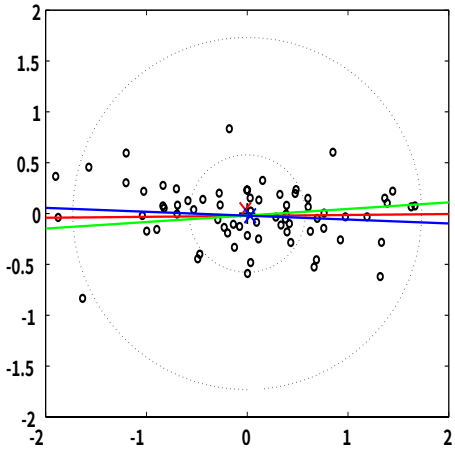


Fig. 22 Gnomic; $\omega = 1$, $\sigma = 0.2$.

Then, the approximation of SPCA is realized by using Euclidean PCA to the projective hyperplane. we call the method **spherical principal component analysis by Euclideanization for stereographic projection (SPCA-ESG)**.

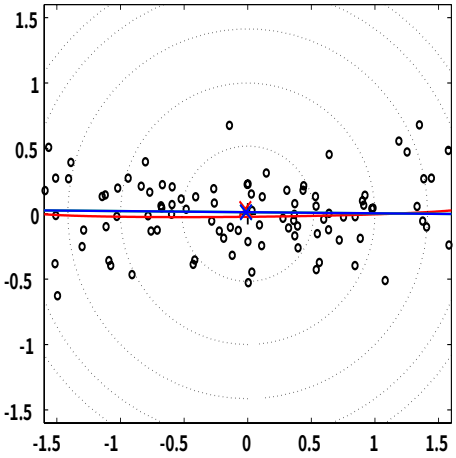


Fig. 23 Equisolidangle; $\omega = 1$, $\sigma = 0.2$.

7.1 The algorithm of SPCA-ESG

The algorithm of SPCA-ESG under the criterion of SLS (SPCA-ESG-SLS) is as follows:

(1) Define the fitting dimension d.

(2) Map the data on the unit hypersphere S^n onto the data on the projective plane Π^n by stereographic projection.

(3) Fit $d+1$-dimensional Euclidean space $\boldsymbol{E}^{d+1}$ to the data on the projective plane Π^n by weighted least squares.

(4) Project the data on Π^n onto $\boldsymbol{E}^{d+1}$ by orthographic projection.

(5) Fit d-dimensional hypersphere for the data on $\boldsymbol{E}^{d+1}$ by weighted least squares.

(6) Map the d-dimensional hypersphere on Π^n onto S^n by stereographic projection, which is the estimated small hypersphere αS^d.

(7) Compute the estimation of $\boldsymbol{z}_f$ by projecting the data $\boldsymbol{z}_f$ onto αS^d.

Note that the projection of $\boldsymbol{z}_f$ onto αS^d is the mapping

$$\boldsymbol{z}_f \quad \mapsto \quad \boldsymbol{c} + \frac{\alpha}{||R^\top \boldsymbol{z}_f||} R R^{\mathrm{T}} \boldsymbol{z}_f,$$

that is the replacing $\boldsymbol{z}_f$ to a point on αS^d which minimizes the distance along geodesic between $\boldsymbol{z}_f$ and the point.

8. Sequential Dimension Reduction (SDR)

8.1 One-dimension reduction

The dimension of the orthogonal complement of R of $n-1$-dimensional hypersphere $\alpha S^{n-1}(\mathrm{C}, R)$ on S^n is one.

Let the normal basis of the orthogonal complement be $\boldsymbol{\lambda}$ and

$$\psi_f = \cos^{-1}(\boldsymbol{z}_f^\top \boldsymbol{\lambda}), \quad \mu = \cos^{-1}||\boldsymbol{c}||, \tag{46}$$

there holds

$$r_f = \psi_f - \mu. \tag{47}$$

Then the cost function J is minimized by Newton's method.

8.2 Newton's method

In this subsection, we consider the minimization of non-negative function J parameterized by $\boldsymbol{\alpha}$.

Let the values of J and $\boldsymbol{\alpha}$ at the n-th iteration as J_n and $\boldsymbol{\alpha}_n$ respectively.

The second order Taylor expansion of J around J_n yields

$$J(\Delta\boldsymbol{\alpha}) = J_n + \boldsymbol{q}_n \Delta\boldsymbol{\alpha} + \frac{1}{2}\Delta\boldsymbol{\alpha}^\top H_n \Delta\boldsymbol{\alpha}, \tag{48}$$

$$\boldsymbol{q}_n = \left.\frac{\partial J}{\partial \boldsymbol{\alpha}}\right|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}_n}, \quad H_n = \left.\frac{\partial^2 J}{\partial \boldsymbol{\alpha}\partial \boldsymbol{\alpha}^\top}\right|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}_n}, \tag{49}$$

where $\Delta\boldsymbol{\alpha} = \boldsymbol{\alpha} - \boldsymbol{\alpha}_n$.

We minimize J under constraint $g = 0$. Let the values of g at the n-th iteration as g_n.

The first order Taylor expansion of g around g_n yields

$$g(\Delta\boldsymbol{\alpha}) = g_n + \boldsymbol{b}_n^\top \Delta\boldsymbol{\alpha} = 0, \quad \boldsymbol{b}_n = \left.\frac{\partial g}{\partial \boldsymbol{\alpha}}\right|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}_n}. \tag{50}$$

Then we apply the Lagrange multiplier method for J with Lagrange multiplier κ as $E = J + \kappa g$, that is, we solve the equations

$$\frac{\partial E}{\partial(\Delta\boldsymbol{\alpha})} = \boldsymbol{q}_n + H_n \Delta\boldsymbol{\alpha} + \kappa \boldsymbol{b}_n = 0, \tag{51}$$

$$\frac{\partial E}{\partial \kappa} = g_n + \boldsymbol{b}_n^\top \Delta\boldsymbol{\alpha} = 0. \tag{52}$$

From these equations, there hold

$$\Delta\boldsymbol{\alpha} = -H_n^{-1}\left\{\boldsymbol{q}_n + \frac{g_n - \boldsymbol{b}_n^\top H_n^{-1}\boldsymbol{q}_n}{\boldsymbol{b}_n^\top H_n^{-1}\boldsymbol{b}_n}\boldsymbol{b}_n\right\} \tag{53}$$

and the parameter $\boldsymbol{\alpha}$ is updated by

$$\boldsymbol{\alpha}_{n+1} = \boldsymbol{\alpha}_n - H_n^{-1}\left\{\boldsymbol{q}_n + \frac{g_n - \boldsymbol{b}_n^\top H_n^{-1}\boldsymbol{q}_n}{\boldsymbol{b}_n^\top H_n^{-1}\boldsymbol{b}_n}\boldsymbol{b}_n\right\}. \tag{54}$$

We minimize eq.(15) by Newton's method with constraint $g = \boldsymbol{\lambda}^\top\boldsymbol{\lambda} - 1$ in the parameter space $\widetilde{\boldsymbol{\lambda}} = \begin{pmatrix}\boldsymbol{\lambda}\\ \mu\end{pmatrix}$.

The updates of parameters are computed by the following values: (Gray et al. [11] gave the case $n = 2$ and $d = 1$)

$$\frac{\partial J}{\partial \boldsymbol{\lambda}} = -\sum_{f=1}^{F}\frac{r_f}{\sin\psi_f}\boldsymbol{z}_f, \quad \frac{\partial J}{\partial \mu} = -\sum_{f=1}^{F} r_f,$$

$$\frac{\partial^2 J}{\partial \boldsymbol{\lambda}\partial \boldsymbol{\lambda}^\top} = \sum_{f=1}^{F}\frac{\sin\psi_f - r_f\cos\psi_f}{\sin^3\psi_f}\boldsymbol{z}_f\boldsymbol{z}_f^\top,$$

$$\frac{\partial^2 J}{\partial \boldsymbol{\lambda}\partial \mu} = \left(\frac{\partial^2 J}{\partial \mu\partial \boldsymbol{\lambda}^\top}\right)^\top = \sum_{f=1}^{F}\frac{1}{\sin\psi_f}\boldsymbol{z}_f,$$

$$\frac{\partial^2 J}{\partial \mu\partial \mu} = \sum_{f=1}^{F} 1 = F, \quad \frac{\partial g}{\partial \widetilde{\boldsymbol{\lambda}}} = 2\begin{pmatrix}\boldsymbol{\lambda}\\ 0\end{pmatrix}. \tag{55}$$

8.3 Sequential Dimension Reduction (SDR)

We consider the sequence of small hyperspheres as

$$S^n \supset \alpha_1 S^{n-1} \supset \alpha_2 S^{n-2} \supset \cdots \supset \alpha_{n-d} S^d, \tag{56}$$

and we estimate $\alpha_1 S^{n-1}$, $\alpha_2 S^{n-2}$, ..., $\alpha_{n-d} S^d$ sequentially.

In each estimation, we use Newton's method referred in previous subsection. We compute the initial value of Newton's method by SLS-ESG [7]. We call this estimation as the **Sequential Dimension Reduction** (**SDR**).

To avoid the cumulative errors by a sequential procedure, z_f are directly projected onto $\alpha_k S^{n-k}$ to estimate $\alpha_{k+1} S^{n-k-1}$.

9. Experiment for small hypersphere fitting

We generated 100 data from 5-dimensional hypersphere on the 20-dimensional unit hypersphere S^{20}. The data are sampled from uniform distribution on the 5-dimensional hypersphere. We add the 0-mean Gauss noise of its variance 0.01 for each axis and normalize data by divided by its norm. The hyperspherical data $\boldsymbol{z}_f$ which we use in experiments are the rotation of the normalized data by some 20-dimensional rotation matrix.

Figure24 shows the dimension of estimated small hypersphere against the mean square error along geodesic. The solid line denotes the mean square error for SDR, and dotted line denotes the mean square error for SLS-ESG.

From 20-dimension to 5-dimension which is the true dimension, both methods give good performance.

However, below the true dimension, that is over compression of the data, the error raised rapidly. In this case, SLS-ESG gives a better estimation than SDR. We can compute

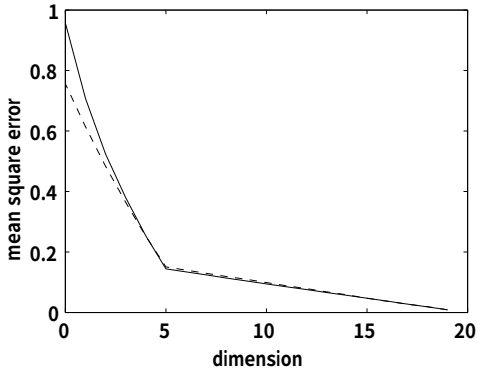


Fig. 24 Training error against dimension reduction: solid line is for SDR and dotted line is for SLS-ESG

the mean square error only by the observed data, we can estimate the "true" dimension of data only by the observed data.

We test the estimation by 10000-test data. Figure 25 shows the square error per test datum for 100 trials. Both methods give good performance for test data.

10. Future

From the proposed methods, we can suggest following two

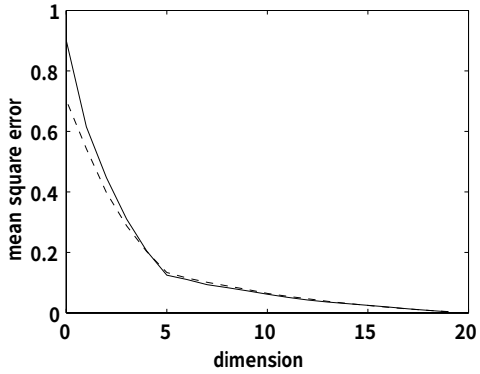


Fig. 25 Test error against dimension reduction: solid line is for SDR and dotted line is for SLS-ESG

points:

(1) Least squares along geodesic for n-dimensional metric manifold is resolved to the weighted least squares in Euclidean space when we can define the continuous and differentiable bijection from the manifold to Euclidean space. That is, Euclideanization is applicable.

(2) The clustering on n-dimensional hypersphere is resolved to the clustering on n-dimensional Euclidean space by Euclideanization.

(3) In this paper, Euclideanization is the putting weights for data points. However, we should consider the weights not only for data points but also the geodesic from the data points to the low dimensional structure for fitting. From this point of view, the proposed Euclideanization should be called **0-th order Euclideanization**, and we should investigate **1-st order Euclideanization**.

Appendix A: Image of hypersphere by stereographic projection

In $n+1$-dimensional Euclidean space $\boldsymbol{R}^{n+1}$, the image of the d-dimensional hypersphere on the n-dimensional hypersphere is also the d-dimensional hypersphere on the n-dimensional hyperplane Π, which is the tangent hyperplane at the antipode of N by the stereographic projection of the n-dimensional hypersphere S^n from the point N.

In this appendix, d-dimensional Euclidean space is regarded as the d-dimensional hypersphere with radius infinity.

Let a diameter of the n-dimensional hypersphere as $2r$. When we set the origin of the $n+1$ Euclidean space as N on the n-dimensional hypersphere, the stereographic projection from N, named f, is represented as

$$f(\boldsymbol{x}) = \frac{4r^2}{||\boldsymbol{x}||^2}\,\boldsymbol{x}.$$

This is the inversion with respect to inversion hypersphere with center N and radius $2r$. By this inversion, any n-dimensional hypersphere is mapped to n-dimensional hy-

persphere because the image of n-dimensional hypersphere $a||\boldsymbol{x}||^2 + \boldsymbol{b}^\top \boldsymbol{x} + c = 0$ is

$$c||\boldsymbol{x}||^2 + r^2 \boldsymbol{b}^\top \boldsymbol{x} + r^4 a = 0,$$

which is also n-dimensional hypersphere.

Generally, a d-dimensional hypersphere is represented as the intersection of $n-d+1$ pieces of n-dimensional hypersphere, then the image of the d-dimensional hypersphere is also an intersection of $n-d+1$ pieces of n-dimensional hypersphere, that is a d-dimensional hypersphere.

Especially, d-dimensional hypersphere on the n-dimensional hypersphere S^n, is a d-dimensional hypersphere on the n-dimensional hyperplane Π, which is the image of S^n.

Appendix B: Representation of equisolidangle projection

Because equisolidangle projection does not change the volume element, there holds $J = 1$, that is,

$$\{f(x)\}^{n-1} f'(x)(1-x^2) - x\{f(x)\}^n = 1. \tag{57}$$

Because there holds

$$\begin{aligned} &\frac{d}{dx}(1-x^2)^{\frac{n}{2}}\{f(x)\}^n \\ &= n(1-x^2)^{\frac{n-2}{2}}\left[(1-x^2)\{f(x)\}^{n-1} f'(x) - x\{f(x)\}^n\right], \end{aligned}$$

the differential Eq.57 is rewritten as

$$\frac{d}{dx}(1-x^2)^{\frac{n}{2}}\{f(x)\}^n = n(1-x^2)^{\frac{n-2}{2}}. \tag{58}$$

Let $x = \cos\theta$ and $f(x) = g(\theta)$, there holds

$$dx = -\sin\theta d\theta \tag{59}$$

and the differential equation of x is rewritten as the differential equation of θ as

$$\begin{aligned} \frac{d}{d\theta}\sin^n\theta\{g(\theta)\}^n &= \left(\frac{d}{dx}(1-x^2)^{\frac{n}{2}}\{f(x)\}^n\right)\frac{dx}{d\theta} \\ &= n(1-x^2)^{\frac{n-2}{2}} \cdot (-\sin\theta) \\ &= -n\sin^{n-1}\theta. \end{aligned} \tag{60}$$

By integrating from π to θ, there holds

$$\sin^n\theta\{g(\theta)\}^n = -n\int_\pi^\theta \sin^{n-1}\phi\, d\phi \tag{61}$$

because there holds

$$\sin^n\theta\{g(\theta)\}^n\Big|_{\theta=\pi} = 0. \tag{62}$$

Let

$$J_n(\theta) = -\int_\pi^\theta \sin^n\phi\, d\phi, \tag{63}$$

there holds

$$\begin{aligned} J_n(\theta) &= \int_\pi^\theta \sin^{n-1}\phi \frac{d\cos\phi}{d\phi}\, d\phi \\ &= \left[\sin^{n-1}\phi\cos\phi\right]_\pi^\theta + (n-1)(J_{n-2}(\theta) - J_n(\theta)) \\ &= \sin^{n-1}\theta\cos\theta + (n-1)(J_{n-2}(\theta) - J_n(\theta)), \end{aligned}$$

that is,

$$J_n(\theta) = \frac{n-1}{n}J_{n-2}(\theta) + \frac{1}{n}\sin^{n-1}\theta\cos\theta \tag{64}$$

for $n \geqq 2$.

For $n = 0, 1$, the J_n are computed as

$$J_0(\theta) = -\int_\pi^\theta d\phi = \pi - \theta, \tag{65}$$

$$J_1(\theta) = -\int_\pi^\theta \sin\phi\, d\phi = \cos\theta + 1, \tag{66}$$

$J_n(\theta)$ are computed recursively by using recurrent Eq.64.

By using $J_n(\theta)$, there holds

$$\sin^n\theta\{g(\theta)\}^n = nJ_{n-1}(\theta), \tag{67}$$

and the solution of the differential Eq.60 is

$$g(\theta) = \frac{\{nJ_{n-1}(\theta)\}^{\frac{1}{n}}}{\sin\theta}. \tag{68}$$

Then the solution of the differential Eq.57 is

$$f(x) = \frac{\{nI_{n-1}(x)\}^{\frac{1}{n}}}{\sqrt{1-x^2}} \tag{69}$$

where $J_n(\theta) = I_n(x)$.

The recurrent equation of $I_n(x)$ is

$$I_0(x) = \pi - \cos^{-1}x, \tag{70}$$

$$I_1(x) = 1 + x, \tag{71}$$

$$I_n(x) = \frac{n-1}{n}I_{n-2} + \frac{1}{n}x(1-x^2)^{\frac{n-1}{2}} \tag{72}$$

and the representation of $I_n(x)$ for $n = 2, 3$ are

$$I_2(x) = \frac{\pi - \cos^{-1}x + x(1-x^2)^{\frac{1}{2}}}{2}, \tag{73}$$

$$I_3(x) = \frac{2 + 3x - x^3}{3}. \tag{74}$$

For $n = 1$, the solution of the differential Eq.57 is

$$f(x) = \frac{\pi - \cos^{-1}x}{\sqrt{1-x^2}}, \tag{75}$$

which is azimuthal equidistant projection. Then the relation between r and x_{n+1} is

$$r = \pi - \cos^{-1}x_{n+1} \iff x_{n+1} = -\cos r. \tag{76}$$

For $n = 2$, the solution of the differential Eq.57 is

$$f(x) = \sqrt{\frac{2}{1-x}} \tag{77}$$

which is well known result of equisolid projection of S^2. The relation between r and x_{n+1} is

$$r = \sqrt{2(1+x)} \iff x_{n+1} = \frac{r^2}{2} - 1 \tag{78}$$

For $n = 3$, the solution of the differential Eq.57 is

$$f(x) = \sqrt[3]{\frac{3}{2}} \frac{\{\pi - \cos^{-1} x + x(1-x^2)^{\frac{1}{2}}\}^{\frac{1}{3}}}{\sqrt{1-x^2}}. \qquad (79)$$

In this case, x_{n+1} is difficult to represent by r, then the inverse mapping from projected space to hypersphere S^3 is difficult to compute.

References

[1] S. Akaho, "Curve fitting that minimizes the mean square of perpendicular distances from sample points," In Proc. of SPIE93, Vision Geometry II, Vol.2060, (1993).

[2] S. Baker and S. K. Nayar, "A theory of catadioptric image formation," In Proc. of IEEE International Conference on Computer Vision (ICCV98), pp.35–42, Jan. 1998.

[3] A. Brace, D. Gątarek, M. Musiela, "The market model of interest rate dynamics," Mathematical Finance, Vol.7, No.2, pp.127-155, (1997).

[4] K. Daniilidis, A. Makadia and T. Bulow, "Image processing in catadioptric planes: spatiotemporal derivatives and optical flow computation," *OMNIVIS02*, pp.3-10, 2002.

[5] N. I. Fisher, T. Lewis and B. J. J. Embleton, "Statistical anaylsis of spherical data," Cambridge Univ. Press, 1987.

[6] J. Fujiki, S. Akaho and N. Murata, "Nonlinear PCA/ICA for the structure from motion problem," In Proc. of ICA04(LNCS 3195), pp. 750-757, Granada, Sep. 2004.

[7] J. Fujiki and S. Akaho, "Small ciecle fitting to the sequence of spherical points -Towards smoothing of camera motions-," Technical Report of IEICE, PRMU2004-149, pp.91-96, 2004 , (In Japanese).

[8] J. Fujiki and S. Akaho, "Epipolar geometry for spherical camera and its calculations," Technical Report of IEICE, PRMU2005-41, pp.41-46, 2005 , (In Japanese).

[9] J. Fujiki and S. Akaho, "Small hypersphere fitting by Spherical Least Square," In Proc. of ICONIP05, pp.439-444, 2005.

[10] C. Geyer and K. Daniilidis, "Catadioptric Projective Geometry," IJCV, vol.45, no.3, pp.223-243, Dec. 2001.

[11] N. H. Gray, P. A. Geiser and J. R. Geiser, "On the least-squares fit of small and great circles to spherically projected orientation data," Mathematical Geology, vol.12, no.3, pp.173–184, 1980.

[12] I. Grubišić and R. Pietersz, "Efficient rank reduction of correlation matrices," Utrecht Univ., preprint, 2005.

[13] A. Makadia and K. Daniilidis, "Direct 3D-rotation estimation from spherical images via a generalized shift theorem," In proc. of CVPR03, pp.II: 217-224, 2003.

[14] J. Mimura, N. Murata and J. Fujiki, "Smoothing of camera motions via small circle fitting for the sequence of spherical points," Technical Report of IEICE, PRMU, 2005, (In Japanese).

[15] K. Miyamoto, "Fish eye lens," Journal of Optical Society of America, vol.54, no.8, pp.1060-1061, Aug. 1964.

[16] S. Oba and S. Ishii, "Kernel density estimation on hypersphere," 2004 Workshop on Information-Based Induction Sciences(IBIS2004), pp.197-202, (2004), (In Japanese).

[17] T. Svoboda and T. Pajdla, "Epipolar Geometry for Central Catadioptric Cameras," IJCV, vol.49, no.1, pp.23-37, 2002.

[18] A. Torii and A. Imiya, "Analysis of Central Camera Systems for Computer Vision," CVIM 154-30, 2006.

Local Fisher Discriminant Analysis

Masashi SUGIYAMA†

† Department of Computer Science, Tokyo Institute of Technology
2-12-1-W8-74, O-okayama, Meguro-ku, Tokyo, 152-8552, Japan.

Abstract Dimensionality reduction is one of the important preprocessing steps in high-dimensional data analysis. In this paper, we consider the supervised dimensionality reduction problem where samples are accompanied with class labels. Fisher discriminant analysis (FDA) for linear dimensionality reduction is a traditional but yet powerful technique for this purpose. However, it tends to give undesired results if samples in some class are *multimodal*. Multimodal data samples may be embedded appropriately by the method of locality-preserving projection (LPP), which aims at preserving local structure of the data. However, it is an unsupervised method so is not necessarily effective for supervised dimensionality reduction. In this paper, we propose a new method called *local Fisher discriminant analysis* (LFDA), which effectively combines the ideas of FDA and LPP so is useful for embedding multimodal labeled samples. Recently, novel dimensionality reduction methods such as neighborhood component analysis (NCA) and maximally collapsing metric learning (MCML) have been proposed, which are shown to work better than FDA. An advantage of LFDA over these state-of-the-art methods is that LFDA has an *analytic* form of the embedding transformation matrix which can be easily computed by solving a generalized eigenvalue problem, while NCA and MCML rely on gradient-based iterative procedures which are computationally inefficient. Numerical examples show that LFDA compares favorably to existing methods in both embedding quality and scalability. We also show that LFDA can be extended to non-linear dimensionality reduction scenarios by the kernel trick.

Key words Dimensionality Reduction, Fisher Discriminant Analysis, Locality Preserving Projection, Multimodality, Kernel Trick

1. Introduction

The goal of dimensionality reduction is to embed high-dimensional data samples in a low-dimensional space while most of 'intrinsic information' contained in the data is preserved (e.g., Roweis and Saul, 2000; Tenenbaum et al., 2000). Once dimensionality reduction is carried out appropriately, one can utilize the compact representation of the data for various succeeding tasks such as visualization, classification, etc. In this paper, we consider the *supervised* dimensionality reduction problem where samples are accompanied with class labels.

Fisher discriminant analysis (FDA) (Fisher, 1936; Fukunaga, 1990) is a popular method for linear supervised dimensionality reduction(*1). FDA embeds the data samples so that between-class scatter is maximized and within-class scatter is minimized. FDA is a traditional but yet useful method for dimensionality reduction. However, it tends to give undesired results if samples in some class form several separate clusters (i.e., *multimodal*) (see e.g., Fukunaga, 1990).

Within-class multimodality can be observed in many practical applications. For example, in disease diagnosis, the distribution of medial checkup samples of sick patients could be multimodal since there may be several different causes even for a single disease. Even in a traditional task of handwritten digit recognition, within-class multimodality appears if digits are classified into, e.g., even and odd numbers. More generally, within-class multimodality can naturally appear if multi-class classification problems are solved by a set of two-class 'one-versus-rest' problems. For this reason, there is a universal need for reducing the dimensionality of multimodal

(*1) : FDA may refer to a classification method which first projects the data samples onto a one-dimensional space and then classifies the samples by thresholding (Fisher, 1936; Duda et al., 2001). The one-dimensional embedding space used here is obtained as the maximizer of the so-called *Fisher criterion*. This Fisher criterion can be used for dimensionality reduction onto a subspace with dimension more than one (Fukunaga, 1990). With some abuse, we refer to the dimensionality reduction method based on the Fisher criterion as FDA (see Section 2. 2 for detail).

data.

In order to embed multimodal data appropriately, it is important to preserve 'local structure' of the data. *Locality-preserving projection* (LPP) (He and Niyogi, 2004) meets this requirement. LPP keeps nearby data pairs in the original space close in the embedding space, by which multimodal data can be embedded without losing its local structure. However, LPP is an unsupervised dimensionality reduction method which does not take the label information into account. Therefore, it does not necessarily work appropriately in supervised dimensionality reduction scenarios.

In this paper, we propose a new dimensionality reduction method called *local Fisher discriminant analysis* (LFDA). LFDA effectively combines the ideas of FDA and LPP: between-class separability is maximized while *within-class local structure* is preserved.

Thanks to the local structure preservation property, multimodal labeled data can be embedded appropriately by LFDA. Furthermore, LFDA can give more separate embedding than FDA. To explain the reason intuitively, we first note that FDA can be regarded as maximizing between-class scatter under the constraint of keeping within-class scatter to a certain level (Fukunaga, 1990, see also Section 3. 3). When samples in some class are multimodal, keeping within-class scatter to a certain level appears to be quite hard since multimodal samples should be typically merged into a single cluster. Due to this strong constraint, less degree of freedom is left for increasing separability, and thus FDA results in less separate embedding. On the other hand, LFDA does not require multimodal samples to collapse into a single cluster. As a result, more degree of freedom is left for increasing separability and thus highly separate embedding can be obtained.

If the locality is chosen to be global in LFDA, it is reduced to the original FDA. Therefore, LFDA may be regarded as a natural extension of FDA to incorporate local structure of the data.

By the so-called *kernel trick* (Schölkopf and Smola, 2002), FDA and LPP can be extended to non-linear dimensionality reduction scenarios (Baudat and Anouar, 2000; Mika et al., 2003; Belkin and Niyogi, 2003; He and Niyogi, 2004). We show that LFDA can also be non-linearized by the kernel trick.

Although non-linear dimensionality reduction is in principle more flexible than linear dimensionality reduction, a lot of attention are still paid to linear dimensionality reduction since linear methods often have more *exploratory* power than non-linear methods. Recently, novel methods have been proposed for linear supervised dimensionality reduction.

The method called *neighborhood component analysis* (NCA) (Goldberger et al., 2005) embeds the data samples so that the number of correctly classified samples by a stochastic variant of nearest neighbor classifiers is maximized. By definition, NCA well separates samples of different classes from each other. Moreover, our experiments (see Section 4.) show that NCA is useful even when multimodal data samples are embedded, although NCA does not explicitly take the multimodality of the data into account. However, the optimization problem which needs to be solved in NCA is non-convex so there is no guarantee that one can obtain the globally optimal solution. Furthermore, a gradient-based iterative procedure is employed for optimization, which is rather demanding in computation.

The method called *maximally collapsing metric learning* (MCML) (Globerson and Roweis, 2006) tries to make samples in the same class collapse into a single point and samples in other classes map to other locations. Since this requirement may not be rigorously achieved by linear dimensionality reduction, MCML instead minimizes the approximation error to the optimal collapse. By definition, MCML does not take the multimodality of the data into account since samples in the same class are required to collapse into a *single* point. However, MCML appears to be still useful even when it is used for embedding multimodal data (see experiments in Section 4.). A drawback of MCML is again the non-convexity of the optimization problem: the MCML algorithm only gives an approximate solution in general(*2). Furthermore, the algorithm employs an iterative updating scheme for optimization, which appears to be computationally expensive.

On the other hand, LFDA has an *analytic* form of the embedding transformation matrix which can be easily computed by solving a generalized eigenvalue problem. Therefore, LFDA is more efficient in computation and more reliable than NCA and MCML.

The rest of this paper is organized as follows. In Section 2., we formulate the problem of linear dimensionality reduction, review the definitions of FDA and LPP, and illustrate how they typically behave. In Section 3., we derive LFDA, show its properties, and discuss its relation to existing methods. In Section 4., numerical examples are shown where LFDA is experimentally compared with existing methods. Finally, in Section 5., concluding remarks and future prospects are given.

(*2) : Note that, if dimensionality of the embedding space is the same as the entire space (i.e., the dimensionality is not reduced but the only the *metric* of the original space is changed), the optimization problem of MCML is convex, thus the globally optimal solution can be obtained.

2. Linear Dimensionality Reduction

In this section, we formulate the problem of linear dimensionality reduction and review typical existing methods.

2.1 Formulation

Let $\boldsymbol{x}_i \in \mathbb{R}^D$ $(i = 1, 2, \ldots, n)$ be D-dimensional samples and $y_i \in \{1, 2, \ldots, C\}$ be associated class labels, where n is the number of samples and C is the number of classes. Let n_c be the number of samples in the class c:

$$\sum_{c=1}^{C} n_c = n. \tag{1}$$

In the following, we use i, j, and k for the index of sample number, c for the index of class, and d for the index of dimension.

Let $\boldsymbol{X}$ be the matrix of all samples:

$$\boldsymbol{X} = (\boldsymbol{x}_1 | \boldsymbol{x}_2 | \cdots | \boldsymbol{x}_n). \tag{2}$$

Let $\boldsymbol{z}_i \in \mathbb{R}^R$ $(1 \leqq R \leqq D)$ be embedded samples, where R is the reduced dimension (i,.e., the dimension of the embedding space). Effectively we consider D to be large and R to be small, but not limited to such cases.

For the moment, we focus on linear dimensionality reduction, i.e., using a $D \times R$ transformation matrix $\boldsymbol{T}$, $\boldsymbol{z}_i$ is given by

$$\boldsymbol{z}_i = \boldsymbol{T}^\top \boldsymbol{x}_i. \tag{3}$$

Later in Section 3.6, we discuss non-linear dimensionality reduction scenarios.

2.2 Fisher Discriminant Analysis

Here we briefly review the definition of *Fisher discriminant analysis* (FDA) (Fisher, 1936; Fukunaga, 1990; Duda et al., 2001) for dimensionality reduction.

Let $\boldsymbol{S}^{(w)}$ and $\boldsymbol{S}^{(b)}$ be the *within-class scatter matrix* and the *between-class scatter matrix* defined by

$$\boldsymbol{S}^{(w)} = \sum_{c=1}^{C} \sum_{i:y_i=c} (\boldsymbol{x}_i - \boldsymbol{\mu}_c)(\boldsymbol{x}_i - \boldsymbol{\mu}_c)^\top, \tag{4}$$

$$\boldsymbol{S}^{(b)} = \sum_{c=1}^{C} n_c (\boldsymbol{\mu}_c - \boldsymbol{\mu})(\boldsymbol{\mu}_c - \boldsymbol{\mu})^\top, \tag{5}$$

where $\sum_{i:y_i=c}$ denotes the summation over i such that $y_i = c$, $^\top$ denotes the transpose, $\boldsymbol{\mu}_c$ is the mean of samples in the class c, and $\boldsymbol{\mu}$ is the mean of all samples:

$$\boldsymbol{\mu}_c = \frac{1}{n_c} \sum_{i:y_i=c} \boldsymbol{x}_i, \tag{6}$$

$$\boldsymbol{\mu} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}_i = \frac{1}{n} \sum_{c=1}^{C} n_c \boldsymbol{\mu}_c. \tag{7}$$

Using $\boldsymbol{S}^{(w)}$ and $\boldsymbol{S}^{(b)}$, the FDA transformation matrix $\boldsymbol{T}_{FDA}$ is defined as follows:

$$\boldsymbol{T}_{FDA} = \underset{\boldsymbol{T} \in \mathbb{R}^{D \times R}}{\operatorname{argmax}} \left[\operatorname{tr} \left((\boldsymbol{T}^\top \boldsymbol{S}^{(w)} \boldsymbol{T})^{-1} \boldsymbol{T}^\top \boldsymbol{S}^{(b)} \boldsymbol{T} \right) \right]. \tag{8}$$

That is, FDA determines $\boldsymbol{T}$ so that between-class scatter is maximized while within-class scatter is minimized.

It is known that $\boldsymbol{T}_{FDA}$ is given by

$$\boldsymbol{T}_{FDA} = (\boldsymbol{\varphi}_1 | \boldsymbol{\varphi}_2 | \cdots | \boldsymbol{\varphi}_R), \tag{9}$$

where $\{\boldsymbol{\varphi}_d\}_{d=1}^{D}$ are the generalized eigenvectors associated to the generalized eigenvalues $_1 \geqq {}_2 \geqq \cdots \geqq {}_D$ of the following generalized eigenvalue problem:

$$\boldsymbol{S}^{(b)} \boldsymbol{\varphi} = \boldsymbol{S}^{(w)} \boldsymbol{\varphi}. \tag{10}$$

2.3 Locality-Preserving Projection

Here we briefly review the definition of *locality-preserving projection* (LPP) (He and Niyogi, 2004). Let $\boldsymbol{A}$ be the *affinity matrix*, i.e., the n-dimensional matrix with the (i, j)-th element being the affinity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. We suppose that elements of $\boldsymbol{A}$ are in $[0, 1]$ and take larger values if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are 'close'. There are several different manners in defining $\boldsymbol{A}$. We give a brief review of typical definitions in Appendix 1..

Using the affinity matrix $\boldsymbol{A}$, the LPP transformation matrix $\boldsymbol{T}_{LPP}$ is defined as follows.

$$\boldsymbol{T}_{LPP} = \underset{\boldsymbol{T} \in \mathbb{R}^{D \times R}}{\operatorname{argmin}} \left(\frac{1}{2} \sum_{i\ j=1}^{n} \boldsymbol{A}_{i\ j} \|\boldsymbol{T}^\top \boldsymbol{x}_i - \boldsymbol{T}^\top \boldsymbol{x}_j\|^2 \right)$$
$$\text{subject to } \boldsymbol{T}^\top \boldsymbol{X} \boldsymbol{D} \boldsymbol{X}^\top \boldsymbol{T} = \boldsymbol{I}, \tag{11}$$

where $\boldsymbol{I}$ is the identity matrix and $\boldsymbol{D}$ is the n-dimensional diagonal matrix with i-th diagonal element being

$$\boldsymbol{D}_{i\ i} = \sum_{j=1}^{n} \boldsymbol{A}_{i\ j}. \tag{12}$$

Eq.(11) implies that LPP determines $\boldsymbol{T}$ so that *nearby* data pairs in the original space $\mathbb{R}^D$ are kept close in the embedding space. Note that $\boldsymbol{T}^\top \boldsymbol{X} \boldsymbol{D} \boldsymbol{X}^\top \boldsymbol{T} = \boldsymbol{I}$ is a constraint to avoid a trivial solution $\boldsymbol{T} = \boldsymbol{O}$.

It is known that the LPP transformation matrix $\boldsymbol{T}_{LPP}$ is given by

$$\boldsymbol{T}_{LPP} = (\boldsymbol{\psi}_{D-R+1} | \boldsymbol{\psi}_{D-R+2} | \cdots | \boldsymbol{\psi}_D), \tag{13}$$

where $\{\boldsymbol{\psi}_d\}_{d=1}^{D}$ are the eigenvectors associated to the eigenvalues $\gamma_1 \geqq \gamma_2 \geqq \cdots \geqq \gamma_D$ of the following eigenvalue problem:

$$\boldsymbol{X} \boldsymbol{L} \boldsymbol{X}^\top \boldsymbol{\psi} = \gamma \boldsymbol{X} \boldsymbol{D} \boldsymbol{X}^\top \boldsymbol{\psi}, \tag{14}$$

where

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}. \tag{15}$$

Note that $\boldsymbol{L}$ is called the *graph-Laplacian* matrix in the spectral graph theory (Chung, 1997), where $\boldsymbol{A}$ is seen as the *adjacency matrix* of a graph.

2.4 Typical Behavior

In Figure 1, dimensionality reduction results obtained by FDA and LPP are illustrated, where two-dimensional two-class data samples are embedded in a one-dimensional subspace(*3). In LPP, the affinity matrix $\boldsymbol{A}$ is determined by the *local scaling method* (Zelnik-Manor and Perona, 2005, see also Appendix 1.).

For the simplest data set depicted in Figure 1(a), both FDA and LPP give reasonable results where samples of different classes are nicely separated. For the data set depicted in Figure 1(b), FDA still works well. However, LPP mixes samples of different classes into one cluster, which is caused by the unsupervised nature of LPP. On the other hand, for the data set depicted in Figure 1(c), LPP works well but FDA collapses samples of different classes into a single cluster. The reason is that the levels of between-class scatter and within-class scatter are not evaluated in an intuitively natural way because of the two separate clusters of the '×'-class (see also Fukunaga, 1990).

3. Local Fisher Embedding

As illustrated above, FDA can perform poorly if samples in some class form several separate clusters (i.e., *multimodal*). In other words, the undesired behavior is caused by the *globality* when evaluating within-class scatter and between-class scatter. On the other hand, LPP can make samples of different classes overlapped if they are close in the original high-dimensional space $\mathbb{R}^D$.

To overcome this problem, we propose combining the idea of FDA and LPP. Namely, we evaluate the levels of within-class scatter and between-class scatter in a *local* manner, by which class separability and local structure preservation could be attained at the same time. We call our new method *local Fisher discriminant analysis* (LFDA).

3.1 Reformulating FDA

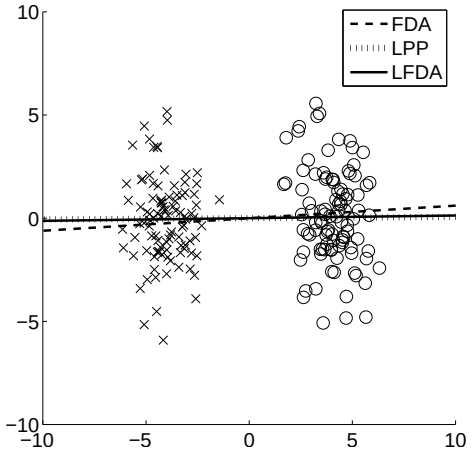
To introduce the new method, let us first reformulate FDA in a *pairwise* manner.

[Lemma 1] $\boldsymbol{S}^{(w)}$ and $\boldsymbol{S}^{(b)}$ defined by Eqs.(4) and (5) are expressed as

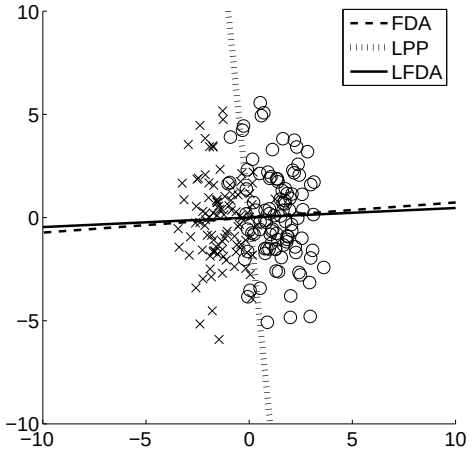
$$\boldsymbol{S}^{(w)} = \frac{1}{2}\sum_{i,j=1}^{n} \boldsymbol{A}^{(w)}_{i,j}(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top, \qquad (16)$$

$$\boldsymbol{S}^{(b)} = \frac{1}{2}\sum_{i,j=1}^{n} \boldsymbol{A}^{(b)}_{i,j}(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top, \qquad (17)$$

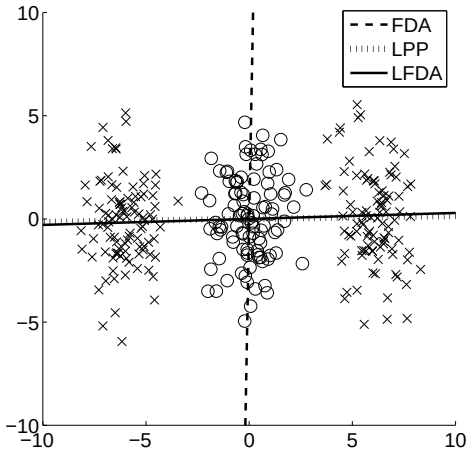
(*3) : In Figure 1, the *range* of the transformation matrix determined by each method is depicted. Since FDA and LPP do not necessarily give an *orthogonal projection matrix* (i.e., $\boldsymbol{T}^2 = \boldsymbol{T}$ and $\boldsymbol{T}^\top = \boldsymbol{T}$ are not necessarily satisfied), the range of the transformation matrix may not display the complete information of embedding results. However, it is sufficient for the discussion here.



(a) Toy data set 1



(b) Toy data set 2



(c) Toy data set 3

Fig. 1 Examples of dimensionality reduction by FDA, LPP and LFDA. Two-dimensional two-class samples are embedded in a one-dimensional space. The line in the figure denotes the embedding space obtained by each method.

where

$$\boldsymbol{A}^{(w)}_{i\,j} = \begin{cases} 1/n_c & \text{if } y_i = y_j = c, \\ 0 & \text{if } y_i \neq y_j, \end{cases} \tag{18}$$

$$\boldsymbol{A}^{(b)}_{i\,j} = \begin{cases} 1/n - 1/n_c & \text{if } y_i = y_j = c, \\ 1/n & \text{if } y_i \neq y_j. \end{cases} \tag{19}$$

(Proof) It follows from Eq.(4) that

$$\begin{aligned} \boldsymbol{S}^{(w)} &= \sum_{c=1}^{C} \sum_{i:y_i=c} \left(\boldsymbol{x}_i - \frac{1}{n_c} \sum_{j:y_j=c} \boldsymbol{x}_j \right) \left(\boldsymbol{x}_i - \frac{1}{n_c} \sum_{k:y_k=c} \boldsymbol{x}_k \right)^\top \\ &= \sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^\top - \sum_{c=1}^{C} \frac{1}{n_c} \sum_{i\,j:y_i=y_j=c} \boldsymbol{x}_i \boldsymbol{x}_j^\top \\ &= \sum_{i=1}^{n} \left(\sum_{j=1}^{n} \boldsymbol{A}^{(w)}_{i\,j} \right) \boldsymbol{x}_i \boldsymbol{x}_i^\top - \sum_{i\,j=1}^{n} \boldsymbol{A}^{(w)}_{i\,j} \boldsymbol{x}_i \boldsymbol{x}_j^\top \\ &= \frac{1}{2} \sum_{i\,j=1}^{n} \boldsymbol{A}^{(w)}_{i\,j} (\boldsymbol{x}_i \boldsymbol{x}_i^\top + \boldsymbol{x}_j \boldsymbol{x}_j^\top - \boldsymbol{x}_i \boldsymbol{x}_j^\top - \boldsymbol{x}_j \boldsymbol{x}_i^\top), \end{aligned} \tag{20}$$

which yields Eq.(16). Let $\boldsymbol{S}^{(m)}$ be the *mixture scatter matrix* (Fukunaga, 1990):

$$\begin{aligned} \boldsymbol{S}^{(m)} &= \boldsymbol{S}^{(w)} + \boldsymbol{S}^{(b)} \\ &= \sum_{i=1}^{n} (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^\top. \end{aligned} \tag{21}$$

Then we have

$$\begin{aligned} \boldsymbol{S}^{(b)} &= \sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^\top - \frac{1}{n} \sum_{i\,j=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_j^\top - \boldsymbol{S}^{(w)} \\ &= \sum_{i=1}^{n} \left(\sum_{j=1}^{n} \frac{1}{n} \right) \boldsymbol{x}_i \boldsymbol{x}_i^\top - \sum_{i\,j=1}^{n} \frac{1}{n} \boldsymbol{x}_i \boldsymbol{x}_j^\top - \boldsymbol{S}^{(w)} \\ &= \frac{1}{2} \sum_{i\,j=1}^{n} \left(\frac{1}{n} - \boldsymbol{A}^{(w)}_{i\,j} \right) (\boldsymbol{x}_i \boldsymbol{x}_i^\top + \boldsymbol{x}_j \boldsymbol{x}_j^\top - \boldsymbol{x}_i \boldsymbol{x}_j^\top - \boldsymbol{x}_j \boldsymbol{x}_i^\top), \end{aligned} \tag{22}$$

which yields Eq.(17). ■

Note that $1/n - 1/n_c$ in Eq.(19) is negative while $1/n_c$ and $1/n$ in Eqs.(18) and (19) are positive.

The above pairwise representation gives us a novel interpretation of FDA: it tries to keep in-class data pairs close (since $\boldsymbol{A}^{(w)}_{i\,j}$ is positive and $\boldsymbol{A}^{(b)}_{i\,j}$ is negative if $y_i = y_j$) and between-class data pairs apart (since $\boldsymbol{A}^{(w)}_{i\,j}$ is zero and $\boldsymbol{A}^{(b)}_{i\,j}$ is positive if $y_i \neq y_j$).

3. 2 Definition and Typical Behavior of LFDA

Based on the above pairwise expression, let us define the *local* within-class scatter matrix $\widetilde{\boldsymbol{S}}^{(w)}$ and the *local* between-class scatter matrix $\widetilde{\boldsymbol{S}}^{(b)}$ as follows.

$$\widetilde{\boldsymbol{S}}^{(w)} = \frac{1}{2} \sum_{i\,j=1}^{n} \widetilde{\boldsymbol{A}}^{(w)}_{i\,j} (\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top, \tag{23}$$

$$\widetilde{\boldsymbol{S}}^{(b)} = \frac{1}{2} \sum_{i\,j=1}^{n} \widetilde{\boldsymbol{A}}^{(b)}_{i\,j} (\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top, \tag{24}$$

where

$$\widetilde{\boldsymbol{A}}^{(w)}_{i\,j} = \begin{cases} \boldsymbol{A}_{i\,j}/n_c & \text{if } y_i = y_j = c, \\ 0 & \text{if } y_i \neq y_j, \end{cases} \tag{25}$$

$$\widetilde{\boldsymbol{A}}^{(b)}_{i\,j} = \begin{cases} \boldsymbol{A}_{i\,j}(1/n - 1/n_c) & \text{if } y_i = y_j = c, \\ 1/n & \text{if } y_i \neq y_j. \end{cases} \tag{26}$$

Compared with the global matrices $\boldsymbol{S}^{(w)}$ and $\boldsymbol{S}^{(b)}$, the values for in-class pairs in $\widetilde{\boldsymbol{S}}^{(w)}$ and $\widetilde{\boldsymbol{S}}^{(b)}$ are weighted according the affinity. This means that *far-apart* in-class pairs have less influence in $\widetilde{\boldsymbol{S}}^{(w)}$ and $\widetilde{\boldsymbol{S}}^{(b)}$. In the sequel, we denote the local counterparts of matrices by symbols with *tilde*.

Using $\widetilde{\boldsymbol{S}}^{(w)}$ and $\widetilde{\boldsymbol{S}}^{(b)}$, we define the LFDA transformation matrix $\boldsymbol{T}_{LFDA}$ as

$$\boldsymbol{T}_{LFDA} = \operatorname*{argmax}_{\boldsymbol{T} \in \mathbb{R}^{D \times R}} \left[\mathrm{tr} \left((\boldsymbol{T}^\top \widetilde{\boldsymbol{S}}^{(w)} \boldsymbol{T})^{-1} \boldsymbol{T}^\top \widetilde{\boldsymbol{S}}^{(b)} \boldsymbol{T} \right) \right]. \tag{27}$$

That is, we determine $\boldsymbol{T}$ so that *nearby* data pairs of the same class are close and data pairs of different classes are apart. Data pairs of the same class but far apart are not imposed to be close.

If the affinity matrix $\boldsymbol{A}$ is taken to be one for all in-class pairs (i.e., all in-class pairs are 'equally close' each other), $\widetilde{\boldsymbol{S}}^{(w)}$ and $\widetilde{\boldsymbol{S}}^{(b)}$ are reduced to $\boldsymbol{S}^{(w)}$ and $\boldsymbol{S}^{(b)}$ so LFDA agrees with the original FDA. Therefore, LFDA may be regarded as a natural localized variant of FDA. Since Eq.(27) is the same form as Eq.(8), we can obtain an *analytic* form of $\boldsymbol{T}_{LFDA}$, which can be computed by solving a generalized eigenvalue problem. In Section 3. 4, we show an efficient method for computing $\boldsymbol{T}_{LFDA}$.

Toy examples of dimensionality reduction by LFDA are illustrated in Figure 1, where the affinity matrix $\boldsymbol{A}$ is again determined by the local scaling method(*4). The figures show that LFDA gives desirable results for all three data sets, i.e., LFDA can compensate the drawbacks of FDA and LPP by effectively combining the idea of FDA and LPP.

3. 3 Why is LFDA good?

LFDA does not impose far-apart data pairs of the same class to be close, by which local structure of the data tends to be preserved. This itself is a useful property, e.g., in data visualization tasks because 'interesting' structure such as within-class multimodality is not lost by dimensionality reduction (this is experimentally demonstrated in Section 4.).

In addition to that, LFDA can even provide more *separate* embedding than the original FDA. To explain the reason, let

(*4) : For LFDA, the nearest neighbor search carried out in the local scaling method is performed in a classwise manner since we do not need the affinity value for between-class pairs (see Eqs.(25) and (26)). This also contributes to reducing the computational cost (see Section 3. 4).

us recall that the FDA transformation matrix $\boldsymbol{T}_{FDA}$ can be expressed as follows (Fukunaga, 1990).

$$\boldsymbol{T}_{FDA} = \underset{\boldsymbol{T}\in\mathbb{R}^{D\times R}}{\operatorname{argmax}} \left[\operatorname{tr} \left(\boldsymbol{T}^\top \boldsymbol{S}^{(b)} \boldsymbol{T} \right) \right] \quad \text{subject to } \boldsymbol{T}^\top \boldsymbol{S}^{(w)} \boldsymbol{T} = \boldsymbol{I}. \tag{28}$$

This representation implies that FDA maximizes between-class scatter under the constraint of keeping within-class scatter to a certain level.

When one of the classes is multimodal, the above constraint is actually quite restrictive since the multimodal data samples should be typically merged into a single cluster. Therefore, the remaining degree of freedom which is used for maximizing between-class scatter may not be that much. As a result, FDA can result in less separate embedding.

On the other hand, it is straightforward to show that $\boldsymbol{T}_{LFDA}$ can also be expressed as

$$\boldsymbol{T}_{LFDA} = \underset{\boldsymbol{T}\in\mathbb{R}^{D\times R}}{\operatorname{argmax}} \left[\operatorname{tr} \left(\boldsymbol{T}^\top \widetilde{\boldsymbol{S}}^{(b)} \boldsymbol{T} \right) \right] \quad \text{subject to } \boldsymbol{T}^\top \widetilde{\boldsymbol{S}}^{(w)} \boldsymbol{T} = \boldsymbol{I}. \tag{29}$$

The constraint in Eq.(29) is less restrictive than that in Eq.(28) since faraway pairs of in-class samples are not imposed to be close. Due to this weaker constraint, the degree of freedom which can be used for maximizing separability between different classes remains more than FDA. For this reason, LFDA can result in more separate embedding than FDA. This qualitative discussion will be experimentally substantiated in Section 4..

3.4 Efficiently Computing LFDA Transformation Matrix

Here, we provide an efficient method to compute the LFDA transformation matrix $\boldsymbol{T}_{LFDA}$.

Let $\widetilde{\boldsymbol{S}}^{(m)}$ be the *local* mixture scatter matrix defined by

$$\begin{aligned} \widetilde{\boldsymbol{S}}^{(m)} &= \widetilde{\boldsymbol{S}}^{(w)} + \widetilde{\boldsymbol{S}}^{(b)} \\ &= \frac{1}{2} \sum_{i\,j=1}^{n} \widetilde{\boldsymbol{A}}^{(m)}_{i\,j} (\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top, \end{aligned} \tag{30}$$

where $\widetilde{\boldsymbol{A}}^{(m)}$ is the n-dimensional matrix with (i,j)-th element being

$$\widetilde{\boldsymbol{A}}^{(m)}_{i\,j} = \widetilde{\boldsymbol{A}}^{(w)}_{i\,j} + \widetilde{\boldsymbol{A}}^{(b)}_{i\,j} = \begin{cases} \boldsymbol{A}_{i\,j}/n & \text{if } y_i = y_j, \\ 1/n & \text{if } y_i \neq y_j. \end{cases} \tag{31}$$

Then the same solution $\boldsymbol{T}_{LFDA}$ can be still obtained if $\widetilde{\boldsymbol{S}}^{(b)}$ in Eq.(27) is replaced by $\widetilde{\boldsymbol{S}}^{(m)}$ because of the following identity (cf. Fukunaga, 1990):

$$\begin{aligned} &\operatorname{tr} \left((\boldsymbol{T}^\top \widetilde{\boldsymbol{S}}^{(w)} \boldsymbol{T})^{-1} \boldsymbol{T}^\top \widetilde{\boldsymbol{S}}^{(m)} \boldsymbol{T} \right) \\ &\quad = \operatorname{tr} \left((\boldsymbol{T}^\top \widetilde{\boldsymbol{S}}^{(w)} \boldsymbol{T})^{-1} \boldsymbol{T}^\top \widetilde{\boldsymbol{S}}^{(b)} \boldsymbol{T} \right) + R. \end{aligned} \tag{32}$$

In the following, we employ the criterion with $\widetilde{\boldsymbol{S}}^{(m)}$ (i.e., the left-hand side of Eq.(32)) since it gives a simpler expression.

The left-hand side of Eq.(32) is the same form as FDA. Therefore, the solution $\boldsymbol{T}_{LFDA}$ is analytically given as follows.

$$\boldsymbol{T}_{LFDA} = (\widetilde{\boldsymbol{\varphi}}_1 | \widetilde{\boldsymbol{\varphi}}_2 | \cdots | \widetilde{\boldsymbol{\varphi}}_R), \tag{33}$$

where $\{\widetilde{\boldsymbol{\varphi}}_d\}_{d=1}^{D}$ are the generalized eigenvectors associated to the generalized eigenvalues $\widetilde{\lambda}_1 \geqq \widetilde{\lambda}_2 \geqq \cdots \geqq \widetilde{\lambda}_D$ of the following generalized eigenvalue problem:

$$\widetilde{\boldsymbol{S}}^{(m)} \widetilde{\boldsymbol{\varphi}} = \widetilde{\lambda} \widetilde{\boldsymbol{S}}^{(w)} \widetilde{\boldsymbol{\varphi}}. \tag{34}$$

Given $\widetilde{\boldsymbol{S}}^{(m)}$ and $\widetilde{\boldsymbol{S}}^{(w)}$, the computational complexity required for obtaining $\boldsymbol{T}_{LFDA}$ is $\mathcal{O}(RD^2)$. Below, we show how $\widetilde{\boldsymbol{S}}^{(m)}$ and $\widetilde{\boldsymbol{S}}^{(w)}$ can be computed efficiently.

Since

$$\begin{aligned} \widetilde{\boldsymbol{S}}^{(m)} &= \frac{1}{2} \sum_{i\,j=1}^{n} \widetilde{\boldsymbol{A}}^{(m)}_{i\,j} (\boldsymbol{x}_i \boldsymbol{x}_i^\top + \boldsymbol{x}_j \boldsymbol{x}_j^\top - \boldsymbol{x}_i \boldsymbol{x}_j^\top - \boldsymbol{x}_j \boldsymbol{x}_i^\top) \\ &= \sum_{i=1}^{n} \left(\sum_{j=1}^{n} \widetilde{\boldsymbol{A}}^{(m)}_{i\,j} \right) \boldsymbol{x}_i \boldsymbol{x}_i^\top - \sum_{i\,j=1}^{n} \widetilde{\boldsymbol{A}}^{(m)}_{i\,j} \boldsymbol{x}_i \boldsymbol{x}_j^\top, \end{aligned} \tag{35}$$

$\widetilde{\boldsymbol{S}}^{(m)}$ can be expressed in a matrix form as

$$\widetilde{\boldsymbol{S}}^{(m)} = \boldsymbol{X} \widetilde{\boldsymbol{L}}^{(m)} \boldsymbol{X}^\top, \tag{36}$$

where

$$\widetilde{\boldsymbol{L}}^{(m)} = \widetilde{\boldsymbol{D}}^{(m)} - \widetilde{\boldsymbol{A}}^{(m)}, \tag{37}$$

and $\widetilde{\boldsymbol{D}}^{(m)}$ is the n-dimensional diagonal matrix with i-th diagonal element being

$$\widetilde{\boldsymbol{D}}^{(m)}_{i\,i} = \sum_{j=1}^{n} \widetilde{\boldsymbol{A}}^{(m)}_{i\,j}. \tag{38}$$

Similarly, $\widetilde{\boldsymbol{S}}^{(w)}$ can be expressed in a matrix form as

$$\widetilde{\boldsymbol{S}}^{(w)} = \boldsymbol{X} \widetilde{\boldsymbol{L}}^{(w)} \boldsymbol{X}^\top, \tag{39}$$

where

$$\widetilde{\boldsymbol{L}}^{(w)} = \widetilde{\boldsymbol{D}}^{(w)} - \widetilde{\boldsymbol{A}}^{(w)}, \tag{40}$$

and $\widetilde{\boldsymbol{D}}^{(w)}$ is the n-dimensional diagonal matrix with i-th diagonal element being

$$\widetilde{\boldsymbol{D}}^{(w)}_{i\,i} = \sum_{j=1}^{n} \widetilde{\boldsymbol{A}}^{(w)}_{i\,j}. \tag{41}$$

$\widetilde{\boldsymbol{L}}^{(m)}$ and $\widetilde{\boldsymbol{L}}^{(w)}$ are n-dimensional matrices so could be very high dimensional. However, $\widetilde{\boldsymbol{L}}^{(w)}$ is block-diagonal if $\{\boldsymbol{x}_i\}_{i=1}^{n}$ are sorted according to the labels. Furthermore, diagonal submatrices of $\widetilde{\boldsymbol{L}}^{(w)}$ can be sparse if the affinity matrix $\boldsymbol{A}$ is sparsely defined (see Appendix 1. for detail). Therefore, calculating $\widetilde{\boldsymbol{S}}^{(w)}$ directly by Eq.(39) may be computationally efficient. On the other hand, computing $\widetilde{\boldsymbol{S}}^{(m)}$ directly

by Eq.(36) is not so efficient since $\widetilde{\boldsymbol{A}}^{(m)}$ is *dense*.

This problem can be alleviated as follows. $\widetilde{\boldsymbol{A}}^{(m)}$ can be decomposed as

$$\widetilde{\boldsymbol{A}}^{(m)} = \mathbf{1}\mathbf{1}^\top/n + \widetilde{\boldsymbol{A}}^{(m')}, \tag{42}$$

where $\mathbf{1}$ is the n-dimensional vector with all ones and $\widetilde{\boldsymbol{A}}^{(m')}$ is the n-dimensional matrix with (i,j)-th element being

$$\widetilde{\boldsymbol{A}}^{(m')}_{i\,j} = \begin{cases} (\boldsymbol{A}_{i\,j} - 1)/n & \text{if } y_i = y_j, \\ 0 & \text{if } y_i \neq y_j. \end{cases} \tag{43}$$

Then $\widetilde{\boldsymbol{S}}^{(m)}$ can be expressed as

$$\widetilde{\boldsymbol{S}}^{(m)} = \boldsymbol{X}\widetilde{\boldsymbol{D}}^{(m)}\boldsymbol{X}^\top - (\boldsymbol{X}\mathbf{1})(\boldsymbol{X}\mathbf{1})^\top/n - \boldsymbol{X}\widetilde{\boldsymbol{A}}^{(m')}\boldsymbol{X}^\top, \tag{44}$$

where $\widetilde{\boldsymbol{D}}^{(m)}$ is expressed by using $\widetilde{\boldsymbol{A}}^{(m')}$ as

$$\widetilde{\boldsymbol{D}}^{(m)}_{i\,i} = 1 + \sum_{j=1}^{n} \widetilde{\boldsymbol{A}}^{(m')}_{i\,j}. \tag{45}$$

$\widetilde{\boldsymbol{A}}^{(m')}$ is a block-diagonal matrix if $\{\boldsymbol{x}_i\}_{i=1}^n$ are sorted according to the labels, which implies that the third term in the right-hand side of Eq.(44) may be computed efficiently. Since the first two terms in the right-hand side of Eq.(44) can also be computed efficiently, computing $\widetilde{\boldsymbol{S}}^{(m)}$ by Eq.(44) may be more efficient than directly by Eq.(36).

To further improve computational efficiency, the affinity matrix $\boldsymbol{A}$ may be computed in a classwise manner since we do not need the affinity value for between-class pairs (see Eqs.(25) and (26)). This speeds up the nearest neighbor search which is often carried out when defining $\boldsymbol{A}$ (see Appendix 1.).

3.5 Comparison with Related Methods

Recently, novel methods have been proposed for linear supervised dimensionality reduction. Here, we qualitatively compare the methods with LFDA.

3.5.1 Neighborhood Component Analysis

First, we review the definition of a method called *neighborhood component analysis* (NCA) (Goldberger et al., 2005). The NCA transformation matrix $\boldsymbol{T}_{NCA}$ is defined as follows.

$$\boldsymbol{T}_{NCA} = \operatorname*{argmax}_{\boldsymbol{T}\in\mathbb{R}^{D\times R}} \left(\sum_{i=1}^{n} \sum_{j:y_j=y_i} p_{i\,j}(\boldsymbol{T}\boldsymbol{T}^\top) \right), \tag{46}$$

where

$$p_{i\,j}(\boldsymbol{U}) = \begin{cases} \dfrac{\exp\left\{-(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{U}(\boldsymbol{x}_i - \boldsymbol{x}_j)\right\}}{\sum_{k\neq i} \exp\left\{-(\boldsymbol{x}_i - \boldsymbol{x}_k)^\top \boldsymbol{U}(\boldsymbol{x}_i - \boldsymbol{x}_k)\right\}} & \text{if } i \neq j, \\ 0 & \text{if } i = j. \end{cases} \tag{47}$$

The above definition corresponds to maximizing the expected number of correctly classified samples by a stochastic variant of nearest neighbor classifiers. Therefore, NCA determines $\boldsymbol{T}$ so that *separability* of data samples in different classes is maximized. By definition, embedded samples obtained by NCA are well separated from each other. Our simulations in Section 4. show that NCA tends to preserve multimodal structure of the data well, although NCA does not take the multimodality of the data explicitly into account.

A crucial weakness of NCA is optimization: the optimization problem (46) is *non-convex*. Therefore, there is no guarantee that one can obtain the globally optimal solution. Goldberger et al. (2005) proposed using a gradient method for optimization:

$$\boldsymbol{T} \leftarrow \boldsymbol{T} + \varepsilon \nabla J_{NCA}(\boldsymbol{T}), \tag{48}$$

where ε (> 0) is the step size and the gradient $\nabla J_{NCA}(\boldsymbol{T})$ is given by

$$\begin{aligned} \nabla J_{NCA}(\boldsymbol{T}) = 2\boldsymbol{T} \sum_{i=1}^{n} \Bigg(& \left\{ \sum_{j:y_j=y_i} p_{i\,j}(\boldsymbol{T}\boldsymbol{T}^\top) \right\} \\ & \left\{ \sum_{j=1}^{n} p_{i\,j}(\boldsymbol{T}\boldsymbol{T}^\top)(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \right\} \\ & - \sum_{j:y_j=y_i} p_{i\,j}(\boldsymbol{T}\boldsymbol{T}^\top)(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \Bigg). \end{aligned} \tag{49}$$

3.5.2 Maximally Collapsing Metric Learning

We begin with reviewing the definition of a method called *maximally collapsing metric learning* (MCML) (Globerson and Roweis, 2006). Let

$$\boldsymbol{U}_{MCML} = \operatorname*{argmin}_{\boldsymbol{U}\in\mathbb{R}^{D\times D}} \left(\sum_{i\;j=1}^{n} p^*_{i\,j} \log \frac{p^*_{i\,j}}{p_{i\,j}(\boldsymbol{U})} \right) \quad \text{subject to } \boldsymbol{U} \in PSD(R), \tag{50}$$

where $PSD(R)$ is the set of all positive semidefinite matrix of rank R (i.e., R eigenvalues are positive and others are zero) and

$$p^*_{i\,j} \propto \begin{cases} 1 & \text{if } y_i = y_j, \\ 0 & \text{if } y_i \neq y_j. \end{cases} \tag{51}$$

$p^*_{i\,j}$ is normalized so that

$$\sum_{j\neq i} p^*_{i\,j} = 1. \tag{52}$$

Note that $p^*_{i\,j}$ is the 'ideal' value of $p_{i\,j}$, which is attained if all samples in the same class collapse into a single point while samples in other classes are mapped to other locations. In reality, any $\boldsymbol{U}$ may not be able to attain $p_{i\,j}(\boldsymbol{U}) = p^*_{i\,j}$, so MCML tries to finds $\boldsymbol{U}_{MCML}$ which gives the optimal approximation to $p^*_{i\,j}$ under the Kullback-Leibler divergence

(Kullback and Leibler, 1951). Once $\boldsymbol{U}_{MCML}$ is obtained, the MCML transformation matrix $\boldsymbol{T}_{MCML}$ is given by

$$\boldsymbol{T}_{MCML} = (\boldsymbol{\phi}_1|\boldsymbol{\phi}_2|\cdots|\boldsymbol{\phi}_R), \tag{53}$$

where $\{\boldsymbol{\phi}_d\}_{d=1}^R$ are the eigenvectors associated to the *positive* eigenvalues $\eta_1 \geqq \eta_2 \geqq \cdots \geqq \eta_R > 0$ of the following eigenvalue problem:

$$\boldsymbol{U}_{MCML}\boldsymbol{\phi} = \eta\boldsymbol{\phi}. \tag{54}$$

MCML does not take the multimodality of the data into account since it requires the samples in the same class to collapse into a *single* point. Nevertheless, our simulations in Section 4. show that it works reasonably well even for multimodal data.

A drawback of MCML is also optimization: the optimization problem (50) is convex only when $R = D$, i.e., the dimensionality is not reduced but the only the *metric* of the original space is changed. This means that if $R < D$ (which is our primal focus in this paper), the unique optimal solution $\boldsymbol{U}_{MCML}$ does not exist. Therefore, there is no guarantee that one can obtain the globally optimal solution $\boldsymbol{T}_{MCML}$.

Globerson and Roweis (2006) proposed the following heuristic algorithm to *approximate* $\boldsymbol{T}_{MCML}$: first, the optimization problem (50) with $R = D$ is solved:

$$\widehat{\boldsymbol{U}}_{MCML} = \operatorname*{argmin}_{\boldsymbol{U}\in\mathbb{R}^{D\times D}} \left(\sum_{i\ j=1}^{n} p^*_{i\ j} \log \frac{p^*_{i\ j}}{p_{i\ j}(\boldsymbol{U})} \right) \quad \text{subject to } \boldsymbol{U} \in PSD(D). \tag{55}$$

The algorithm for obtaining the solution $\widehat{\boldsymbol{U}}_{MCML}$ will be explained below. Then eigenvalue decomposition of $\widehat{\boldsymbol{U}}_{MCML}$ is carried out and eigenvalues $\widehat{\eta}_1 \geqq \widehat{\eta}_2 \geqq \cdots \geqq \widehat{\eta}_D$ and associated eigenvectors $\{\widehat{\boldsymbol{\phi}}_d\}_{d=1}^D$ are obtained:

$$\widehat{\boldsymbol{U}}_{MCML}\widehat{\boldsymbol{\phi}} = \widehat{\eta}\widehat{\boldsymbol{\phi}}. \tag{56}$$

Then $\{\boldsymbol{\phi}_d\}_{d=1}^R$ appeared in Eq.(53) are replaced by $\{\widehat{\boldsymbol{\phi}}_d\}_{d=1}^R$, which yields

$$\boldsymbol{T}_{MCML} \approx (\widehat{\boldsymbol{\phi}}_1|\widehat{\boldsymbol{\phi}}_2|\cdots|\widehat{\boldsymbol{\phi}}_R). \tag{57}$$

This approximation is shown to be practically useful, although there seems to be no theoretical analysis for this approximation.

Even though $\widehat{\boldsymbol{U}}_{MCML}$ uniquely exists, its analytic form is not known yet. Globerson and Roweis (2006) proposed using the following alternate iterative procedure for obtaining $\widehat{\boldsymbol{U}}_{MCML}$.

$$\boldsymbol{U} \leftarrow \boldsymbol{U} - \varepsilon\nabla J_{MCML}(\boldsymbol{U}), \tag{58}$$

$$\boldsymbol{U} \leftarrow \sum_{d=1}^{D} \max(0, \widehat{\eta}_d)\widehat{\boldsymbol{\phi}}_d\widehat{\boldsymbol{\phi}}_d^\top, \tag{59}$$

where ε (> 0) is the step size, $\widehat{\eta}_d$ and $\widehat{\boldsymbol{\phi}}_d$ are eigenvalues and eigenvectors of $\boldsymbol{U}$, and the gradient $\nabla J_{MCML}(\boldsymbol{U})$ is given by

$$\nabla J_{MCML}(\boldsymbol{U}) = \sum_{i\ j=1}^{n} (p^*_{i\ j} - p_{i\ j}(\boldsymbol{U}))(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top. \tag{60}$$

3. 5. 3 Discussion

NCA and MCML appear to work reasonably well for dimensionality reduction of multimodal data samples (see Section 4.). However, they both involve non-convex optimization problems so there is no guarantee that one can obtain the optimal solution.

In NCA, the non-convex optimization problem (46) is solved by the gradient ascent iteration (48), which is computationally rather inefficient and the quality of the obtained solution depends on the *initialization* of the matrix $\boldsymbol{U}$. Also, the choice of the step size ε is troublesome: if the step size is small enough, convergence to one of the local optima is guaranteed but such a choice makes convergence very slow; if the step size is too large, gradient flow oscillates and convergence may not be guaranteed anymore. Furthermore, the choice of termination condition in the iterative algorithm is often cumbersome in practice.

MCML has an advantage over NCA in computation: there exists the analytic approximation (57) to the optimal solution, which can be computed using the solution of another convex optimization problem (55). However, MCML still relies on the gradient-based alternate iterative algorithm (58)–(59) to solve the convex optimization problem (55), which is expensive in computation since eigenvalue decomposition should be carried out in each iteration (see Eq.(59)). Furthermore, the difficulty of appropriately choosing the step size and termination condition still remains.

On the other hand, in LFDA, an *analytic* form of the optimal solution is available (see Eq.(33)), which can be efficiently computed by solving the generalized eigenvalue problem (34). Therefore, LFDA is computationally more efficient and *reliable* than NCA and MCML.

3. 6 Kernel LFDA for Non-Linear Dimensionality Reduction

So far, we focused on linear dimensionality reduction. Here we extend our discussion to non-linear dimensionality reduction scenarios.

From Eqs.(36) and (39), the generalized eigenvalue problem (34) can be expressed as

$$\boldsymbol{X}\widetilde{\boldsymbol{L}}^{(m)}\boldsymbol{X}^\top\widetilde{\boldsymbol{\varphi}} = \widetilde{}\boldsymbol{X}\widetilde{\boldsymbol{L}}^{(w)}\boldsymbol{X}^\top\widetilde{\boldsymbol{\varphi}}. \tag{61}$$

When the range of $\boldsymbol{X}$ spans the entire space $\mathbb{R}^D$, any vector $\widetilde{\boldsymbol{\varphi}} \in \mathbb{R}^D$ can be expressed by using some vector $\widetilde{\boldsymbol{\alpha}} \in \mathbb{R}^n$ as

$$\widetilde{\boldsymbol{\varphi}} = \boldsymbol{X}\widetilde{\boldsymbol{\alpha}}. \tag{62}$$

Then, multiplying Eq.(61) by $\boldsymbol{X}^\top$ from the left-hand side yields

$$\boldsymbol{K}\widetilde{\boldsymbol{L}}^{(m)}\boldsymbol{K}\widetilde{\boldsymbol{\alpha}} = \widetilde{}\boldsymbol{K}\widetilde{\boldsymbol{L}}^{(w)}\boldsymbol{K}\widetilde{\boldsymbol{\alpha}}, \tag{63}$$

where $\boldsymbol{K}$ is the n-dimensional matrix with the (i,j)-th element being

$$\boldsymbol{K}_{i\,j} = \boldsymbol{x}_i^\top \boldsymbol{x}_j. \tag{64}$$

This implies that $\{\boldsymbol{x}_i\}_{i=1}^n$ appear only in terms of their inner products. Therefore, we can obtain a non-linear variant of LFDA by the *kernel trick* (Vapnik, 1998; Schölkopf et al., 1998), which is explained below.

Let us consider a non-linear mapping $(\boldsymbol{x})$ from $\mathbb{R}^D$ to a *reproducing kernel Hilbert space* $\mathcal{H}$ (Aronszajn, 1950). Let $K(\boldsymbol{x},\boldsymbol{x}')$ be the reproducing kernel of $\mathcal{H}$. A typical choice of the kernel function would be the Gaussian kernel:

$$K(\boldsymbol{x},\boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|^2}{2\sigma^2}\right), \tag{65}$$

with $\sigma > 0$. For other choices, see, e.g., Wahba (1990), Vapnik (1998), and Schölkopf and Smola (2002). Because of the reproducing property of $K(\boldsymbol{x},\boldsymbol{x}')$, $\boldsymbol{K}$ is now the *kernel matrix*, i.e., the (i,j)-th element is given by

$$\boldsymbol{K}_{i\,j} = \langle (\boldsymbol{x}_i), (\boldsymbol{x}_j)\rangle = K(\boldsymbol{x}_i,\boldsymbol{x}_j), \tag{66}$$

where $\langle\cdot,\cdot\rangle$ denotes the inner product in $\mathcal{H}$. Then the embedded image of $(\boldsymbol{x}')$ by LFDA in $\mathcal{H}$ is given by

$$(\widetilde{\boldsymbol{\alpha}}_1|\widetilde{\boldsymbol{\alpha}}_2|\cdots|\widetilde{\boldsymbol{\alpha}}_R)^\top \begin{pmatrix} K(\boldsymbol{x}_1,\boldsymbol{x}') \\ K(\boldsymbol{x}_2,\boldsymbol{x}') \\ \vdots \\ K(\boldsymbol{x}_n,\boldsymbol{x}') \end{pmatrix}, \tag{67}$$

where $\{\widetilde{\boldsymbol{\alpha}}_d\}_{d=1}^D$ is the generalized eigenvectors associated to the generalized eigenvalues $\widetilde{}_1 \geqq \widetilde{}_2 \geqq \cdots \geqq \widetilde{}_D$ of the generalized eigenvalue problem (63). Note that, following the discussion in Section 3. 4, we can compute $\boldsymbol{K}\widetilde{\boldsymbol{L}}^{(m)}\boldsymbol{K}$ efficiently by

$$\boldsymbol{K}\widetilde{\boldsymbol{L}}^{(m)}\boldsymbol{K} = \boldsymbol{K}\widetilde{\boldsymbol{D}}^{(m)}\boldsymbol{K} - (\boldsymbol{K}\boldsymbol{1})(\boldsymbol{K}\boldsymbol{1})^\top/n - \boldsymbol{K}\widetilde{\boldsymbol{A}}^{(m')}\boldsymbol{K}. \tag{68}$$

4. Numerical Results

In this section, we apply the proposed and existing dimensionality reduction methods to benchmark data sets for visualization. In LPP and LFDA, the affinity matrix is determined by the local scaling method. For LFDA, the affinity matrix is computed in a classwise manner (see Section 3. 4).

We use *Letter recognition*, *Segment*, *Thyroid disease*, and *Iris* data sets available from the UCI machine learning repository (Blake and Merz, 1998). Table 1 describes the specification of employed data sets. Each data set contains three types of samples specified by '×', '+', and '∘'. We treat '×' and '+' as a single class, by which two-class problem is created. That is, '×' and '+' are treated as *hidden* labels, and for embedding, we only use the information whether each sample belongs to the '×'-and-'+' class or the '∘' class. Our interest is to evaluate separability between two classes (i.e., '×' and '+' are well separated from '∘') and multimodality preservation within a class (i.e., '×' and '+' are well grouped). Figure 2–5 show the embedded samples in two-dimensional spaces obtained by LFDA, FDA, LPP, NCA, and MCML.

Table 1 List of employed data sets.

Data Set	D	'×'-and-'+' class	'∘' class
Letter recognition	16	'A' & 'C'	'B'
Segment	18	'Brickface' & 'Sky'	'Foliage'
Thyroid disease	5	'Hyper' & 'Hypo'	'Normal'
Iris	4	'Setosa' & 'Virginica'	'Versicolour'

First, we compare the results obtained by LFDA, FDA, and LPP. For *Letter recognition* and *Segment* data sets (see the top three graphs in Figure 2 and Figure 3), FDA achieves good separability (i.e., '×' and '+' are well separated from '∘'), but multimodality of the '×'-and-'+' class is lost (i.e., '×' and '+' are mixed). On the other hand, LFDA provides well separate embedding, and at the same time, multimodal structure of the data is well preserved. LPP clearly separates the samples in two clusters, but labels are mixed in one of the clusters. For *Thyroid disease* data set (see the top three graphs in Figure 4), LFDA, FDA, and LPP separate '×' and '+' from '∘' well. FDA collapses multimodal structure of the '×'-and-'+' class, while LFDA and LPP well preserve the multimodality. For *Iris* data set (see the top three graphs in Figure 5), FDA tends to mix samples of different classes, while LFDA can achieve good separability. We conjecture this difference is due to the fact that LFDA has more degree of freedom for enhancing separability than FDA (see Section 3. 3). In addition to the separability, LFDA also clearly preserves the multimodal structure of the '×'-and-'+' class. LPP also works well for this data set because three clusters are well separated from each other in the original high-dimensional space. Overall, LFDA is found to be more appropriate for embedding labeled multimodal data samples than FDA and LPP.

Next, we compare LFDA with NCA and MCML. For *Letter recognition* data set (see the top and the bottom two graphs in Figure 2), LFDA, NCA and MCML achieve good separability (i.e., '×' and '+' are well separated from '∘'). However, MCML collapses '×' and '+' into a single cluster (which is actually aimed in MCML) while LFDA and NCA preserve multimodal structure of the '×'-and-'+' class quite clearly. Note that the NCA result may vary if the initial

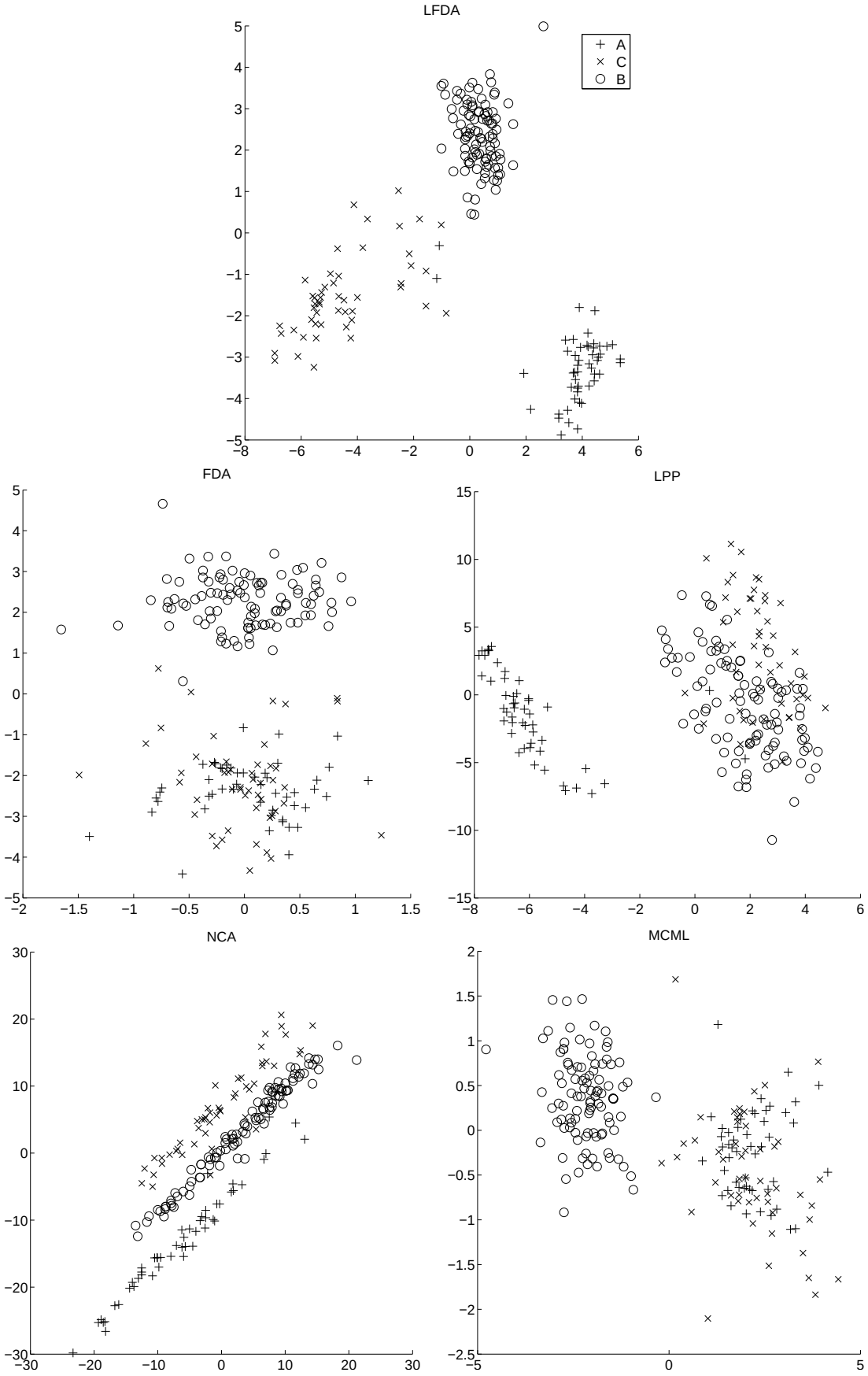


Fig. 2 Data visualization for *Letter recognition* data set.

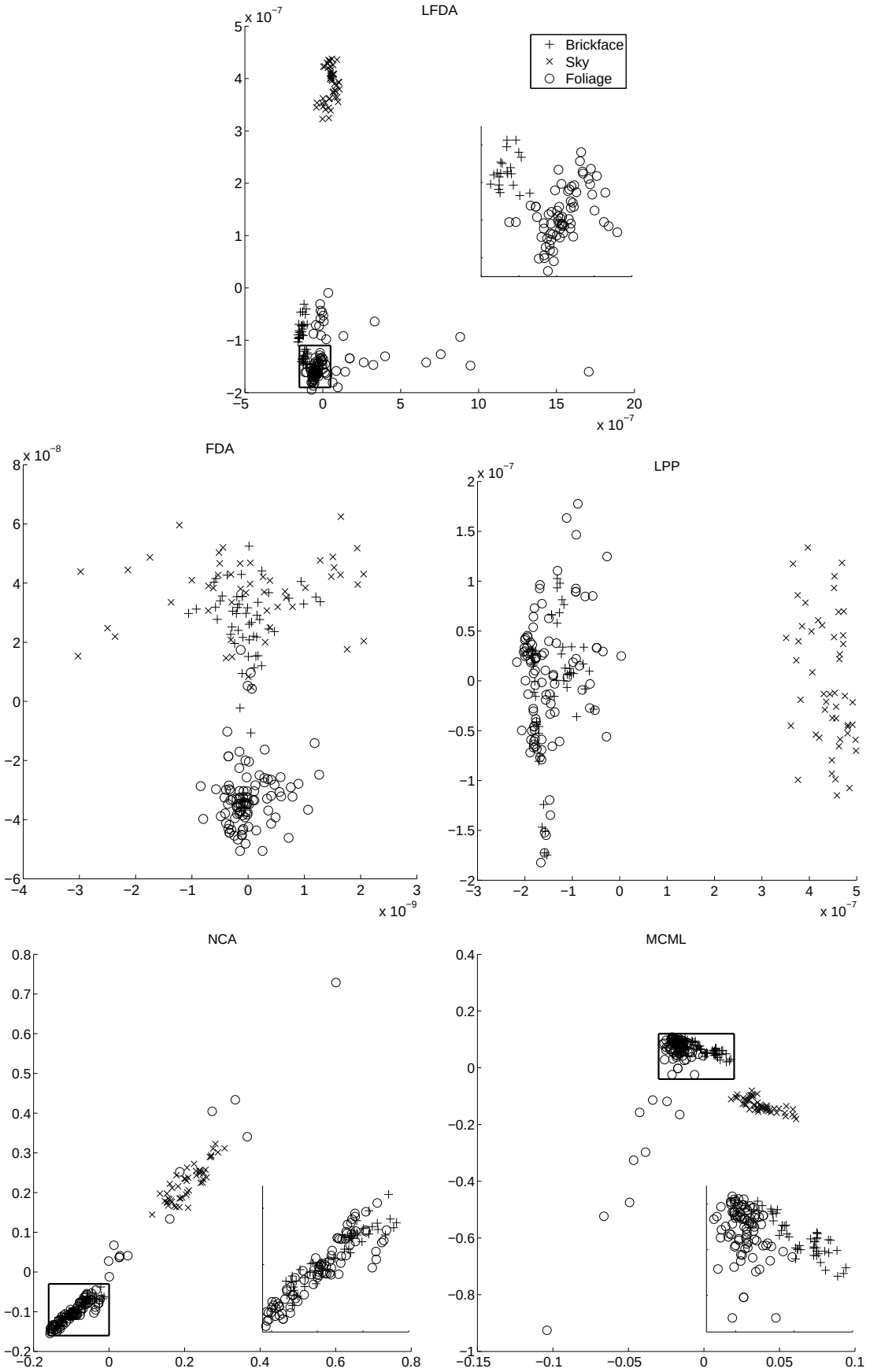


Fig. 3 Data visualization for *Segment* data set. In the plots of LFDA, NCA, and MCML, the area surrounded by the bold line is enlarged for clear visibility.

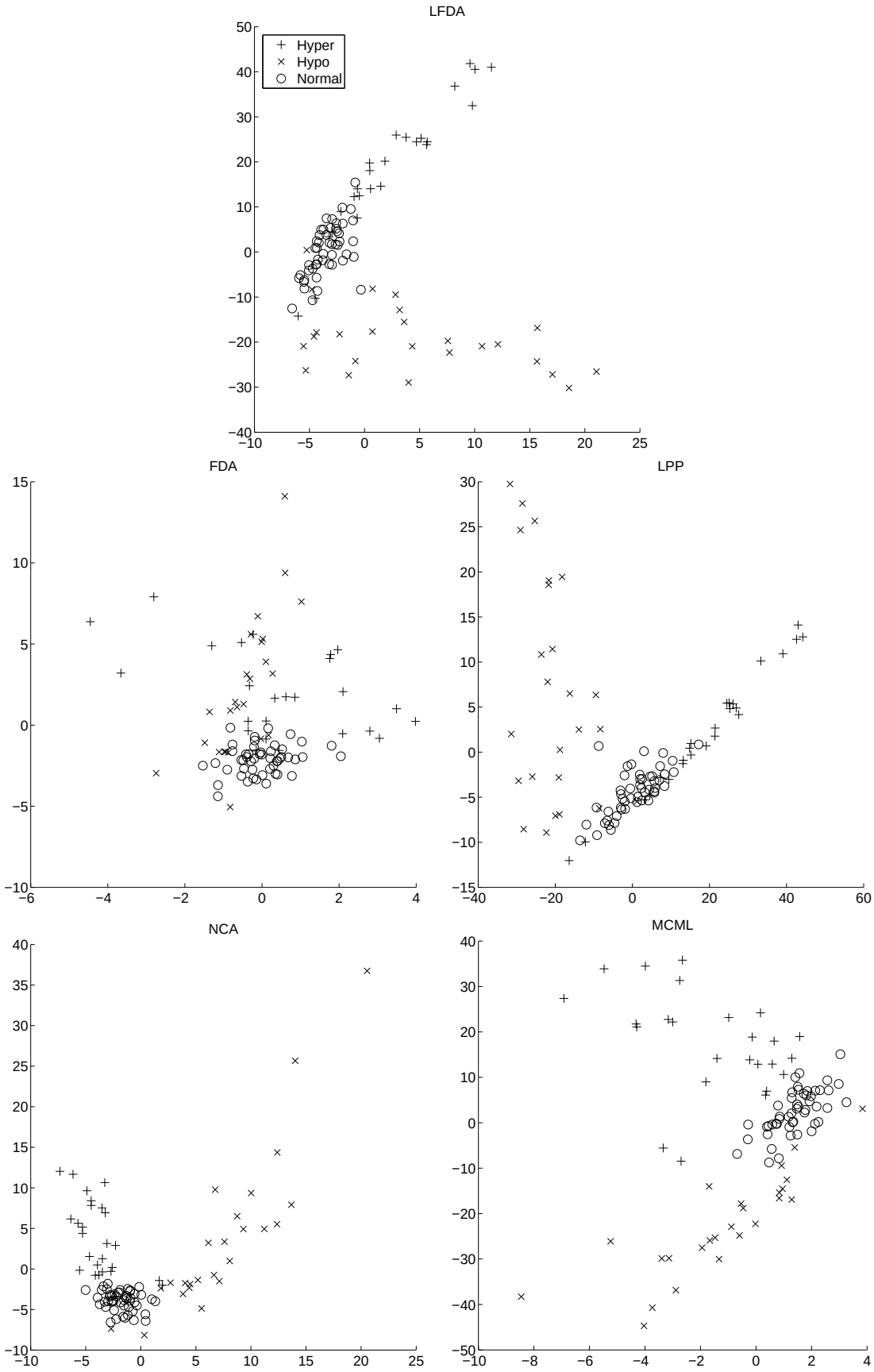


Fig. 4 Data visualization for *Thyroid disease* data set.

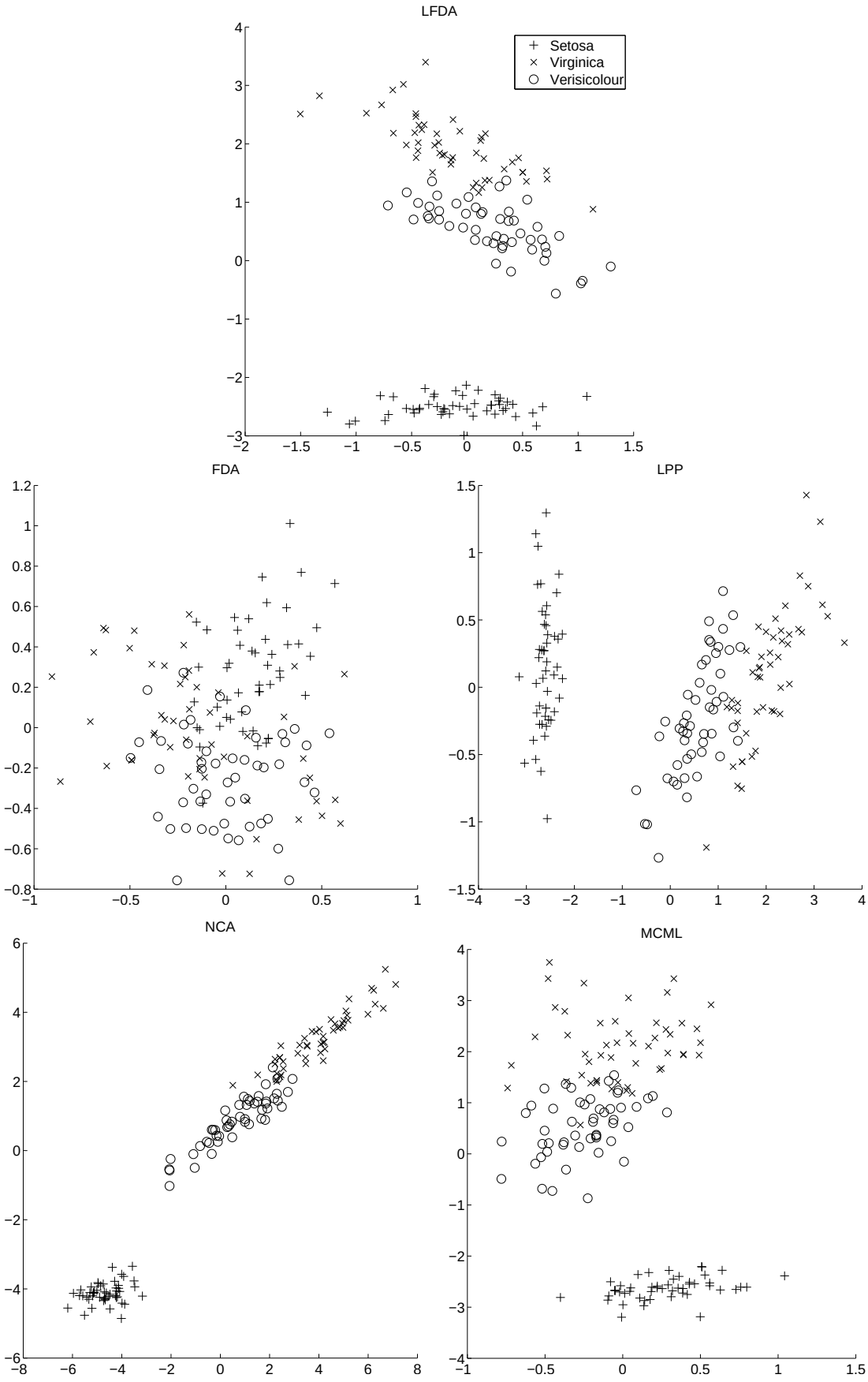


Fig. 5 Data visualization for *Iris* data set.

value for the gradient ascent algorithm is changed. For *Segment* data set (see the top and the bottom two graphs in Figure 3), NCA mixes '+' and 'o', which may be caused by local optima. LFDA and MCML work quite well in both between-class separation and within-class multimodality preservation. For *Thyroid disease* and *Iris* data sets (see the top and the bottom two graphs in Figure 4 and Figure 5), LFDA, NCA and MCML all work very well in between-class separation and within-class multimodality preservation. Overall, NCA sometimes gives a slightly poor result due to local minima, while MCML can lose within-class multimodal structure (although it is the aim of MCML). On the other hand, LFDA consistently gives suitable results for all four data sets. Furthermore, LFDA is computationally more efficient and reliable than NCA and MCML, thanks to the analytic form of the optimal solution. Therefore, LFDA would be a promising alternative to the state-of-the-art NCA and MCML.

Based on the above simulation results, we conclude that LFDA is useful in visualizing multimodal data samples.

5. Conclusions

We discussed the problem of supervised dimensionality reduction. FDA (Fisher, 1936; Fukunaga, 1990; Duda et al., 2001) works well for this purpose, given data samples of each class are *unimodal* Gaussian. On the other hand, samples in some class can be multimodal, e.g., when multi-class classification problems are solved by a set of two-class 'one-versus-rest' problems. LPP (He and Niyogi, 2004) is suitable for embedding multimodal data, although it is an unsupervised method so it does not necessarily work appropriately in supervised dimensionality reduction scenarios. In this paper, we proposed LFDA, which effectively combines the ideas of FDA and LPP: between-class separability is maximized while within-class *local* structure is preserved. The derivation of LFDA is based on the novel *pairwise* interpretation of the original FDA (see Section 3. 1): FDA keeps in-class data pairs close and between-class data pairs apart. We experimentally showed that LFDA can preserve multimodal structure of the data better than FDA (see Section 4.). Furthermore, we showed that LFDA can even provide more separate embedding than FDA since it has more degree of freedom for achieving high separability than LFDA (see Section 3. 3).

We also compared LFDA with the state-of-the-art methods such as NCA and MCML (see Section 3. 5). Although NCA and MCML do not explicitly take the multimodality of the data into account, our experiments showed that they are reasonably useful in dimensionality reduction of multimodal data samples. However, they involve non-convex optimization problem so there is no guarantee that one can obtain the globally optimal solution. Furthermore, they employ gradient-based iterative algorithms for optimization, which are rather inefficient and which require careful choice of the step size and termination condition. On the other hand, in LFDA, an analytic form of the optimal solution is available, which can be reliably computed by solving a generalized eigenvalue problem. We also provided an efficient computation method of the LFDA solution (see Section 3. 4).

If between-class separability is the only issue and we do not care local structure of the data, we may put the affinity values for in-class pairs of samples very close to zero in LFDA. However, we experimentally confirmed that this does not give reasonable embedding. This implies that, in order to obtain useful embedding, it is very important to have a reasonable objective function for the in-class pairs of samples. In LFDA, such an objective function is naturally derived as an extension of FDA.

We showed in Section 3. 6 that a non-linear variant of LFDA can be obtained by employing the standard kernel trick (Schölkopf and Smola, 2002). FDA, LPP, and MCML can also be kernelized similarly (Baudat and Anouar, 2000; Mika et al., 2003; Belkin and Niyogi, 2003; He and Niyogi, 2004; Globerson and Roweis, 2006). As shown in these papers, the performance of the kernelized methods heavily depend on the choice of the family and parameters of kernel functions. How to optimally determine the kernel function for supervised dimensionality reduction seems to be an open problem, which needs to be explored.

LFDA is an efficient dimensionality reduction method, given an appropriate affinity matrix of the data samples. In this paper, we simply employed the standard affinity matrix (see Appendix 1.). Although the standard choice appeared to be quite reasonable in experiments (see Section 4.), it is important to find the optimal way to define the affinity matrix in the context of supervised dimensionality reduction.

Acknowledgments

The author would like to thank Klaus-Robert Müller, Hideki Asoh, and Stefan Harmeling for their fruitful comments. He also acknowledges financial support from MEXT (Grant-in-Aid for Young Scientists 17700142). A part of this work has been done when MS is staying at University of Edinburgh, which is supported by Erasmus Mundus Scholarship. He would like to thank Sethu Vijayakumar for the warm hospitality.

Appendix

1. Definitions of Affinity Matrix

In this appendix, we briefly review typical choices of the affinity matrix $\boldsymbol{A}$.

1.1 Heat Kernel

Belkin and Niyogi (2003) proposed defining $\boldsymbol{A}_{i\,j}$ as

$$\boldsymbol{A}_{i\,j} = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{\sigma^2}\right), \qquad \text{(A·1)}$$

where σ (> 0) is a tuning parameter which determines the 'decay' of the affinity.

1.2 Euclidean Neighbor

The heat kernel gives a non-sparse affinity matrix. It would be computationally advantageous if the affinity matrix is sparcified. A sparse affinity matrix may be obtained as follows Tenenbaum et al. (2000). For ϵ (> 0) which is a tuning parameter, $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are judged to be *neighbors* if

$$\|\boldsymbol{x}_i - \boldsymbol{x}_j\| \leqq \epsilon. \qquad \text{(A·2)}$$

Then $\boldsymbol{A}_{i\,j}$ is defined by Eq.(A·1) for two neighboring samples and $\boldsymbol{A}_{i\,j} = 0$ for non-neighbors.

This definition includes two tuning parameters (ϵ and σ), which are rather troublesome to be determined. To ease the problem, we may simply let $\boldsymbol{A}_{i\,j} = 1$ if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are neighbors and $\boldsymbol{A}_{i\,j} = 0$ otherwise. This corresponds to putting $\sigma = \infty$.

1.3 Nearest Neighbor

Tuning the distance threshold ϵ is practically rather cumbersome since the relation between the number of neighbors and the value of ϵ is not intuitively clear. Another option to determine neighbors is to directly specify the number of neighbors (Roweis and Saul, 2000; Tenenbaum et al., 2000). Let $kNN(\boldsymbol{x}_i)$ be the set of k nearest neighbor samples of $\boldsymbol{x}_i$ under the Euclidean distance, where k is a tuning parameter. If $\boldsymbol{x}_j \in kNN(\boldsymbol{x}_i)$ or $\boldsymbol{x}_i \in kNN(\boldsymbol{x}_j)$, we regard $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ as neighbors; otherwise they are regarded as non-neighbors. Then the affinity matrix may be defined by the heat kernel or in the simple zero-one manner.

1.4 Local Scaling

A drawback of the above definitions is that the affinity is computed globally in the same way. The density of data samples may be different depending on regions and it would be more appropriate to take the local scaling of the data into account. Following this idea, Zelnik-Manor and Perona (2005) proposed defining the affinity matrix as

$$\boldsymbol{A}_{i\,j} = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{\sigma_i \sigma_j}\right). \qquad \text{(A·3)}$$

σ_i represents the local scaling of the data samples around $\boldsymbol{x}_i$, which is defined by

$$\sigma_i = \|\boldsymbol{x}_i - \widetilde{\boldsymbol{x}}_i^{(k)}\|, \qquad \text{(A·4)}$$

where $\widetilde{\boldsymbol{x}}_i^{(k)}$ is the k-th nearest neighbor of $\boldsymbol{x}_i$. The parameter k is a tuning parameter, but Zelnik-Manor and Perona (2005) demonstrated that $k = 7$ is a 'universal' value, by which no tuning parameter remains. This would be a convenient choice for those who do not have any subjective/prior preference. We employed this definition all though the paper.

For computational efficiency, we may further sparsify the above affinity matrix based on, e.g., the nearest neighbor idea.

References

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337–404, 1950.

G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12(10): 2385–2404, 2000.

M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.

F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, Providence, R.I., 1997.

R. O. Duda, P. E. Hart, and D. G. Stor. *Pattern Classification*. Wiley, New York, 2001.

R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, Inc., Boston, second edition, 1990.

A. Globerson and S. Roweis. Metric learning by collapsing classes. In *Advances in Neural Information Processing Systems 18*, pages 451–458, Cambridge, MA, 2006. MIT Press.

J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 513–520. MIT Press, Cambridge, MA, 2005.

X. He and P. Niyogi. Locality preserving projections. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.

S. Mika, G. Rätsch, J. Weston, B. Schölkopf, A. Smola, and K.-R. Müller. Constructing descriptive and discriminative

nonlinear features: Rayleigh coefficients in kernel feature spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):623–628, 2003.

S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

V. N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

G. Wahba. *Spline Model for Observational Data*. Society for Industrial and Applied Mathematics, Philadelphia and Pennsylvania, 1990.

L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1601–1608. MIT Press, Cambridge, MA, 2005.

Equivalence of Non-Iterative Algorithms for Simultaneous Low Rank Approximations of Matrices

Kohei INOUE†, Kenji HARA†, and Kiichi URAHAMA†

† Department of Visual Communication Design, Faculty of Design, Kyushu University
4-9-1, Shiobaru, Minami-ku, Fukuoka, 815-8540, Japan

Abstract Recently four non-iterative algorithms for simultaneous low rank approximations of matrices (SLRAM) have been presented by several researchers. In this paper, we show that those algorithms are equivalent to each other because they are reduced to the eigenvalue problems of row-row and column-column covariance matrices of given matrices. Also, we show a relationship between the non-iterative algorithms and another algorithm which is claimed to be an analytical algorithm for the SLRAM. Experimental results show that the analytical algorithm does not necessarily give the optimal solution of the SLRAM. Additionally, we show a relationship between the above algorithms and non-iterative generalized low rank approximation of matrices which is also a non-iterative algorithm for the SLRAM.

Key words dimensionality reduction, simultaneous low rank approximations of matrices, higher-order singular value decomposition, two-dimensional PCA

1. Introduction

Dimensionality reduction is an important topic in computer vision and pattern recognition researches. Traditional methods for reducing the dimensionality of data, such as principal component analysis (PCA) and singular value decomposition (SVD) are based on the vector space model. Therefore each datum must be represented as a vector. If the original data are given as a set of matrices then each matrix has to be transformed into a vector preliminarily. However, these transformations do not preserve the spatial relationships of neighboring elements in each matrix. So the correlations between neighboring elements of matrices are not fully used in the vector space model. Moreover, the dimension of the vector space might become very high.

Recently, alternative methods for reducing the dimensionality of matrices have been presented by several researchers. Those methods are classified into iterative or non-iterative algorithms for the simultaneous low rank approximations of matrices (SLRAM). In this paper, we discuss the relationships of those methods and show the equivalence of four non-iterative algorithms for the SLRAM.

The rest of this paper is organized as follows. Section 2 reviews related works. Section 3 formulates the SLRAM and summarizes an iterative algorithm. Section 4 shows the equivalence of four non-iterative algorithms, and shows the relationships with two other non-iterative algorithms. Section 5 shows experimental results. Section 6 concludes the paper.

2. Related Work

The problem of low rank approximations of matrices has recently received broad attention in areas such as computer vision, machine learning and pattern recognition. Yang et al. [1] proposed a two-dimensional PCA (2DPCA) which can treat an image as a matrix without transformation into a vector. However, the 2DPCA is approximately equivalent to the traditional PCA operated on the row vectors of matrices [2], [3]. Cai et al. [4] considered an image as a second-order tensor and proposed tensor subspace learning algorithms (tensor PCA, tensor LDA). Ding and Ye [5] proposed a two-dimensional SVD (2dSVD) based on row-row and column-column covariance matrices and studied the optimality properties of the 2dSVD. Zhang et al. [3] adopted a natural representation for images using eigenimages. Inoue and Urahama [6] proposed a dyadic SVD (DSVD) based on the higher-order SVD (HOSVD) presented by Lathauwer et al. [7]. In this paper, we show the equivalence of the tensor PCA, 2dSVD, eigenimage method and DSVD. These methods give approximate solutions of the SLRAM. Ye [8], [9] proposed an iterative algorithm for the SLRAM. His method is called the generalized low rank approximations of matri-

ces (GLRAM). Liang and Shi [10] proposed an analytical algorithm for the SLRAM and proved its optimality. However, their analytical algorithm does not necessarily give the optimal solution of the SLRAM. We show this with a counterexample for their claim. Liu and Chen [13] proposed non-iterative generalized low rank approximation of matrices (NIGLRAM) which is also a non-iterative algorithm for the SLRAM. We show a relationship between the above non-iterative algorithms and the NIGLRAM.

3. Simultaneous Low Rank Approximations of Matrices

Let $A_i \in \Re^r \times \Re^c$ for $i = 1, ..., n$, where $\Re^r \times \Re^c$ denotes the product space of real r- and c-dimensional spaces $\Re^r$ and $\Re^c$. Ye [8], [9] considered the following optimization problem:

$$\begin{aligned} &\min_{L,R,M_i} \sum_{i=1}^{n} \|A_i - LM_iR^T\|_F^2 \\ &\text{subj.to } L^TL = I_{\tilde{r}},\ R^TR = I_{\tilde{c}}, \end{aligned} \tag{1}$$

where $L \in \Re^r \times \Re^{\tilde{r}}$, $R \in \Re^c \times \Re^{\tilde{c}}$, $M_i \in \Re^{\tilde{r}} \times \Re^{\tilde{c}}$ for $i = 1, ..., n$. $I_{\tilde{r}} \in \Re^{\tilde{r}} \times \Re^{\tilde{r}}$ and $I_{\tilde{c}} \in \Re^{\tilde{c}} \times \Re^{\tilde{c}}$ are the identity matrices, where $\tilde{r} \leqq r$ and $\tilde{c} \leqq c$. In this paper, we call this problem the simultaneous low rank approximations of matrices (SLRAM). Let E be the objective function in Eq. (1). Then E can be transformed into $E = \sum_{i=1}^{n} \mathrm{tr}(M_iM_i^T - 2LM_iR^TA_i^T + A_iA_i^T)$. From $\partial E/\partial M_i = 2M_i^T - 2R^TA_i^TL = 0$, we obtain $M_i = L^TA_iR$. Substitution of this equation into Eq. (1) gives

$$\begin{aligned} &\max_{L,R} \sum_{i=1}^{n} \|L^TA_iR\|_F^2 \\ &\text{subj.to } L^TL = I_{\tilde{r}},\ R^TR = I_{\tilde{c}}. \end{aligned} \tag{2}$$

The solutions L^*, R^* of Eq. (2) are also the solutions of Eq. (1) and the minimizer for M_i in Eq. (1) is given by $M_i^* = L^{*T}A_iR^*$. Ye [8], [9] presented the following iterative algorithm:

[GLRAM]

Step 0: Initialize L as $L^{(\xi)} = [I_{\tilde{r}}, 0]^T$, $\xi = 0$.

Step 1: Compute the $\tilde{c}$ eigenvectors $\phi_1^{R\,(\xi+1)}, ..., \phi_{\tilde{c}}^{R\,(\xi+1)}$ of $\sum_{i=1}^{n} A_i^TL^{(\xi)}L^{(\xi)\,T}A_i$ corresponding to the largest $\tilde{c}$ eigenvalues and form $R^{(\xi+1)} = [\phi_1^{R\,(\xi+1)}, ..., \phi_{\tilde{c}}^{R\,(\xi+1)}]$.

Step 2: Compute the $\tilde{r}$ eigenvectors $\phi_1^{L\,(\xi+1)}, ..., \phi_{\tilde{c}}^{L\,(\xi+1)}$ of $\sum_{i=1}^{n} A_iR^{(\xi+1)}R^{(\xi+1)\,T}A_i^T$ corresponding to the largest $\tilde{r}$ eigenvalues and form $L^{(\xi+1)} = [\phi_1^{L\,(\xi+1)}, ..., \phi_{\tilde{c}}^{L\,(\xi+1)}]$.

Step 3: If $L^{(\xi+1)}, R^{(\xi+1)}$ are not convergent then increase ξ by 1 and go to Step 1, otherwise proceed to Step 4.

Step 4: Let $L^* = L^{(\xi+1)}, R^* = R^{(\xi+1)}$ and compute $M_i^* = L^{*T}A_iR^*$ for $i = 1, ..., n$.

This algorithm is called the generalized low rank approximations of matrices (GLRAM). The GLRAM monotonically non-increases E, hence it converges to a local optimum of Eq. (1). Benito and Peña [11] also presented a similar algorithm and called it the alternating least squares (ALS) [12] algorithm.

4. Non-Iterative Algorithms

The GLRAM described above is an iterative algorithm and therefore is time-consuming. On the other hand, several non-iterative algorithms for the SLRAM have also been presented recently. Those algorithms give near-optimal solutions and are computationally efficient. In this section, we summarize those non-iterative algorithms and show their equivalence.

4.1 Tensor PCA

Let $M_i = L^TA_iR$ be the projection of A_i. Then the Tensor PCA presented by Cai et al. [4] is formulated as follows:

$$\begin{aligned} &\max_{L,R} \sum_{i=1}^{n} \|M_i - \bar{M}\|_F^2 \\ &\text{subj.to } L^TL = I_r,\ R^TR = I_c, \end{aligned} \tag{3}$$

where $\bar{M} = \sum_{i=1}^{n} M_i/n = L^T\bar{A}R$, $\bar{A} = \sum_{i=1}^{n} A_i/n$. Note that the constraints in Eq. (3) are different from those in Eq. (1) or (2), that is, L and R in Eq. (3) are square matrices of order r and c respectively. Let E^{TPCA} be the objective function in Eq. (3). Then

$$E^{\mathrm{TPCA}} = \sum_{i=1}^{n} \mathrm{tr}[(M_i - \bar{M})(M_i - \bar{M})^T] \tag{4}$$

$$= \sum_{i=1}^{n} \mathrm{tr}(L^T\tilde{A}_iRR^T\tilde{A}_i^TL) \tag{5}$$

$$= \mathrm{tr}\left[L^T\left(\sum_{i=1}^{n} \tilde{A}_iRR^T\tilde{A}_i^T\right)L\right], \tag{6}$$

where $\tilde{A}_i = A_i - \bar{A}$. It follows from $R^TR = I_c$ that $RR^T = I_c$. Consequently

$$E^{\mathrm{TPCA}} = \mathrm{tr}\left[L^T\left(\sum_{i=1}^{n} \tilde{A}_i\tilde{A}_i^T\right)L\right]. \tag{7}$$

Hence, if $\phi_1^L, ..., \phi_r^L$ are the eigenvectors of $\tilde{C}_L = \sum_{i=1}^{n} \tilde{A}_i\tilde{A}_i^T$ arranged in descending order of the corresponding eigenvalues, then the optimal solution for L in Eq. (3) is given by $L = [\phi_1^L, ..., \phi_r^L]$. Similarly we have the alternative expression of E^{TPCA} as follows:

$$E^{\mathrm{TPCA}} = \mathrm{tr}\left[R^T\left(\sum_{i=1}^{n} \tilde{A}_i^T\tilde{A}_i\right)R\right]. \tag{8}$$

Hence, if $\phi_1^R, ..., \phi_c^R$ are the eigenvectors of $\tilde{C}_R = \sum_{i=1}^{n} \tilde{A}_i^T\tilde{A}_i$ arranged in descending order of the corresponding eigenvalues, then the optimal solution for R in Eq. (3) is given by $R = [\phi_1^R, ..., \phi_c^R]$. The dimension of A_i is reduced by $M_i = L_{\tilde{r}}^TA_iR_{\tilde{c}}$, where $L_{\tilde{r}} = [\phi_1^L, ..., \phi_{\tilde{r}}^L]$ for $\tilde{r} \leqq r$ and

$R_{\tilde{c}} = [\phi_1^R, .., \phi_{\tilde{c}}^R]$ for $\tilde{c} \leqq c$. From Eq. (5), we have

$$E^{\mathrm{TPCA}} = \sum_{i=1}^{n} \mathrm{tr}(L^T \tilde{A}_i R R^T \tilde{A}_i^T L) = \sum_{i=1}^{n} \|L^T \tilde{A}_i R\|_F^2, \quad (9)$$

which implies that the Tensor PCA gives an approximate solution of the SLRAM because the right-most side of Eq. (9) coincides with the objective function of Eq. (2), except for the subtraction of $\bar{A}$, i.e., $\tilde{A}_i = A_i - \bar{A}$.

4.2 Two-Dimensional SVD

Let the singular value decomposition (SVD) of a matrix S be $S = U\Sigma V^T$, where Σ is a diagonal matrix whose diagonal elements are the singular values of S, and U, V are matrices whose columns are left and right singular vectors of S respectively. Since $SS^T = U\Sigma^2 U^T$ and $S^T S = V\Sigma^2 V^T$, we also have U and V from the spectral decompositions of SS^T and $S^T S$ respectively. Ding and Ye [5] extended this relationship between the singular value decomposition and the spectral decomposition of a single matrix to that of multiple matrices as follows. Define the averaged row-row and column-column covariance matrices as

$$\tilde{C}_L = \sum_{i=1}^{n} \tilde{A}_i \tilde{A}_i^T, \quad (10)$$

$$\tilde{C}_R = \sum_{i=1}^{n} \tilde{A}_i^T \tilde{A}_i, \quad (11)$$

where $\tilde{A}_i = A_i - \bar{A}$ as we stated in the previous subsection. Let $\phi_1^L, ..., \phi_{\tilde{r}}^L$ be the eigenvectors of $\tilde{C}_L$ corresponding to the largest $\tilde{r}$ eigenvalues and let $\phi_1^R, .., \phi_{\tilde{c}}^R$ be the eigenvectors of $\tilde{C}_R$ corresponding to the largest $\tilde{c}$ eigenvalues. Then the dimension of A_i is reduced by $M_i = L_{\tilde{r}}^T A_i R_{\tilde{c}}$, where $L_{\tilde{r}} = [\phi_1^L, ..., \phi_{\tilde{r}}^L]$ and $R_{\tilde{c}} = [\phi_1^R, .., \phi_{\tilde{c}}^R]$. Ding and Ye [5] called this algorithm the two-dimensional SVD (2dSVD).

4.3 Eigenimage Method

The eigenimage method presented by Zhang et al. [3] is related to the two-dimensional PCA (2DPCA) presented by Yang et al. [1], so we briefly review the 2DPCA.

4.3.1 Two-Dimensional PCA

Let $B_i = A_i R$ be the projection of A_i onto a low dimensional space. Then the optimal projection matrix R is the solution of the following problem:

$$\max_R \quad \mathrm{tr}(\tilde{C}_B)$$
$$\text{subj.to } R^T R = I_{\tilde{c}}, \quad (12)$$

where $\tilde{C}_B = \sum_{i=1}^{n} (B_i - \bar{B})(B_i - \bar{B})^T$, $\bar{B} = \sum_{i=1}^{n} B_i / n$. The matrix $\tilde{C}_B$ is called the image covariance matrix [1]. The objective function in Eq. (12) can be expressed as:

$$\begin{aligned}
\mathrm{tr}(\tilde{C}_B) &= \mathrm{tr}\left\{ \sum_{i=1}^{n} \left[\tilde{A}_i R (\tilde{A}_i R)^T \right] \right\} \\
&= \mathrm{tr}\left\{ \left[\tilde{A}_1, ..., \tilde{A}_n\right] R \left(\left[\tilde{A}_1, ..., \tilde{A}_n\right] R \right)^T \right\} \\
&= \mathrm{tr}\left(R^T \left[\tilde{A}_1, ..., \tilde{A}_n\right]^T \left[\tilde{A}_1, ..., \tilde{A}_n\right] R \right) \\
&= \mathrm{tr}(R^T \tilde{C}_R R). \quad (13)
\end{aligned}$$

Therefore, the optimal solution of Eq. (12) is given by $R_{\tilde{c}} = [\phi_1^R, ..., \phi_{\tilde{c}}^R]$, where $\phi_1^R, ..., \phi_{\tilde{c}}^R$ are the eigenvectors of $\tilde{C}_R$ corresponding to the largest $\tilde{c}$ eigenvalues.

4.3.2 Eigenimage Method

Let $\tilde{A}_{\mathrm{col}} = [\tilde{A}_1^T, ..., \tilde{A}_n^T]^T$ be a concatenation of $\tilde{A}_1, ..., \tilde{A}_n$. Then we have $\tilde{C}_R = \tilde{A}_{\mathrm{col}}^T \tilde{A}_{\mathrm{col}}$, which is equivalent to Eq. (11). The eigenimage method also uses another concatenation, denoted as $\tilde{A}_{\mathrm{row}} = [\tilde{A}_1, ..., \tilde{A}_n]$. Then we have $\tilde{C}_L = A_{\mathrm{row}} A_{\mathrm{row}}^T$, which is equivalent to Eq. (10). We compute the $\tilde{r}$ eigenvectors $\phi_1^L, ..., \phi_{\tilde{r}}^L$ of $\tilde{C}_L$ corresponding to the largest $\tilde{r}$ eigenvalues and form $L_{\tilde{r}} = [\phi_1^L, ..., \phi_{\tilde{r}}^L]$. The eigenimage is defined as

$$\Phi_{i_1 i_2} = \phi_{i_1}^L \phi_{i_2}^{R\,T} \quad (14)$$

and each A_i is approximated as

$$\hat{A}_i = \sum_{i_1=1}^{\tilde{r}} \sum_{i_2=1}^{\tilde{c}} \tilde{m}_{i_1 i_2 i} \Phi_{i_1 i_2} + \bar{A}, \quad (15)$$

where $\tilde{m}_{i_1 i_2 i} = \phi_{i_1}^{L\,T} \tilde{A}_i \phi_{i_2}^R$ is the (i_1, i_2) element of $\tilde{M}_i = [\tilde{m}_{i_1 i_2 i}] = L_{\tilde{r}}^T \tilde{A}_i R_{\tilde{c}}$.

4.4 Dyadic SVD

The dyadic SVD (DSVD) presented by Inoue and Urahama [6] is based on the higher-order SVD (HOSVD) presented by Lathauwer et al. [7]. Here we briefly review the HOSVD especially for the third-order tensors.

4.4.1 Higher-Order SVD

Let $\mathcal{A} = [a_{i_1 i_2 i_3}]$ for $i_1 = 1, ..., r;\ i_2 = 1, ..., c;\ i_3 = 1, ..., n$ be a third-order tensor comprising n matrices $A_1, ..., A_n$, i.e., $a_{i_1 i_2 i_3}$ is the (i_1, i_2) element of A_{i_3}.

[Definition 1] The 1-mode matrix unfolding $A_{(1)}$ of $\mathcal{A}$ contains the element $a_{i_1 i_2 i_3}$ at the position with row number i_1 and column number equal to

$$(i_2 - 1)n + i_3. \quad (16)$$

The 2-mode matrix unfolding $A_{(2)}$ of $\mathcal{A}$ contains the element $a_{i_1 i_2 i_3}$ at the position with row number i_2 and column number equal to

$$(i_3 - 1)r + i_1. \quad (17)$$

The 3-mode matrix unfolding $A_{(3)}$ of $\mathcal{A}$ contains the element $a_{i_1 i_2 i_3}$ at the position with row number i_3 and column number equal to

$$(i_1 - 1)c + i_2. \quad (18)$$

[Definition 2] The 1-mode product of $\mathcal{A}$ by a matrix $X = [x_{i'_1 i_1}] \in \Re^{r'} \times \Re^{r}$, denoted by $\mathcal{A} \times_1 X$, is an $(r' \times c \times n)$-tensor of which the entries are given by

$$(\mathcal{A} \times_1 X)_{i'_1 i_2 i_3} = \sum_{i_1=1}^{r} a_{i_1 i_2 i_3} x_{i'_1 i_1}. \tag{19}$$

The 2-mode product of $\mathcal{A}$ by a matrix $Y = [y_{i'_2 i_2}] \in \Re^{c'} \times \Re^{c}$, denoted by $\mathcal{A} \times_2 Y$, is an $(r \times c' \times n)$-tensor of which the entries are given by

$$(\mathcal{A} \times_2 Y)_{i_1 i'_2 i_3} = \sum_{i_2=1}^{c} a_{i_1 i_2 i_3} y_{i'_2 i_2}. \tag{20}$$

The 3-mode product of $\mathcal{A}$ by a matrix $Z = [z_{i'_3 i_3}] \in \Re^{n'} \times \Re^{n}$, denoted by $\mathcal{A} \times_3 Z$, is an $(r \times c \times n')$-tensor of which the entries are given by

$$(\mathcal{A} \times_3 Z)_{i_1 i_2 i'_3} = \sum_{i_3=1}^{n} a_{i_1 i_2 i_3} z_{i'_3 i_3}. \tag{21}$$

Equations (19), (20) and (21) can be expressed as $(\mathcal{A} \times_1 X)_{(1)} = XA_{(1)}$, $(\mathcal{A} \times_2 Y)_{(2)} = YA_{(2)}$ and $(\mathcal{A} \times_3 Z)_{(3)} = ZA_{(3)}$ respectively. The low rank approximation of a third-order tensor is formulated as follows:

$$\begin{aligned} &\min_{L,R,H,\mathcal{S}} \quad \|\mathcal{A} - \mathcal{S} \times_1 L \times_2 R \times_3 H\|_F^2 \\ &\text{subj.to } L^T L = I_{\tilde{r}},\ R^T R = I_{\tilde{c}},\ H^T H = I_{\tilde{n}}, \end{aligned} \tag{22}$$

where $L \in \Re^{r} \times \Re^{\tilde{r}}$, $R \in \Re^{c} \times \Re^{\tilde{c}}$, $H \in \Re^{n} \times \Re^{\tilde{n}}$ and $\mathcal{S} \in \Re^{\tilde{r}} \times \Re^{\tilde{c}} \times \Re^{\tilde{n}}$. The HOSVD approximately solves Eq. (22) as follows:

[HOSVD]

Step 1: Compute the $\tilde{r}$ left singular vectors $\phi_1^L, ..., \phi_{\tilde{r}}^L$ of $A_{(1)}$ corresponding to the largest $\tilde{r}$ singular values and form $L_{\tilde{r}} = [\phi_1^L, ..., \phi_{\tilde{r}}^L]$.

Step 2: Compute the $\tilde{c}$ left singular vectors $\phi_1^R, ..., \phi_{\tilde{c}}^R$ of $A_{(2)}$ corresponding to the largest $\tilde{c}$ singular values and form $R_{\tilde{c}} = [\phi_1^R, ..., \phi_{\tilde{c}}^R]$.

Step 3: Compute the $\tilde{n}$ left singular vectors $\phi_1^H, ..., \phi_{\tilde{n}}^H$ of $A_{(3)}$ corresponding to the largest $\tilde{n}$ singular values and form $H_{\tilde{n}} = [\phi_1^H, ..., \phi_{\tilde{n}}^H]$.

Step 4: Compute $\mathcal{S} = \mathcal{A} \times_1 L_{\tilde{r}}^T \times_2 R_{\tilde{c}}^T \times_3 H_{\tilde{n}}^T$.

4.4.2 Dyadic SVD

Let E^{HOSVD} be the objective function in Eq. (22). Then

$$E^{\mathrm{HOSVD}} = \|\mathcal{A} - \mathcal{M} \times_1 L \times_2 R\|_F^2 \tag{23}$$

$$= \sum_{i=1}^{n} \|A_i - LM_i R^T\|_F^2, \tag{24}$$

where $\mathcal{M} = [m_{i_1 i_2 i_3}] = \mathcal{S} \times_3 H$ and $M_i = [m_{i_1 i_2 i}]$ for $i_1 = 1, ..., \tilde{r}$; $i_2 = 1, ..., \tilde{c}$. Since Eq. (24) coincides with E, the following algorithm is derived from the HOSVD:

[DSVD]

Step 1: Compute the $\tilde{r}$ left singular vectors $\phi_1^L, ..., \phi_{\tilde{r}}^L$ of $A_{(1)}$ corresponding to the largest $\tilde{r}$ singular values and form $L_{\tilde{r}} = [\phi_1^L, ..., \phi_{\tilde{r}}^L]$.

Step 2: Compute the $\tilde{c}$ left singular vectors $\phi_1^R, ..., \phi_{\tilde{c}}^R$ of $A_{(2)}$ corresponding to the largest $\tilde{c}$ singular values and form $R_{\tilde{c}} = [\phi_1^R, ..., \phi_{\tilde{c}}^R]$.

Step 3: Compute $M_i = L_{\tilde{r}}^T A_i R_{\tilde{c}}$ for $i = 1, ..., n$.

Inoue and Urahama [6] called this algorithm the dyadic SVD (DSVD). Let $C_L = A_{(1)} A_{(1)}^T$ and $C_R = A_{(2)} A_{(2)}^T$. Then $\phi_1^L, ..., \phi_{\tilde{r}}^L$ are also the $\tilde{r}$ eigenvectors of C_L corresponding to the largest $\tilde{r}$ eigenvalues and $\phi_1^R, ..., \phi_{\tilde{c}}^R$ are also the $\tilde{c}$ eigenvectors of C_R corresponding to the largest $\tilde{c}$ eigenvalues. If $\bar{A} = 0$, then $C_L = \tilde{C}_L$ and $C_R = \tilde{C}_R$.

4.5 Equivalence of Non-Iterative Algorithms

In the rest of this paper, we assume that $\bar{A} = 0$ to simplify the notations. Although the above four algorithms, tensor PCA, 2dSVD, eigenimage method and DSVD, are formulated from different viewpoints, they are reduced to the eigenvalue problems of a row-row covariance matrix C_L and a column-column covariance matrix C_R. Therefore, they are equivalent to each other. Eventually the non-iterative algorithm (NIA) for the SLRAM is summarized as follows:

[NIA]

Step 1: Compute the $\tilde{r}$ eigenvectors $\phi_1^L, ..., \phi_{\tilde{r}}^L$ of C_L corresponding to the largest $\tilde{r}$ eigenvalues and form $L_{\tilde{r}} = [\phi_1^L, ..., \phi_{\tilde{r}}^L]$.

Step 2: Compute the $\tilde{c}$ eigenvectors $\phi_1^R, ..., \phi_{\tilde{c}}^R$ of C_R corresponding to the largest $\tilde{c}$ eigenvalues and form $R_{\tilde{c}} = [\phi_1^R, ..., \phi_{\tilde{c}}^R]$.

Step 3: Compute $M_i = L_{\tilde{r}}^T A_i R_{\tilde{c}}$ for $i = 1, ..., n$.

From the relationship between HOSVD and DSVD, we obtain an upper-bound of E as follows:

$$E \leqq \sum_{i_1=\tilde{r}+1}^{r} {\sigma_{i_1}^{(1)}}^2 + \sum_{i_2=\tilde{c}+1}^{c} {\sigma_{i_2}^{(2)}}^2 = E^{\mathrm{NIA}}, \tag{25}$$

where $\sigma_i^{(j)} = \|\mathcal{S}_{i_j=i}\|_F$ are the j-mode singular values of $\mathcal{A}$ [7].

4.6 Relationship with an Analytical Algorithm

Ding and Ye [5] considered a variant of 2dSVD as follows:

[NIA(1)]

Step 1: Compute the $\tilde{r}$ eigenvectors $\phi_1^L, ..., \phi_{\tilde{r}}^L$ of C_L corresponding to the largest $\tilde{r}$ eigenvalues and form $L_{\tilde{r}}^{\mathrm{NIA}(1)} = [\phi_1^L, ..., \phi_{\tilde{r}}^L]$.

Step 2: Compute the $\tilde{c}$ eigenvectors ${\phi'_1}^R, ..., {\phi'_{\tilde{c}}}^R$ of $C'_R = \sum_{i=1}^{n} A_i^T L_{\tilde{r}}^{\mathrm{NIA}(1)} {L_{\tilde{r}}^{\mathrm{NIA}(1)}}^T A_i$ corresponding to the largest $\tilde{c}$ eigenvalues and form $R_{\tilde{c}}^{\mathrm{NIA}(1)} = [{\phi'_1}^R, ..., {\phi'_{\tilde{c}}}^R]$.

Step 3: Compute $M_i = {L_{\tilde{r}}^{\mathrm{NIA}(1)}}^T A_i R_{\tilde{c}}^{\mathrm{NIA}(1)}$ for $i = 1, ..., n$.

This algorithm gives the following upper-bound of E:

$$E \leqq \sum_{k=\tilde{r}+1}^{r} \lambda_{Lk} + \sum_{k=\tilde{c}+1}^{c} \lambda'_{Rk} = E^{\mathrm{NIA}(1)}, \tag{26}$$

where $\lambda_{L1} \geqq \cdots \geqq \lambda_{Lr}$ are the eigenvalues of C_L and $\lambda'_{R1} \geqq \cdots \geqq \lambda'_{Rc}$ are the eigenvalues of C'_R. Alternatively, we can construct the following algorithm:

[NIA(2)]

Step 1: Compute the $\tilde{c}$ eigenvectors $\phi_1^R, ..., \phi_{\tilde{c}}^R$ of C_R corresponding to the largest $\tilde{c}$ eigenvalues and form $R_{\tilde{c}}^{\mathrm{NIA}(2)} = [\phi_1^R, ..., \phi_{\tilde{c}}^R]$.

Step 2: Compute $\tilde{r}$ eigenvectors $\phi_1'^L, ..., \phi_{\tilde{r}}'^L$ of $C'_L = \sum_{i=1}^n A_i R_{\tilde{c}}^{\mathrm{NIA}(2)^T} R_{\tilde{c}}^{\mathrm{NIA}(2)} A_i^T$ corresponding to the largest $\tilde{r}$ eigenvalues and form $L_{\tilde{r}}^{\mathrm{NIA}(2)} = [\phi_1'^L, ..., \phi_{\tilde{r}}'^L]$.

Step 3: Compute $M_i = L_{\tilde{r}}^{\mathrm{NIA}(2)^T} A_i R_{\tilde{c}}^{\mathrm{NIA}(2)}$ for $i = 1, ..., n$.

This algorithm gives the following upper-bound of E:

$$E \leqq \sum_{k=\tilde{r}+1}^{r} \lambda'_{Lk} + \sum_{k=\tilde{c}+1}^{c} \lambda_{Rk} = E^{\mathrm{NIA}(2)}, \tag{27}$$

where $\lambda_{R1} \geqq \cdots \geqq \lambda_{Rc}$ are the eigenvalues of C_R and $\lambda'_{L1} \geqq \cdots \geqq \lambda'_{Lr}$ are the eigenvalues of C'_L. Note that the NIA(2) coincides with the first iteration of the GLRAM, i.e., $L_{\tilde{r}}^{\mathrm{NIA}(2)} = L^{(1)}$ and $R_{\tilde{c}}^{\mathrm{NIA}(2)} = R^{(1)}$. The NIA(1) and NIA(2) give smaller upper-bounds than NIA, i.e., we have $\max(E^{\mathrm{NIA}(1)}, E^{\mathrm{NIA}(2)}) \leqq E^{\mathrm{NIA}}$. From Eq. (26) and (27) we have

$$E \leqq \min\left(E^{\mathrm{NIA}(1)},\ E^{\mathrm{NIA}(2)}\right) = E^{\mathrm{NIA}(3)}. \tag{28}$$

Combining the NIA(1) and NIA(2), we obtain the following algorithm which attains the minimal upper-bound $E^{\mathrm{NIA}(3)}$:

[NIA(3)]

Step 1: Compute $L_{\tilde{r}}^{\mathrm{NIA}(1)}$ and $R_{\tilde{c}}^{\mathrm{NIA}(1)}$.

Step 2: Compute $L_{\tilde{r}}^{\mathrm{NIA}(2)}$ and $R_{\tilde{c}}^{\mathrm{NIA}(2)}$.

Step 3: Compute $M_i = L_{\tilde{r}}^{\mathrm{NIA}(k^*)^T} A_i R_{\tilde{c}}^{\mathrm{NIA}(k^*)}$ for $i = 1, ..., n$, where $k^* = \arg\min_{k \in \{1,2\}}(E^{\mathrm{NIA}(k)})$.

This algorithm is essentially equivalent to an analytical algorithm presented by Liang and Shi [10]. Let

$$d_1 = \sum_{k=1}^{r} \lambda_{Lk} + \sum_{k=1}^{c} \lambda'_{Rk} - E^{\mathrm{NIA}(1)}, \tag{29}$$

$$d_2 = \sum_{k=1}^{r} \lambda'_{Lk} + \sum_{k=1}^{c} \lambda_{Rk} - E^{\mathrm{NIA}(2)}. \tag{30}$$

Then the analytical algorithm is described as follows:

[NIA(3')]

Step 1: Compute $L_{\tilde{r}}^{\mathrm{NIA}(1)}$ and $R_{\tilde{c}}^{\mathrm{NIA}(1)}$.

Step 2: Compute $L_{\tilde{r}}^{\mathrm{NIA}(2)}$ and $R_{\tilde{c}}^{\mathrm{NIA}(2)}$.

Step 3: Compute $M_i = L_{\tilde{r}}^{\mathrm{NIA}(l^*)^T} A_i R_{\tilde{c}}^{\mathrm{NIA}(l^*)}$ for $i = 1, ..., n$, where $l^* = \arg\max_{l \in \{1,2\}}(d_l)$.

Since l^* in the NIA(3') is equal to k^* in the NIA(3), the NIA(3') is equivalent to NIA(3). Although the NIA(3') is called an "analytical" algorithm, it does not necessarily give an optimal solution of the SLRAM. In Section 5, we show a counterexample.

4.7 Relationship with Non-Iterative Generalized Low Rank Approximation of Matrices

Recently, Liu and Chen [13] proposed non-iterative generalized low rank approximation of matrices (NIGLRAM) as follows:

[NIGLRAM]

Step 1: Compute the $\tilde{r}$ eigenvectors $\phi_1^L, ..., \phi_{\tilde{r}}^L$ of C_L corresponding to the largest $\tilde{r}$ eigenvalues and form $L_{\tilde{r}}^{\mathrm{NIGLRAM}} = [\phi_1^L, ..., \phi_{\tilde{r}}^L]$.

Step 2: Compute the $\tilde{c}$ eigenvectors $\phi_1''^R, ..., \phi_{\tilde{c}}''^R$ of $C''_R = \sum_{i=1}^n A_i^T L_{\tilde{r}}^{\mathrm{NIGLRAM}} L_{\tilde{r}}^{\mathrm{NIGLRAM}^T} A_i$ corresponding to the largest $\tilde{c}$ eigenvalues and form $R_{\tilde{c}}^{\mathrm{NIGLRAM}} = [\phi_1''^R, ..., \phi_{\tilde{c}}''^R]$.

Step 3: Compute $M_i = L_{\tilde{r}}^{\mathrm{NIGLRAM}^T} A_i R_{\tilde{c}}^{\mathrm{NIGLRAM}}$ for $i = 1, ..., n$.

Since $L_{\tilde{r}}^{\mathrm{NIGLRAM}} = L_{\tilde{r}}^{\mathrm{NIA}(1)}$, $C''_R = C'_R$ and $R_{\tilde{c}}^{\mathrm{NIGLRAM}} = R_{\tilde{c}}^{\mathrm{NIA}(1)}$, the NIGLRAM is equivalent to the NIA(1). Let $E^{\mathrm{NIGLRAM}} = E^{\mathrm{NIA}(1)}$. Then we have

$$E^{\mathrm{NIA}(3)} \leqq E^{\mathrm{NIGLRAM}} \leqq E^{\mathrm{NIA}}. \tag{31}$$

Liu and Chen [13] also proposed a method for determining $\tilde{r}$ and $\tilde{c}$ automatically on the basis of the normalized mean square error (NMSE).

5. Experimental Results

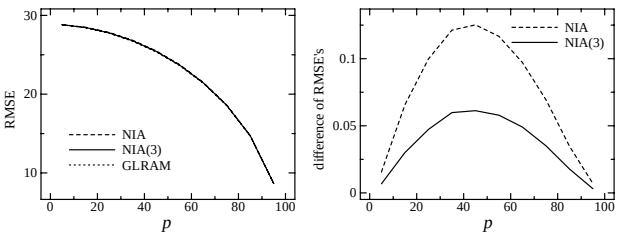
In this section, we experimentally evaluate the performance of the GLRAM, NIA and NIA(3) with respect to the capability of the SLRAM and computational efficiency. We use 500 matrices of size 100×100, i.e., $n = 500$ and $r = c = 100$. All the entries of the matrices are random numbers picked from a uniform distribution on the interval $[0, 1]$. It has been experimentally shown that the GLRAM converges much slower on these random data than on natural image data [9]. We evaluate the capability of the SLRAM with the root mean squared error (RMSE):

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \|A_i - L M_i R^T\|_F^2}. \tag{32}$$

We stop the iteration of the GLRAM when the following inequality holds:

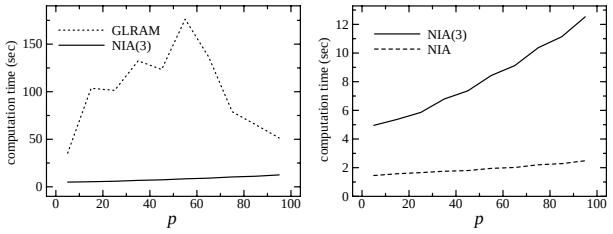
$$\frac{\mathrm{RMSE}^{(\xi-1)} - \mathrm{RMSE}^{(\xi)}}{\mathrm{RMSE}^{(\xi-1)}} < \eta, \tag{33}$$

where $\mathrm{RMSE}^{(\xi)}$ is the RMSE value at the ξth iteration and η is a threshold value. In our experiments, we choose $\eta = 10^{-6}$. We set the size of M_i as $\tilde{r} = \tilde{c} = p$ and vary p from 5 to 95. The RMSE is shown in Fig. 1(a), where the horizontal axis denotes the value of p, and the vertical axis denotes the RMSE value. The NIA, NIA(3) and GLRAM are denoted by broken, solid and dotted lines respectively. However, it



(a) (b)

Fig. 1 Comparison of the RMSE. (a) RMSEs using NIA, NIA(3) and GLRAM are denoted by broken, solid and dotted lines respectively. (b) Difference from $\mathrm{RMSE}^{\mathrm{GLRAM}}$, where NIA and NIA(3) are denoted by broken and solid lines respectively.



(a) (b)

Fig. 2 Computation time. (a) GLRAM vs. NIA(3); (b) NIA(3) vs. NIA.

is difficult to distinguish the three lines in Fig. 1(a). Let $\mathrm{RMSE}^{\mathrm{NIA}}$, $\mathrm{RMSE}^{\mathrm{NIA(3)}}$, $\mathrm{RMSE}^{\mathrm{GLRAM}}$ be the RMSE for NIA, NIA(3) and GLRAM respectively. Then the differences

$$\mathrm{RMSE}^{\mathrm{NIA}} - \mathrm{RMSE}^{\mathrm{GLRAM}}, \tag{34}$$

$$\mathrm{RMSE}^{\mathrm{NIA(3)}} - \mathrm{RMSE}^{\mathrm{GLRAM}} \tag{35}$$

are shown in Fig. 1(b). The values of Eq. (34) and (35) are denoted by broken and solid lines respectively. Figure 1(b) indicates that the following inequality holds:

$$\mathrm{RMSE}^{\mathrm{GLRAM}} < \mathrm{RMSE}^{\mathrm{NIA(3)}} < \mathrm{RMSE}^{\mathrm{NIA}}. \tag{36}$$

The former inequality shows that the NIA(3) or NIA(3') does not necessarily provide the optimal solution of the SLRAM.

We performed our experiments using Matlab with TensorToolbox [14], [15] on a Pentium IV 3.4GHz machine with 2GB RAM. Figure 2(a) shows the computation time of the GLRAM and NIA(3) denoted by dotted and solid lines respectively. The computation time of the NIA is shown in Fig. 2(b), where broken line denotes the NIA and solid line, which is identical to that in Fig. 2(a), denotes the NIA(3). Although the NIA is slightly inferior to GLRAM and NIA(3) in the ability of the SLRAM, the NIA is computationally the most efficient algorithm in the three ones.

6. Conclusion

In this paper, we have shown the equivalence of four non-iterative algorithms for the simultaneous low rank approximations of matrices (SLRAM). Those algorithms are reduced to a couple of eigenvalue problems of row-row and column-column covariance matrices. We also experimentally verified that another non-iterative algorithm presented by Liang and Shi[10] does not necessarily give the optimal solution of the SLRAM. An extension of the SLRAM to the simultaneous low rank approximations of higher-order tensors is a relevant future work.

Acknowledgment This work was partially supported by the Ministry of Education, Culture, Sports, Science and Technology, Grant-in-Aid for Young Scientists (B) in Japan, 17700197.

References

[1] J. Yang, D. Zhang, A. F. Frangi, and J. Yang, "Two-Dimensional PCA: A New Approach to Appearance-Based Face Representation and Recognition," IEEE Trans. Patt. Anal. and Mach. Intell., vol. 26, no. 1, pp. 131–137, Jan. 2004.

[2] L. Wang, X. Wang, X. Zhang, and J. Feng, "The equivalence of two-dimensional PCA to line-based PCA," Patt. Recog. Lett., vol. 26, no. 1, pp. 57–60, Jan. 2005.

[3] D. Zhang, S. Chen, and J. Liu, "Representing Image Matrices: Eigenimages Versus Eigenvectors," 2nd Int. Symp. on Neural Networks (ISNN 2005), LNCS 3497, pp. 659–664, 2005.

[4] D. Cai, X. He, and J. Han, "Subspace learning based on tensor analysis," Dept. of Computer Science Tech. Report No. 2572, Univ. of Illinois at Urbana-Champaign (UIUCDCS-R-2005-2572), May 2005.

[5] C. Ding, and J. Ye, "2-Dimensional singular value decomposition for 2D maps and images," Proc. of SIAM Int. Conf. on Data Mining (SDM 2005), Apr. 2005.

[6] K. Inoue, and K. Urahama, "DSVD: A tensor-based image compression and recognition method," IEEE Int. Symp. on Circ. and Syst. (ISCAS 2005), pp. 6308–6311, Mat 2005.

[7] Lieven De Lathauwer and Bart De Moor and Joos Vandewalle, "A multilinear singular value decomposition," SIAM J. Matrix Anal. Appl., vol. 21, no. 4, pp. 1253–1278, 2000.

[8] J. Ye, "Generalized low rank approximations of matrices," International Conference on Mach. Learning (ICML 2004), pp. 887–894, July 2004.

[9] J. Ye, "Generalized low rank approximations of matrices," Mach. Learning, vol. 61, pp. 167-191, 2005.

[10] Zhizheng Liang and Pengfei Shi, "An analytical algorithm for generalized low-rank approximations of matrices," Patt. Recog., vol. 38, no. 11, pp. 2217–2219, Nov. 2005.

[11] M. Benito, and D. Peña, "A fast approach for dimensionality reduction with image data," Patt. Recog., vol. 38, no. 12, pp. 2400–2408, Dec. 2005.

[12] P. M. Kroonenberg, Three-Mode Principal Component Analysis: Theory and Applications, DSWO Press, Leiden, The Netherlands, 1983.

[13] J. Liu, and S. Chen, "Non-iterative generalized low rank approximation of matrices," Patt. Recog. Lett., vol. 27, no. 9, pp. 1002–1008, July 2006.

[14] B. W. Bader, and T. G. Kolda, "MATLAB Tensor Classes for Fast Algorithm Prototyping," Tech. Report SAND2004-5187, Sandia National Lab., Oct. 2004.
URL: http://csmr.ca.sandia.gov/ tgkolda/TensorToolbox/

[15] B. W. Bader, and T. G. Kolda, "Algorithm xxx: MATLAB tensor classes for fast algorithm prototyping," ACM Tran. Math. Software, to appear.

Principal Curve Analysis on Manifolds

Atsushi IMIYA†, Nobuhiko YAMAGISHI††, and Tomoya SAKAI†

† Institute of Media and Information Technology, Chiba University
Yayoicho 1-33, Inage-ku, Chiba, Japan
†† School of Science and Technology, Chiba University
Yayoicho 1-33, Inage-ku, Chiba, Japan

Abstract In this paper, we develop an algorithm for the learning of the principal curve on manifolds, employing the principal curve analysis. The principal curve analysis is a generalization of principal axis analysis, which is a standard method for data analysis in pattern recognition.

Key words Manifold Learning, Princpal Curve Analysis

1. Introduction

Symbolic expressions of these point sets are required for the visual interface for the data mining systems. Furthermore, these data are sometimes transformed as a point distribution in lower dimensional vector spaces, usually tow or three dimensional spaces, for the visualisation of data distribution on CRT. Therefore, the extraction of the symbolic features of random point sets in two and three dimensional is a basic process of the visual interpretation of random point sets for the visualisation of the data space. Principal axis analysis a standard method for the data compression in Euclidean space. Some data lie on a manifold in a higher dimensional Euclidean space. Therefore, we are required to extend principal axis analysis methodology to data on manifold. On a manifold principal axis [8] becomes curve. Therfore, we develop an algorithm for the learning of the principal curves [1]〜[4] on manifolds.

Setting $\mathbf{M}$ to be a m-dimensional manifold in n-dimensional Euclidean space$\mathbf{R}^n$ for $m \leqq n-1$, we develop an algorithm for the principal-curve detection form random point set $\mathbf{P}$ on $\mathbf{M}$. We assume that our measurements are $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{\varepsilon}$ for $\boldsymbol{x} \in \mathbf{M} \subset \mathbf{R}^n$ and $\boldsymbol{\varepsilon} \in \mathbf{R}^n$ such that $|\boldsymbol{\varepsilon}| \ll 1$.

There are basically two types of problems.

（1） Compute the principal curve of $\mathbf{P}$ for known $\mathbf{M}$.

（2） Compute the principal curve of $\mathbf{P}$ with estimating $\mathbf{M}$.

In this paper, we develop an algorithm for problem 1. For the case that $\mathbf{M}$ is a collection of linear manifolds, an algorithm for the estimation of them was proposed in the reference [9].

Computational geometry provides combinatorial methods for the recovery of boundary curves as polygonal curves. These algorithms are based on Voronoi tessellation, Delaunay triangulation, Gabriel graphs, crust, α-shape, and -skeleton [3]〜[5]. The reconstructed curves, by these methods using combinatorial optimisation are piecewise linear. We develope an analytical method for the extraction of picewise liner curves using principal component analysis, which is robust against noise. Furthermore, the solutions are sensitive against noise and outlayers, since these methods construct polygons and polyhedrons using all sample points.

Section 2 is a brief survey of the principal curve analysis. In section 3, we develop an algorithm for the first problem. In section 4, we show examples for the evaluation of the algorithm when the manifold $\mathbf{M}$ is a sphere S^2 in three-dimensional Euclidean space.

2. Principal Axes and Curves

2.1 Principal Curve in Euclidean Space

Let $\mathbf{X}$ be a mean-zero point distribution in $\mathbf{R}^2$. The major principal component $\boldsymbol{w}$ maximises the criterion

$$J(\boldsymbol{w}) = E_{\boldsymbol{x}\in\mathbf{X}}|\boldsymbol{x}^\top \boldsymbol{w}|^2 \tag{1}$$

with respect to $|\boldsymbol{w}| = 1$, where $E_{\boldsymbol{x}\in\mathbf{X}}$ expresses the expectation over set $\mathbf{X}$. Line $\boldsymbol{x} = t\boldsymbol{w}$ is a one-dimensional linear subspace which approximates $\mathbf{X}$. A maximisation criterion

$$J(\boldsymbol{P}) = E_{\boldsymbol{x}\in\mathbf{X}}|\boldsymbol{P}\boldsymbol{x}|^2 \tag{2}$$

with respect to $rank\boldsymbol{P} = 1$, determines a one dimensional linear subspace which approximates $\mathbf{X}$. If $\mathbf{X}$ is not a mean-zero point distribution in $\mathbf{R}^2$ and the centroid of $\mathbf{X}$ is not predetermined, the maximisation criterion

$$J(\boldsymbol{P}, \boldsymbol{g}) = E_{\boldsymbol{x}\in\mathbf{X}}|\boldsymbol{P}(\boldsymbol{x} - \boldsymbol{g})|^2 \tag{3}$$

with respect to $rank\boldsymbol{P} = 1$, determines a one-dimensional linear manifold which approximates point distribution $\mathbf{X}$. If $\boldsymbol{g} = 0$, $\boldsymbol{P}$ is computed using PCA [8].

For the partition of $\mathbf{X}$ into $\{\mathbf{X}_i\}_{i=1}^N$ such that $\mathbf{X} = \cup_{i=1}^N \mathbf{X}_i$, vectors $\boldsymbol{g}_i$ and $\boldsymbol{w}_i$ which maximise the criterion

$$J(\boldsymbol{w}_1, \cdots, \boldsymbol{w}_N, \boldsymbol{g}_1, \cdots, \boldsymbol{g}_N) = \sum_{i=1}^{N} E_{\boldsymbol{x} \in \mathbf{X}_i} |(\boldsymbol{x} - \boldsymbol{g}_i)^\top \boldsymbol{w}_i|^2 \quad (4)$$

determine a polygonal curve [6], $\boldsymbol{l} = \boldsymbol{g}_i + t\boldsymbol{w}_i$. Furthermore, for an appropriate partition of $\mathbf{X}$ into $\{\mathbf{X}\}_{i=1}^N$, such that $\mathbf{X} = \cup_{i=1}^N \mathbf{X}_i$, vector $\boldsymbol{g}_i$ and orthogonal projector $\boldsymbol{P}_i$, which maximise the criterion

$$J(\boldsymbol{P}_1, \cdots, \boldsymbol{P}_N, \boldsymbol{g}_1, \cdots, \boldsymbol{g}_N) = \sum_{i=1}^{N} E_{\boldsymbol{x} \in \mathbf{X}_i} |\boldsymbol{P}_i(\boldsymbol{x} - \boldsymbol{g}_i)|^2 \quad (5)$$

with respect to $rank\boldsymbol{P}_i = 1$, determine a piecewise linear curve, $\mathbf{C}_i = \{\boldsymbol{x} + \boldsymbol{g}_i | \boldsymbol{P}_i \boldsymbol{x} = \boldsymbol{x}\}$. This piecewise linear is called the principal curve [6].

2.2 Curve Detection

Set $\mathbf{D}$ and $\mathbf{S}$ to be a random point set and the vertices of polygonal curve, respectively, and the distance between point $\boldsymbol{x} \in \mathbf{S}$ and $\boldsymbol{y} \in \mathbf{D}$ is defined as $d(\boldsymbol{x}, \mathbf{D}) = \min_{\boldsymbol{y} \in \mathbf{D}} d(\boldsymbol{x}, \boldsymbol{y})$ for the Euclidean distance in a plane.

The initial shapes $\mathbf{S}$ and $\mathbf{C}$ are a line segment whose direction is equivalent to the major component $\boldsymbol{w}_1$ of a random point set and a regular triangle whose vertices are determined from the principal components $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$. For a sequence of vertices $\langle \boldsymbol{v}_1, \boldsymbol{v}_2, \cdots \boldsymbol{v}_n \rangle$ of a polygonal curve, we define the tessellation as

$$\begin{aligned} V_\alpha &= \{\boldsymbol{x} | d(\boldsymbol{x}, \boldsymbol{v}_\alpha) < d(\boldsymbol{x}, \boldsymbol{v}_i), \\ &\quad d(\boldsymbol{x}, \boldsymbol{v}_\alpha) < d(\boldsymbol{x}, \boldsymbol{e}_{ij}), \alpha \neq i\}, \\ E_{\alpha\alpha+1} &= \{\boldsymbol{x} | d(\boldsymbol{x}, \boldsymbol{e}_{\alpha\alpha+1}) < d(\boldsymbol{x}, \boldsymbol{v}_i), \\ &\quad d(\boldsymbol{x}, \boldsymbol{e}_{\alpha\alpha+1}) < d(\boldsymbol{x}, \boldsymbol{e}_{ii+1}), \alpha \neq i\}, \end{aligned}$$

where $\boldsymbol{e}_{ii+1}$ is the edge which connects $\boldsymbol{v}_i$ and $\boldsymbol{v}_{i+1}$. The minimisation criterion of reference [7] is expressed as

$$I = \sum_{\boldsymbol{v}_k \in \mathbf{C}} F(\boldsymbol{v}_k, \mathbf{D}) + \sum_{\boldsymbol{v}_k \in \mathbf{P}} \sum_{i=-1}^{1} \frac{\boldsymbol{v}_{i-1i}^\top \boldsymbol{v}_{ii+1}}{|\boldsymbol{v}_{i-1i}||\boldsymbol{v}_{ii+1}|} \quad (6)$$

for

$$\begin{aligned} F(\boldsymbol{v}_k, \mathbf{D}) = &\sum_{\boldsymbol{x} \in E_{k-1k}} d(\boldsymbol{x}, \boldsymbol{v}_k) \\ &+ \sum_{\boldsymbol{x} \in V_k} d(\boldsymbol{x}, \boldsymbol{v}_k) + \sum_{\boldsymbol{x} \in E_{kk+1}} d(\boldsymbol{x}, \boldsymbol{v}_k). \end{aligned}$$

Using this criterion, we obtain an algorithm for the detection of the principal curve [7] where I_K is the value of I with K vertices.

Algorithm PCuA

(1) Set the vertices of the initial curve as $\mathbf{S}$.

(2) Move all vertices $\boldsymbol{v}_i$ $i = 1, 2, \cdots, K$, to minimise I_K.

(3) Generate the new vertex $\boldsymbol{v}_{K+1}$ on the curve $\mathbf{S}$.

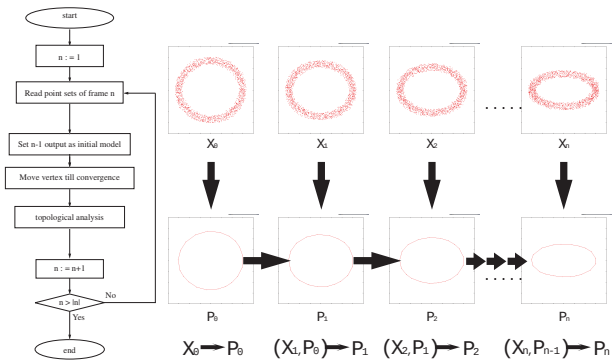
(4) If $|I_K - I_{K-1}| \leqq \varepsilon$ for a positive constant ε, then stop, else set $\mathbf{S} := \mathbf{S} \cup \{\boldsymbol{v}_{K+1}\}$ and go to 2.

This incremental algorithm preserves the topology of the initial curve, since the algorithm generates new vertices on the curve. This geometrical property leads to the conclusion that this algorithm reconstructs closed or open curves, if the initial curve is closed or open, respectively. As the initial curves for closed curves, we adopt a triangle using the direction of principal axes of point distribution.

2.3 Principal Curve of Temporal Point Set

For the data compression of time varying point sets, we are required to detect the time dependent principal curves of the sequence of time varying point sets. Setting $\mathbf{V}(t_n)$ to be the point set at time $t_n = \Delta n$ for $n = 0, 1, 2, \cdots$, the algorithm developed in the previous section permits us to compute the principal curve $\mathbf{C}(t_n)$ from point set $\mathbf{V}(t_n)$.

It is possible to assume that both principal curves $\mathbf{C}(t_n)$ and $\mathbf{C}(t_{n+1})$ exist in a finite region, we can adopt $\mathbf{C}(t_n)$ as the initial curve for the computation of $\mathbf{C}(t_{n+1})$. This variation of the algorithm saves the time for the computation of the sequence of the principal curves form a sequence of point sets. Figure 1 shows this process.



(a) (b)

Fig. 1 Detection of principal curves of temporal point set. (a) shows the flow chart of algorithm. (b) shows the generation scheme for the generation of the principal curve of the successive curve.

3. Principal Curve on Manifolds

The distance between a point and a set is defined as

$$d(\boldsymbol{x}, \mathbf{F}) = \min_{\boldsymbol{y} \in \mathbf{F}} d(\boldsymbol{x}, \boldsymbol{y}). \quad (7)$$

Furthermore, the distance between a pair of disjoint sets $\boldsymbol{F}$ and $\boldsymbol{G}$ in a space is defined as

$$d(\mathbf{F}, \mathbf{G}) = \min_{\boldsymbol{x} \in \mathbf{F}} \min_{\boldsymbol{y} \in \mathbf{G}} d(\boldsymbol{x}, \boldsymbol{y}), \quad (8)$$

for the distance $d(\boldsymbol{x}, \boldsymbol{y})$ in $\mathbf{R}^n$.

Assuming that $\boldsymbol{x} \in \mathbf{R}^2$ and $\boldsymbol{x} \in \mathbf{M}$, the projection of $\boldsymbol{x}$ onto $\mathbf{M}$ is defined as

$$\boldsymbol{P}_{\mathbf{M}}\boldsymbol{x} = \boldsymbol{y},\ d(\boldsymbol{x}, \boldsymbol{y}) = \min d(\boldsymbol{x}, \boldsymbol{M}). \tag{9}$$

If $\mathbf{M}$ is convex $\boldsymbol{y} \in \mathbf{M}$ is uniquely determined.

Using this property, we define the projection of curve c in $\mathbf{R}^n$ to $\mathbf{M}$ as following.

[Definition 1] For all point $\boldsymbol{x}$ in a curve c in $\mathbf{R}^n$

$$\boldsymbol{P}_{\mathbf{M}}c = \{\boldsymbol{y} | \boldsymbol{y} = \boldsymbol{P}_{\mathbf{M}}\boldsymbol{x}, \forall \boldsymbol{x} \in c\}. \tag{10}$$

Since, we assume that the manifold $\mathbf{M}$ is known, for the preservation of uniqueness for the projected curve, we pre-process the separation of $\mathbf{M}$ to convex and concave parts. If manifold is $\mathbf{M}$ is convex, for instance $\mathbf{M} = S^2$ in $\mathbf{R}^3$, we can skip this pre-processing.

Since we deal with a curve constructed from data points with the properties that

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{\varepsilon},\ \boldsymbol{x} \in \mathbf{M} \subset \mathbf{R}^n,\ \boldsymbol{\varepsilon} \in \mathbf{R}^n\ \ |\boldsymbol{\varepsilon}| \ll 1,$$

a principal curve $\bar{c}$ computer from $\boldsymbol{y}$ passes through a region close to the manifold $\mathbf{M}$. Therefore, the projection of $\bar{c}$, which is computed as the principal curve in $\mathbf{R}^n$, to $\mathbf{M}$ is acceptable as the principal curve on $\mathbf{M}$, if the number of vertices are appropriately large.

Algorithm PCuA on Manifold

(1) Set the vertices of the initial curve as $\mathbf{S}$.

(2) Move all vertices $\boldsymbol{v}_i$ $i = 1, 2, \cdots, K$, to minimise I_K.

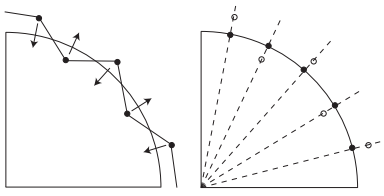
(3) Compute projections of vertices to $\mathbf{M}$.

(4) Generate the new vertex $\boldsymbol{v}_{K+1}$ on the curve $\mathbf{S}$.

(5) Compute projections of vertices to $\mathbf{M}$.

(6) If $|I_K - I_{K-1}| \leqq \varepsilon$ for a positive constant ε, then stop, else set $\mathbf{S} := \mathbf{S} \cup \{\boldsymbol{v}_{K+1}\}$ and go to 2.

The projection operation is shownb in Figure 2.



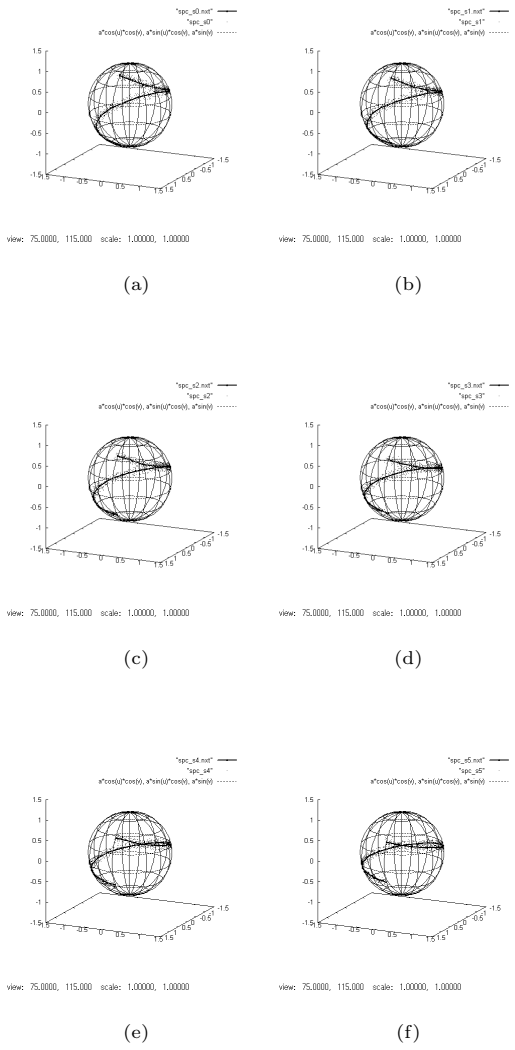
(a) (b)

Fig. 2 Projection of points on a curve to a manifold.

Figures 2 and 3 show the detection of temporal principal curve on the sphere for open and closed curves, respectively. These results show that this algorithm detects the principal curves on a manifold.

4. Conclusions

We developed an algorithm for the learning of the principal curves on manifolds, employing the principal curve analysis, which is a generalisation of principal axis analysis.



(a) (b) (c) (d) (e) (f)

Fig. 3 The temporal principal curve on the sphere. The algorithm detects the trajectry of the principal curve of temporal point sets on the sphere from (a) to (f), when the principal curves are open.

References

[1] Bookstein, F. L., The line-skeleton, CVGIP, **11**, 1233-137, 1979

[2] Rosenfeld, A., Axial representations of shapes, CVGIP, **33**, 156-173, 1986.

[3] Amenta, N., Bern, M., and Eppstein, D., The crust and the β-skeleton:Combinatorial curve reconstruction, Graphical Models and Image Processing, **60**, 125-135, 1998.

[4] Attali, D. and Montanvert, A., Computing and simplifying 2D and 3D continuous skeletons, CVIU, **67**, 261-273, 1997.

[5] Edelsbrunner, H., Shape reconstruction with Delaunay complex, Lecture Notes in Computer Science, **1380**, 119-132, 1998.

[6] Hasite, T., Stuetzle, T., Principal curves, J. Am. Statistical

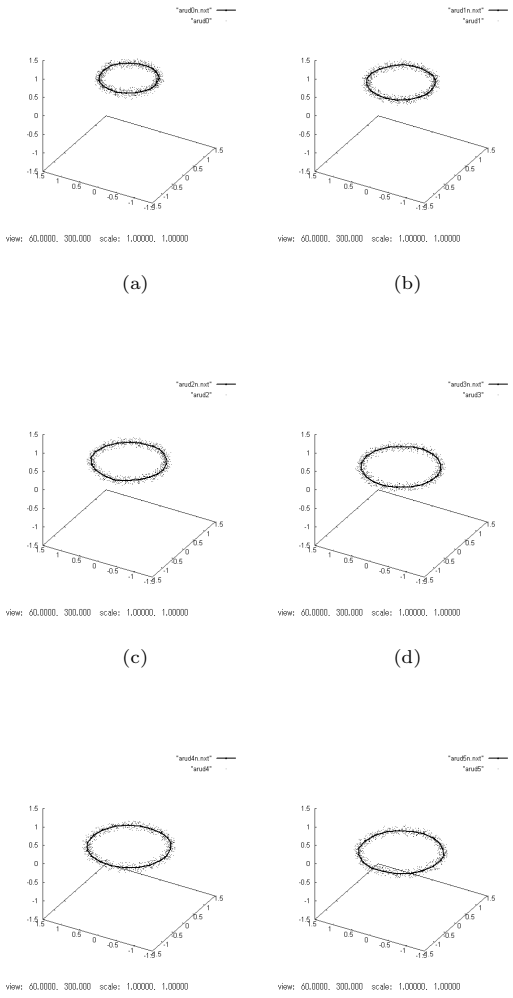


Fig. 4 The temporal principal curve on the sphere. The algorithm detects the trajectry of the principal curve of temporal point sets on the sphere from (a) to (f), when the principal curves are closed.

Assoc., **84**, 502-516, 1989.

[7] Kégl, B., Krzyzak, A., Linder, T., Zeger, K., Learning and design of principal curves, IEEE PAMI, **22**, 281-297, 2000.

[8] Oja, E., Principal components, minor components, and linear neural networks, Neural Networks, **5**, 927-935, 1992.

[9] Imiya, A., Ootani, H., Kwamoto, K., Linear manifolds analysis: theory and algorithm, Neurocomputing, **57**, 171-187, 2004.

局所部分空間法と変形不変性を用いた画像パターン分類

堀田　政二†　　喜安　千弥†　　宮原　末治†

† 長崎大学 情報システム工学科　〒 852–8154 長崎市文教町 1–14
E-mail: † hotta,kiyasu,miyahara @cis.nagasaki-u.ac.jp

あらまし　線型部分空間法の一種である CLAFIC や投影距離法は，次元削減と識別を同時に実現できる優れた手法であるが，パターンの分布が線型分離困難な場合には十分な識別率を達成することができない．そこでカーネル非線型写像を用いる方法が考えられるが，適切なカーネルとそのパラメータ，および部分空間の次元数を決定する必要があることと，データ数が増加すると必要となるメモリ量が増大するという難点がある．本稿では，未知パターンの近傍学習パターンから求めた線型多様体に基づく識別によって，線型分離困難な場合でも高い識別率を達成できる局所部分空間法 (Local Subspace Classifier, LSC) を紹介する．さらに，画像パターンの識別において変形不変性を達成するために，パターン間の距離を接距離 (Tangent Distance) で求めた後，接ベクトル (Tangent Vector) を用いて作成した近似変形パターンを識別に利用する方法を提案し，手書き数字パターンを用いた実験により識別率が向上することを示す．
キーワード　局所部分空間法，変形不変性，接距離，接ベクトル，画像パターン分類

Local Subspace Classifier with Transformation Invariance for Image Pattern Classification

Seiji HOTTA†, Senya KIYASU†, and Sueharu MIYAHARA†

† Faculty of Engineering, Nagasaki University
Bunkyomachi 1–14, Nagasaki-shi, Nagasaki, 852–8521 Japan
E-mail: † hotta,kiyasu,miyahara @cis.nagasaki-u.ac.jp

Abstract　Linear subspace methods such as CLAFIC and projection distance method have been widely used for pattern recognition in the past, because they can achieve dimension reduction and classification concurrently. However, the recognition rates of them decrease when data are not separable linearly. For overcoming this difficulty, we may separate such data by using kernel mappings but need to select adequate parameters and the type of kernels. Moreover, we require a large amount of memory for kernel matrices. This paper shows Local Subspace Classifier (LSC) proposed by Laaksonen that achieves high recognition performance even if data are not separable linearly. In addition, we propose improved LSC that selects neighbors with tangent distance and adopts transformed images obtained with tangent vectors for transformation-invariance image pattern classification. The performance of the proposed method is verified with experiments on handwritten digit pattens.
Key words　local subspace method, transformation-invariance, tangent distance, tangent vector, image pattern classification

1. はじめに

線型部分空間法の一種である CLAFIC [1] や複合類似度法 [2], 及び投影距離法 [3] は，次元削減と識別を同時に実現できる優れた手法であり多くの改良法が提案されている [4], [5]．また，部分空間法では識別部をクラス毎に設計できるため，多クラス識別を容易に実現できることも魅力の一つである．しかし，線型部分空間法は縮退したガウス分布を仮定した識別器であるため，パターンの分布が線型分離困難な場合には十分な識別率を達成することができない．そこで，線型分離が困難なパターン分布に対応できるように，カーネル非線型写像によりパターンを高次元空間に写像し，写像先の空間で線型部分空間法を適用する方法 [6], [7] が提案されている．しかし，カーネル写像を用いる手法は適切なカーネルとそのパラメータ，および部分空間の次元数を決定する必要があることと，データ数が増加するとグラム行列に必要なメモリ量が増大するという難点がある．一方，Laaksonen は未知パターンの近傍学習パターンを各クラスで求め，それらから各クラスの線型多様体 (アフィン部分空間) を作成し，未知パターンと各線型多様体との投影距離に基づき識別を行う Local Subspace Classifier (LSC) [8], [9] を提案した．

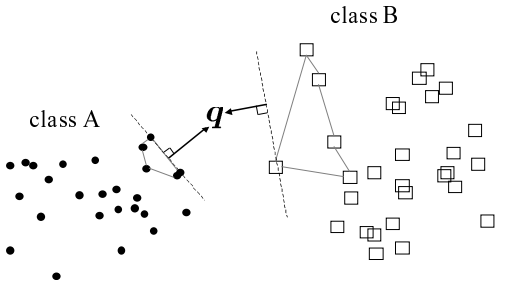


図 1　二クラス二次元パターンでの LSC の概念図．未知パターン $\boldsymbol{q}$ は各クラスの一次元線型多様体までの距離 (投影距離) が最小となるクラスに識別される．

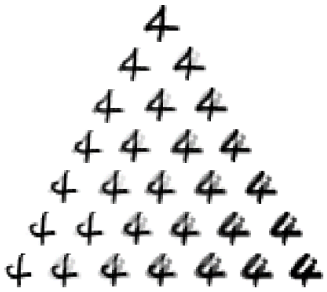


図 2　三つの手書き数字パターン “4” によって生成された二次元線型多様体．頂点が元パターンで，それ以外はそれらの線型結合パターン (係数はすべて正で合計が 1) を表す．

この方法を用いれば，識別に要する計算コストは増加するが，少ないパラメータ (近傍数のみ) で線型分離が困難なパターン分布に対しても高い識別率を達成できる．

ところで，線型部分空間法は文字認識や顔画像認識など，画像パターン分類にも適用されるが，一般に画像パターン分類において変形不変性 (Transformation Invariance) を実現することで識別率が向上することが知られている．その一例として，画像パターンに回転や位置シフト等の変形を加えることで生成される非線型な多様体を接平面で近似し，接平面間の最短距離を求めることで変形不変性を実現できる接距離 (Tangent Distance) が挙げられる [10]〜[12]．このアイデアは様々な識別法に取り入れられており，高い識別率を達成できることが報告されている [13]〜[16]．しかし，この接距離のアイデアを取り入れるためには未知パターンと学習パターンとの距離を計算する必要があるが，CLAFIC 等のすべての学習パターンから求めた部分空間を用いる方法では未知パターンと学習パターンとの距離を計算することはない．したがって，これらの方法において変形不変性を実現するためには，不変性を実現するカーネル [14]〜[16] を設計 · 使用するか特徴抽出を行う必要がある．

本稿では，識別性能の高い LSC と接距離を組合わせて用いることで，変形不変性を実現する高精度な画像パターン識別法を提案する．具体的には，未知画像パターンの近傍パターンを接距離で求めた後，接平面上の点として表される未知画像パターンと各近傍画像パターンの近似変形パターンを用いて LSC による識別を行う．この方法により，LSC と接距離を単体で用いるよりも識別率が向上することを手書き数字画像パターンを用いた実験により示す．以下，本稿は次のように構成されている．**2.** では LSC を紹介し，**3.** では接距離の原理と計算方法について紹介する．**4.** では LSC と接距離を併用した画像パターン識別法を提案する．**5.** では LSC の性質を調べる実験と，手書き数字画像パターンを用いた実験結果を示し，**6.** でまとめを述べる．

2.　局所部分空間法 (Local Subspace Classifier)

ここでは Laaksonen が提案した局所部分空間法 (Local Subspace Classifier, LSC) を紹介する．LSC では図 1 に示すように，各クラスで未知パターンに近い k 個の学習パターンを求め，各クラスの最近傍パターンが原点となるような線型多様体を作成し，その線型多様体への投影距離が最小となるクラスに未知パターンを分類する．この分類法の画像パターンに対する効果を理解するために，図 2 に三つの手書き数字の “4” の画像によって生成された二次元線型多様体を示す．図中の三角形の頂点が元の三つの手書き数字パターンであり，その内側にはそれらの線型結合パターン (係数はすべて正で合計が 1) を示す．もし未知パターンが真に属すクラスで近傍パターンが適切に選択されていれば，未知パターンに類似した線型結合パターンが線型多様体上に存在すると期待でき，反対に未知パターンが属さないクラスではそのようなパターンが生成されないと考えられるので線型多様体との距離 (投影距離) を利用すれば未知パターンを正しく分類できると期待できる．また，パターンの次元数が高い場合にはパターンの分布がスパースとなるので，高い識別率を達成するためには多くの学習パターンが必要となるが，LSC のように線型多様体との比較を行えば，学習パターンを増加させたのと同様の効果が得られ，識別率が向上すると期待できる．

2.1　LSC のアルゴリズム

ここでは LSC の具体的な識別手順を示す．クラス $j(j = 1,...,C)$ に属する N_j 個の d 次元学習パターンを $\boldsymbol{x}_i^j = [x_{i1}^j,...,x_{id}^j]^\top (i = 1,...,N_j)$ で表す．また，同じ次元数の未知パターンを $\boldsymbol{q} = [q_1,...,q_d]^\top$ とする．以下に LSC のアルゴリズムを示す．

Step 1　クラス j において，未知パターンと学習パターンとの距離をユークリッド距離

$$D(\boldsymbol{q}, \boldsymbol{x}_i^j) = \|\boldsymbol{q} - \boldsymbol{x}_i^j\| \quad (i = 1,...,N_j) \tag{1}$$

を用いて測り，距離値の小さい上位 k 個 (ここでは便宜上 $k \leqq d$ とし，各クラス共通とする) の学習パターン $\boldsymbol{x}_1^j,...,\boldsymbol{x}_k^j$ を求める．

Step 2　最近傍パターン $\boldsymbol{x}_1^j$ がクラス j の線型多様体 $\mathcal{L}_j$ の原点となるように，各近傍パターンから最近傍パターンを引いた $k-1$ 本のベクトルの集合を $\tilde{\mathbf{X}}_j = \{\boldsymbol{x}_2^j - \boldsymbol{x}_1^j,...,\boldsymbol{x}_k^j - \boldsymbol{x}_1^j\}$ とおく．

Step 3　シュミットの直交化や固有値分解を用いて $\tilde{\mathbf{X}}_j$ から正規直交ベクトルを求め，それらを並べた行列 $\mathbf{U}_j = [\boldsymbol{u}_1^j \cdots \boldsymbol{u}_{k-1}^j]$ を $k-1$ 次元の $\mathcal{L}_j$ の基底行列とする．

Step 4　未知パターン $\boldsymbol{q}$ と $\mathcal{L}_j$ との残差 $\tilde{\boldsymbol{q}}_j$ は

$$\tilde{\boldsymbol{q}}_j = \boldsymbol{q} - \boldsymbol{y}_j = \boldsymbol{q} - (\boldsymbol{x}_1^j + \hat{\boldsymbol{q}}_j) = (\mathbf{I} - \mathbf{U}_j\mathbf{U}_j^\top)(\boldsymbol{q} - \boldsymbol{x}_1^j) \tag{2}$$

と書ける (図 3 を参照)．ここで $\hat{\boldsymbol{q}}_j$ は $\boldsymbol{q}$ の $\mathcal{L}_j$ への正射影であり

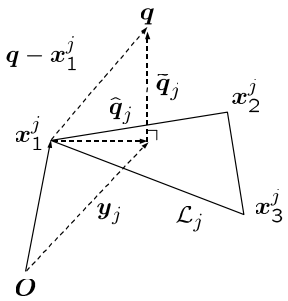


図 3　$k=3$ の場合の LSC の概念図．未知パターン $\boldsymbol{q}$ と線型多様体 $\mathcal{L}_j$ との距離は $\tilde{\boldsymbol{q}}_j$ の長さである投影距離で測られる．

$$\hat{\boldsymbol{q}}_j = \mathbf{U}_j\mathbf{U}_j^\top(\boldsymbol{q}-\boldsymbol{x}_1^j) \tag{3}$$

である．したがって，未知パターン $\boldsymbol{q}$ の投影距離の二乗は

$$\begin{aligned}\|\tilde{\boldsymbol{q}}_j\|^2 &= \|(\mathbf{I}-\mathbf{U}_j\mathbf{U}_j^\top)(\boldsymbol{q}-\boldsymbol{x}_1^j)\|^2 \\ &= \|\boldsymbol{q}-\boldsymbol{x}_1^j\|^2 - \|\mathbf{U}_j^\top(\boldsymbol{q}-\boldsymbol{x}_1^j)\|^2\end{aligned} \tag{4}$$

となる．これをすべてのクラスで計算し，値が最小となるクラスに未知パターンを分類する．すなわち，未知パターンのクラスを ω とすると識別規則は以下の様に書ける．

$$\omega = \arg\min_{j=1\ \cdots\ C} \|\tilde{\boldsymbol{q}}_j\|^2 \tag{5}$$

ちなみに，式 (4) から投影距離は最近傍パターンとのユークリッド距離より小さくなることが分かる．また，$k=1$ の場合は $\tilde{\mathbf{X}}_j$ が空集合となり，$\|\tilde{\boldsymbol{q}}_j\|^2 = \|\boldsymbol{q}-\boldsymbol{x}_1^j\|^2$ となるので LSC は最近傍決定則と等価になることが分かる．

2.2　凸射影を取り入れた LSC

Laaksonen は上記の LSC において，近傍学習パターンを頂点とする凸領域の外側で未知パターンとクラスとの距離が評価されないように，LSC に凸射影を取り入れた LSC+ という識別法も提案している [8], [9]．図 4 に LSC と LSC+ の違いを簡単な例を用いて示す．この例において，LSC では未知パターン $\boldsymbol{q}$ はクラス A と識別されるが，LSC+ ではクラス A において凸領域の外側で距離が評価されるため，凸領域で最も距離の小さい最近傍パターンとの距離が用いられる．したがって，未知パターンは距離値の小さいクラス B に識別される．この処理により，例えば学習パターンまでの距離値が大きいのにも関わらず，それらから定義される線型多様体との投影距離が小さいために誤識別されるといった不都合が解消できると期待できる．

LSC+ の具体的な識別手順を述べる前に，準備としてクラス j における k 個の近傍学習パターンを並べた行列を $\mathbf{X}_j = [\boldsymbol{x}_1^j \cdots \boldsymbol{x}_k^j]$ とし，各学習パターンの係数を並べたベクトルを $\boldsymbol{c}_j = [c_1^j, ..., c_k^j]^\top$ とおく．すると未知パターン $\boldsymbol{q}$ との距離が最小となる射影ベクトル $\boldsymbol{y}_j$ (図 3 参照) は以下のように書くことができる:

$$\boldsymbol{y}_j = \boldsymbol{x}_1^j + \hat{\boldsymbol{q}}_j = \sum_{i=1}^{k} c_i^j \boldsymbol{x}_i^j = \mathbf{X}_j\boldsymbol{c}_j, \quad \sum_{i=1}^{k} c_i^j = 1 \tag{6}$$

ここで式 (6) を $\boldsymbol{c}_j$ について解くことを考えると，擬似逆行列または特異値分解 $\mathbf{X}_j = \mathbf{L}_j \ {}_j\mathbf{R}_j^\top$ を利用して

$$\boldsymbol{c}_j = \mathbf{X}_j^\dagger\boldsymbol{y}_j = (\mathbf{X}_j^\top\mathbf{X}_j)^{-1}\mathbf{X}_j^\top\boldsymbol{y}_j = \mathbf{R}_j \ {}_j^\dagger\mathbf{L}_j^\top\boldsymbol{y}_j \tag{7}$$

と解くことができる．ここで，$\mathbf{L}_j$ は左特異ベクトルが並んだ $d\times k$ の行列，${}_j$ は最初の $k\times k$ の行列部の対角要素に特異値が並んだ $d\times k$ の行列，$\mathbf{R}_j$ は右特異ベクトルが並んだ $k\times k$ の行列である．また，${}_j^\dagger$ は ${}_j$ の最初の $k\times k$ の行列部の対角要素の逆数を要素とした行列 ${}_j^{-1}$ を転置したものを表す．

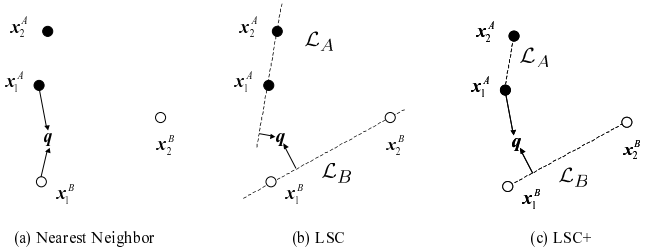


図 4　最近傍決定則と LSC 及び LSC+ の概念図.

LSC+ では射影ベクトル $\boldsymbol{y}_j$ が近傍学習パターンを頂点とする凸領域内に含まれるかどうかを判定する必要があるが，Laaksonen は式 (7) で求めた $\boldsymbol{c}_j$ を用いる判定方法を提案している [8], [9]．具体的には，まず $\boldsymbol{c}_j$ の要素の最小値を調べ，その値が正の場合には $\boldsymbol{y}_j$ はすべての近傍学習パターンの正の係数での線型結合で表される，すなわち凸領域に含まれていると判断できるので，線型多様体との投影距離を未知パターンとクラス j との距離とする．一方，$\boldsymbol{c}_j$ の最小値が負の場合には $\boldsymbol{y}_j$ は凸領域の外に存在していると判断できるので，係数が最小値になった学習パターンを $\mathbf{X}_j$ から取り除く．その後，再び $\boldsymbol{c}_j$ を計算してその要素の最小値を調べる．そして，その値が正になるまで学習パターンの除去と $\boldsymbol{c}_j$ の再計算を繰り返す．ただし，$\mathbf{X}_j$ に含まれる学習パターンの数が一つになった場合は，最近傍学習パターンとの距離値をクラス j との距離とする．このようにすれば，未知パターンとクラスとの距離を凸領域との距離で測ることができる．LSC+ の + の記号の意味は，近傍学習パターンの係数がすべて正である場合の投影距離を用いることに由来している．

3.　接距離による変形不変な画像パターンの識別

ここでは Simard らが提案したモノクロ画像間の距離尺度の一種である接距離 (Tangent Distance, TD) [10], [11] について概説する．パターン認識の分野で多く用いられる k 近傍決定則 (k-nearest neighbor, kNN) や Radial Basis Function 等の識別器では，パターン間の距離としてマンハッタン距離やユークリッド距離がよく用いられるが，これらの距離は画像などの変形に敏感であるという難点がある．例として，図 5 に未知パターンの例である “9” の画像と，数字の 9 と 4 にそれぞれ属す学習パターンの例を示す．これらの学習パターンとのユークリッド距離 (対応する画素の二乗誤差の総和) に基づく最近傍決定則を行うと，数字 4 の学習パターンとの距離の方が小さいため未知パターンは誤識別となる．そこで，もしこれらの画像パターンに対して，事前に水平・垂直移動，スケール変換などの変形を施

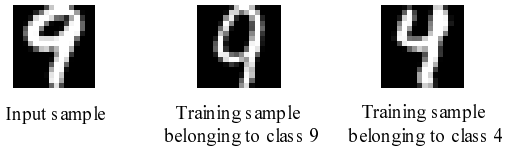


図 5　数字 9 の未知パターンと，数字 9 と数字 4 の学習パターン例.

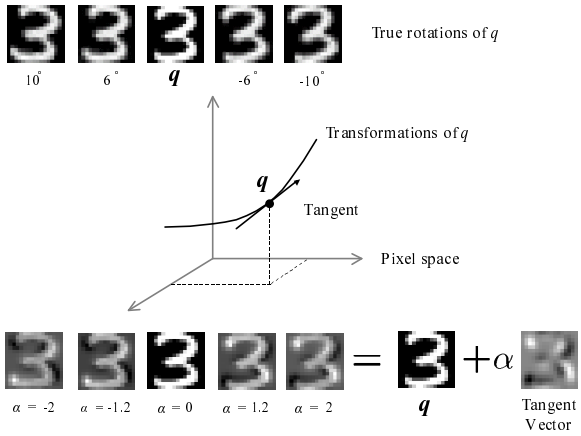


図 6　回転を加えた場合の画像パターンの描く軌跡と，接ベクトルによる回転パターンの線型近似の概念図.

すことができれば，未知パターンと数字 9 に属す学習パターンとの距離の方が小さくなると考えられる．しかし，複数の変形を施しながらパターン間の最短距離を探索するための計算コストは組合わせの数が多すぎるため莫大なものとなってしまう.

3.1　非線型多様体の線型近似

上記の問題に対し，Simard らはパターンに変形を施した場合に描くパターン空間上の非線型な多様体をパターン点における接平面で線型近似し，接平面間の最短距離である接距離をパターン間の距離として kNN で識別を行う方法を提案した [10], [11]．図 6 の上段と中段に回転を加えた手書き数字パターンと，それらのパターンがパターン空間で描く軌跡の概念図を示す．ある画像パターン $\boldsymbol{q}$ が与えられ，その画像にガウス関数による平滑化処理を適用した後，パラメータ α に依存する一つの変形 s (図 6 では s が回転で α が回転角) を施すと，変形パターン $s(\alpha, \boldsymbol{q})$ はパターン空間上で一次元の連続な曲線 (非線型多様体) を描くが，この非線型多様体は点 $\boldsymbol{q}$ における接ベクトル (Tangent Vector, TV) の張る接平面を用いて線型近似することができる．図 6 の下段には回転に対する TV とさまざまな α に対する近似変形パターンを示す．図 6 の上段に示した実際のパターンに回転を施したものと類似したパターンが得られているのが分かる．なお，r 個の変形 (回転，平行移動，スケール変換等) を考えた場合，非線型多様体は r 次元多様体となり，その線型近似はそれぞれの変形に対応する r 個の TV と，元パターンとの線型結合によって実現できる.

3.2　パターン間の接距離

ここで二つの画像パターン $\boldsymbol{q}$ と $\boldsymbol{x}$ の距離について考える．先に述べた通り単純なユークリッド距離は変形に敏感であるが，各パターンに対応する非線型多様体 S_q と S_x との最短距離 (図 7 を参照) を二つのパターン間の距離とすれば，変形不変

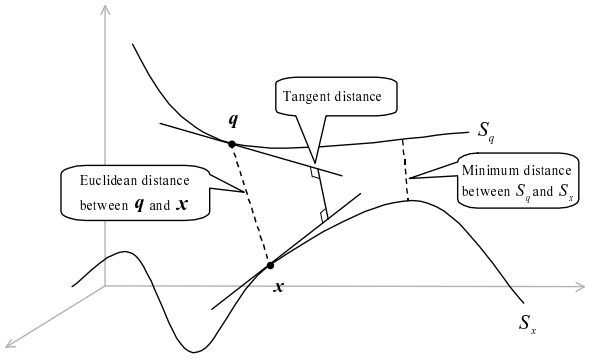


図 7　パターン $\boldsymbol{q}$ と $\boldsymbol{x}$ のユークリッド距離と接距離の概念図.

という意味で適切な距離尺度となりうる．しかし，非線型な多様体間の最短距離を見つける問題は局所解を多く持つ最適化問題でないと定式化できない．そこで Simard らは S_q と S_x との最短距離を計算する替わりに，パターン点 $\boldsymbol{q}$ と $\boldsymbol{x}$ での接平面間の最短距離である接距離を計算する方法を提案した．図 7 に多次元空間での $\boldsymbol{q}$ と $\boldsymbol{x}$ の接距離の概念図を示す (変形の種類は 1 種類)．図のように各変形パターンは $\boldsymbol{q}$ と $\boldsymbol{x}$ を通る一次元の曲線で表され，点 $\boldsymbol{q}$ と $\boldsymbol{x}$ での接平面が直線で表されている (これらの直線は三次元以上の多次元空間で交差することはない)．Simard らは手書き数字パターンを用いた kNN による識別実験を行い識別率が向上することを示している [10], [11].

3.3　接距離の計算方法

ここではパターン間の接距離の具体的な計算方法について述べる．パターン空間におけるパターンの r 個の変形を，元パターン $\boldsymbol{x}$ とパラメータ $\boldsymbol{\alpha} = [\alpha_1, ..., \alpha_r]^\top$ を用いて微分可能な関数 $s(\boldsymbol{\alpha}, \boldsymbol{x})$ で表す．変形が与えられていない場合 ($\boldsymbol{\alpha} = \boldsymbol{0}$) には $s(\boldsymbol{0}, \boldsymbol{x}) = \boldsymbol{x}$ となる．s は微分可能なため，変形パターンの描く非線型な多様体は微分可能となる．この多様体を線型近似する接平面 H_x を張る TV は以下の行列の各列ベクトルとして与えられる (具体的な TV の求め方は後述)：

$$\mathbf{T}_x = \left.\frac{\partial s(\boldsymbol{\alpha}, \boldsymbol{x})}{\partial \boldsymbol{\alpha}}\right|_{\boldsymbol{\alpha}=\boldsymbol{0}} = \left.\left[\frac{\partial s(\boldsymbol{\alpha}, \boldsymbol{x})}{\partial \alpha_1} \cdots \frac{\partial s(\boldsymbol{\alpha}, \boldsymbol{x})}{\partial \alpha_r}\right]\right|_{\boldsymbol{\alpha}=\boldsymbol{0}} \tag{8}$$

ここで二つのパターン $\boldsymbol{q}$ と $\boldsymbol{x}$ の TV をそれぞれ $\mathbf{T}_q$，$\mathbf{T}_x$ とすると，二つの接平面間の最短距離である両側接距離 $D_T(\boldsymbol{q}, \boldsymbol{x})$ の二乗は

$$D_T^2(\boldsymbol{q}, \boldsymbol{x}) = \min_{\tilde{\boldsymbol{q}} \in H_q} \min_{\tilde{\boldsymbol{x}} \in H_x} \|\tilde{\boldsymbol{q}} - \tilde{\boldsymbol{x}}\|^2 \tag{9}$$

と書ける．ここで接平面 H_q と H_x 上のパターン $\tilde{\boldsymbol{q}}$ と $\tilde{\boldsymbol{x}}$ は，それぞれの TV とパラメータベクトルを用いて

$$\tilde{\boldsymbol{q}} = \boldsymbol{q} + \mathbf{T}_q \boldsymbol{\alpha}_q \tag{10}$$

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} + \mathbf{T}_x \boldsymbol{\alpha}_x \tag{11}$$

と書けるので式 (9) は

$$D_T^2(\boldsymbol{q}, \boldsymbol{x}) = \min_{\boldsymbol{\alpha}_q} \min_{\boldsymbol{\alpha}_x} \|\tilde{\boldsymbol{q}} - \tilde{\boldsymbol{x}}\|^2 \tag{12}$$

と書き直すことができる．この距離を最小にする $\boldsymbol{\alpha}_q$ と $\boldsymbol{\alpha}_x$ は

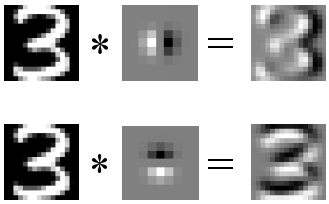


図 8　水平 (上段) と垂直 (下段) 方向の移動に対する TV の例．左が元画像，中央が $\partial g_\sigma/\partial x$ 及び $\partial g_\sigma/\partial y$，右が得られた TV．

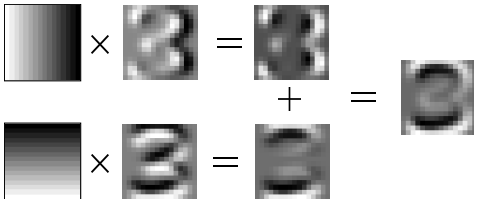


図 9　左: x 方向と y 方向の変位を表した二次元パターン $D_x(x,y)$ と $D_y(x,y)$．これらのパターンと $I*\partial g_\sigma/\partial x$ 及び $I*\partial g_\sigma/\partial y$ (二列目) の対応する画素を掛けたもの (三列目) を足すことで拡大縮小に対する TV (右) が得られる．

以下の条件を満たす:

$$\frac{\partial D_T^2(\boldsymbol{q},\boldsymbol{x})}{\partial \boldsymbol{\alpha}_q} = 2(\tilde{\boldsymbol{q}}-\tilde{\boldsymbol{x}})^\top \mathbf{T}_q = \mathbf{0} \tag{13}$$

$$\frac{\partial D_T^2(\boldsymbol{q},\boldsymbol{x})}{\partial \boldsymbol{\alpha}_x} = 2(\tilde{\boldsymbol{q}}-\tilde{\boldsymbol{x}})^\top \mathbf{T}_x = \mathbf{0} \tag{14}$$

したがって，上式に式 (10) 及び (11) を代入して $\boldsymbol{\alpha}$ について解けば

$$\boldsymbol{\alpha}_q = (\mathbf{T}_{qx}\mathbf{T}_{xx}^{-1}\mathbf{T}_{xq} - \mathbf{T}_{qq})^{-1}(\mathbf{T}_{qx}\mathbf{T}_{xx}^{-1}\mathbf{T}_x^\top - \mathbf{T}_q^\top)(\boldsymbol{x}-\boldsymbol{q}) \tag{15}$$

$$\boldsymbol{\alpha}_x = (\mathbf{T}_{xx} - \mathbf{T}_{xq}\mathbf{T}_{qq}^{-1}\mathbf{T}_{qx})^{-1}(\mathbf{T}_{xq}\mathbf{T}_{qq}^{-1}\mathbf{T}_q^\top - \mathbf{T}_x^\top)(\boldsymbol{x}-\boldsymbol{q}) \tag{16}$$

となる．ここで $\mathbf{T}_{qq}=\mathbf{T}_q^\top\mathbf{T}_q$，$\mathbf{T}_{xq}=\mathbf{T}_x^\top\mathbf{T}_q$，$\mathbf{T}_{qx}=\mathbf{T}_q^\top\mathbf{T}_x$，$\mathbf{T}_{xx}=\mathbf{T}_x^\top\mathbf{T}_x$ である．接距離 D_T は式 (15) と (16) で求めた $\boldsymbol{\alpha}$ を用いて式 (12) で計算すればよい．なお，本稿では計算時間の短縮のために文献 [10] にならって，両側接距離が小さい k 個の学習パターンを求める場合には次の様にした: まず，ユークリッド距離の小さい上位 K 個の学習パターン ($K=100$ 程度で良い) を求め，その中から両側接距離の小さい上位 k 個を選んだ．

ちなみに，未知パターンを固定して，学習パターン点における接平面との最短距離を計算する片側接距離では

$$D_T^2(\boldsymbol{q},\boldsymbol{x}) = \min_{\boldsymbol{\alpha}_x} \|\boldsymbol{q}-\tilde{\boldsymbol{x}}\|^2 \tag{17}$$

を最小化すればよく，その場合の最適解は

$$\boldsymbol{\alpha}_x = (\mathbf{T}_x^\top\mathbf{T}_x)^{-1}\mathbf{T}_x^\top(\boldsymbol{q}-\boldsymbol{x}) \tag{18}$$

で与えられる．文献 [12] では両側接距離は片側接距離と比べて識別精度の向上はわずかであると述べているが，実際には大幅に精度が向上することもあるので特別な事情がない場合には両側接距離を試みる価値はあるといえる．

3.4　TV の計算方法

ここでは，本稿の実験で用いた七つの TV である，水平 · 垂直移動，回転，拡大 · 縮小，水平方向への伸縮変形，ねじれ，細め · 太めに対する TV の実際の求め方を概説する (詳細は [11] を参照されたい)．d 個の離散的な画素値からなる画像を I とする．まず，TV を求めるためには I が微分可能でなくてはならない．そこで分散が σ であるガウスフィルタ

$$g_\sigma(x,y) = \exp\left[-\frac{x^2+y^2}{2\sigma^2}\right] \tag{19}$$

との畳み込み $I'=I*g_\sigma$ を行えば I' は微分可能となり，この性質を利用すれば TV を求めることができるが，以下に示すように実際の計算では $\partial g_\sigma/\partial x$ と $\partial g_\sigma/\partial y$ を用いる．また，ガウスフィルタの σ を適切に設定することで，誤差の大きい近似を回避することができる．

- **水平移動に対する TV**：ここで x 方向 (水平方向) の移動を表す作用素 $T_X=\frac{\partial}{\partial x}$ を I' に適用すると

$$T_X(I') = \frac{\partial}{\partial x}(I*g_\sigma) = I*\frac{\partial g_\sigma}{\partial x} \tag{20}$$

となる．すなわち，元の画像 I と g_σ を x で偏微分したフィルタとの畳み込みを計算すればよく，その結果として得られるパターンが横方向の移動を近似するための TV となる．

- **垂直移動に対する TV**：y 方向 (垂直方向) の移動を表す作用素 $T_Y=\frac{\partial}{\partial y}$ を I' に適用すると

$$T_Y(I') = \frac{\partial}{\partial y}(I*g_\sigma) = I*\frac{\partial g_\sigma}{\partial y} \tag{21}$$

となる．すなわち，元の画像 I と g_σ を y で偏微分したフィルタとの畳み込みを計算すればよく，その結果として得られるパターンが垂直方向の移動を近似するための TV となる．これらの演算例を図 8 に示す．

- **拡大 · 縮小に対する TV**：拡大 · 縮小を表す作用素 $T_S=x\frac{\partial}{\partial x}+y\frac{\partial}{\partial y}$ を I' に適用すると

$$T_S(I') = \left[x\frac{\partial}{\partial x}+y\frac{\partial}{\partial y}\right](I*g_\sigma) = \left[I*\frac{\partial g_\sigma}{\partial x}\right]+\left[I*\frac{\partial g_\sigma}{\partial y}\right] \tag{22}$$

となる．ここで x と y は水平と垂直方向の変位を表しており，実際の計算では元の画像 I と同じサイズの二次元パターン $D_x(x,y)=x-x_0$，$D_y(x,y)=y-y_0$ との要素毎の積を計算することになる．ただし (x_0,y_0) はパターンの中心座標を表す．この演算例を図 9 に示す．

- **回転に対する TV**：回転を表す作用素 $T_R=y\frac{\partial}{\partial x}-x\frac{\partial}{\partial y}$ を I' に適用した

$$T_R(I') = \left[I*\frac{\partial g_\sigma}{\partial x}\right]-\left[I*\frac{\partial g_\sigma}{\partial y}\right] \tag{23}$$

が回転に対する TV となる．

- **水平方向への伸縮変形に対する TV**：水平方向への伸縮変形を表す作用素 $T_P=x\frac{\partial}{\partial x}-y\frac{\partial}{\partial y}$ を I' に適用した

$$T_P(I') = \left[I*\frac{\partial g_\sigma}{\partial x}\right]-\left[I*\frac{\partial g_\sigma}{\partial y}\right] \tag{24}$$

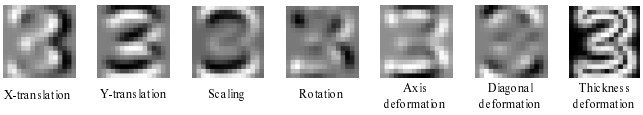


図 10　本稿の実験で用いた七つの TV の例.

が水平方向への伸縮変形に対する TV となる.

- **ねじれに対する TV**:ねじれを表す作用素 $T_D = y\frac{\partial}{\partial x} + x\frac{\partial}{\partial y}$ を I' に適用した

$$T_D(I') = y\left(I * \frac{\partial g_\sigma}{\partial x}\right) + x\left(I * \frac{\partial g_\sigma}{\partial y}\right) \tag{25}$$

がねじれに対する TV となる.

- **細め · 太めに対する TV**:細め · 太めに対する TV は画像の各画素の標準化された勾配によって与えられる [11]. 本稿ではまず以下のパターン

$$T_T(I') = \sqrt{\left(I * \frac{\partial g_\sigma}{\partial x}\right)^2 + \left(I * \frac{\partial g_\sigma}{\partial y}\right)^2} \tag{26}$$

を求めた. 次に $T_T(I')$ をベクトルと見なしたときのノルム $\|T_T(I')\|$ を求め, $T_T(I')$ の各画素をノルムで割ったものを細め · 太めの TV とした.

図 10 には上記の方法で求めた七つの TV の例を示す.

4. LSC と接距離を利用した画像パターン分類

ここでは LSC と接距離を組合わせて変形にロバストな画像パターン分類法を提案する. 通常, パターン間の距離は互いのパターンが可能な限り似るように変換されるまでは計算しない方がよい [12]. しかし, LSC では無批判にユークリッド距離を用いており, 本質的に類似していないパターンを用いて識別を行っている可能性がある. 一方, 接距離を用いた識別では kNN がよく用いられるが, kNN は次元数が大きくなると識別性能が低下することが知られている [17]. この原因の一つとして, 次元が高くなるとパターン空間中でのパターンの分布がスパースになるため, 学習パターン数が十分でないと誤識別を起こし易くなるためと考えられる.

そこで本稿では, LSC と接距離での互いの難点を克服するために, 両手法を併用する方法について考える. 具体的には, まず, 未知画像パターンの k 近傍学習パターンを各クラスで求める際, ユークリッド距離の替わりに両側接距離を用いて, 本質的に類似していないパターンを識別時に排除する. 次に, 両側接距離を計算する際に求めた未知パターンと近傍学習パターンのそれぞれ k 個の近似変形パターンを用いて LSC による識別を行う. ただし, LSC では未知パターンはパターン空間上の一点として表される必要があるので, クラス毎に k 個の近似変形未知パターンの平均パターン (これらはすべて同一接平面上に存在している) を求め, それらを各クラスでの未知パターンと見なす. クラス毎に平均化するのは, 両側接距離で求めた各クラスの変形近似未知パターンはクラス特有の変形を受けているためである. 一方, 両側接距離ではなく未知パターンを固定した片側接距離を用いることも考えられるが, 両側接距離を用いた方がエラー率が低くなることを後の実験で示す.

4.1 提案手法のアルゴリズム

ここでは提案する画像パターンの識別法の具体的な手順を示す. ここで用いる記号は **2.** の LSC の説明で用いたものと同じである.

Step 1　クラス j において, 未知パターンと学習パターンとの距離を式 (12) の両側接距離 D_T で計算し, 値が小さい上位 k 個の学習パターン $\boldsymbol{x}_1^j, ..., \boldsymbol{x}_k^j$ を求める. このときクラス j の第 i 近傍 $(i = 1, ..., k)$ を求める際に計算した接平面上の近似変形未知パターン

$$\tilde{\boldsymbol{q}}_i^j = \boldsymbol{q} + \mathbf{T}_q \boldsymbol{\alpha}_i^j \quad (i = 1, ..., k) \tag{27}$$

と近似変形学習パターン

$$\tilde{\boldsymbol{x}}_i^j = \boldsymbol{x}_i^j + \mathbf{T}_{x_i}^j \boldsymbol{\alpha}_{x_i}^j \quad (i = 1, ..., k) \tag{28}$$

を保存しておく. ここで $\boldsymbol{\alpha}_i^j$ は未知パターンとクラス j の第 i 近傍学習パターンとの両側接距離を計算するために求めた未知パターンの係数ベクトルであり, $\mathbf{T}_{x_i}^j$ と $\boldsymbol{\alpha}_{x_i}^j$ は $\boldsymbol{x}_i^j$ の TV とその係数ベクトルである.

Step 2　最近傍近似変形学習パターン $\tilde{\boldsymbol{x}}_1^j$ がクラス j の線型多様体 $\mathcal{L}_j$ の原点となるように, 残りの $k-1$ 本の $\tilde{\boldsymbol{x}}_2^j, ..., \tilde{\boldsymbol{x}}_k^j$ から $\tilde{\boldsymbol{x}}_1^j$ を引いたベクトルを並べた行列を $\tilde{\mathbf{X}}_j = [\tilde{\boldsymbol{x}}_2^j - \tilde{\boldsymbol{x}}_1^j \cdots \tilde{\boldsymbol{x}}_k^j - \tilde{\boldsymbol{x}}_1^j]$ とおく. また, $\tilde{\boldsymbol{q}}_i^j$ の平均パターン

$$\bar{\boldsymbol{q}}^j = \frac{1}{k}\sum_{i=1}^{k} \tilde{\boldsymbol{q}}_i^j = \boldsymbol{q} + \mathbf{T}_q \frac{1}{k}\sum_{i=1}^{k} \boldsymbol{\alpha}_i^j \tag{29}$$

を求めておく.

Step 3　**Step 2** で求めた $\tilde{\mathbf{X}}_j$ を線型多様体を作成するためのパターン行列, $\bar{\boldsymbol{q}}^j$ をクラス j における未知パターンと見なして **2.** で示した LSC の **Step 3** 以降を実行し, 最も投影距離が小さいクラスを未知パターンの属すクラスとして出力する.

5. 実　　験

ここでは, はじめに人工パターンを用いて LSC の識別境界を観察し, LSC が線型分離困難な場合でもパターンを分類できることを確認する. 次に, 手書き数字パターンを用いた識別実験を行い, 提案手法の有効性を検証する.

5.1 LSC の識別境界

まず, LSC の識別境界を線型分離困難な二クラス二次元パターンを用いて求め, 投影距離法 (Projection Distance Method, PDM) 及びカーネル非線型投影距離法 (Kernel PDM, KPDM) と比較してみる. 実験に用いた二次元データの生成方法は文献 [7] を参考にした. すなわち, x_1, y_1, x_2, y_2 を平均と分散 $(\ , \sigma^2)$ がそれぞれ $(0, 10), (10, 5), (3, 10), (20, 5)$ の正規分布からランダムにサンプリングされた値とし, クラス ω_1 とクラス ω_2 に属する二次元パターンをそれぞれ $(x_1, 0.5x_1^2 + y_1) \in \omega_1$, $(x_2, -0.5x_2^2 + y_2) \in \omega_2$ としてパターンを生成した.

図 11 に学習パターン数が 300 の場合の PDM, KPDM, LSC 及び LSC+ によって得られる識別境界の一例を示す. 二クラスのパターンを点の種類で区別し, 識別境界を線で示した. 図から, PDM では識別境界面が直線となり, クラスを上手く分離

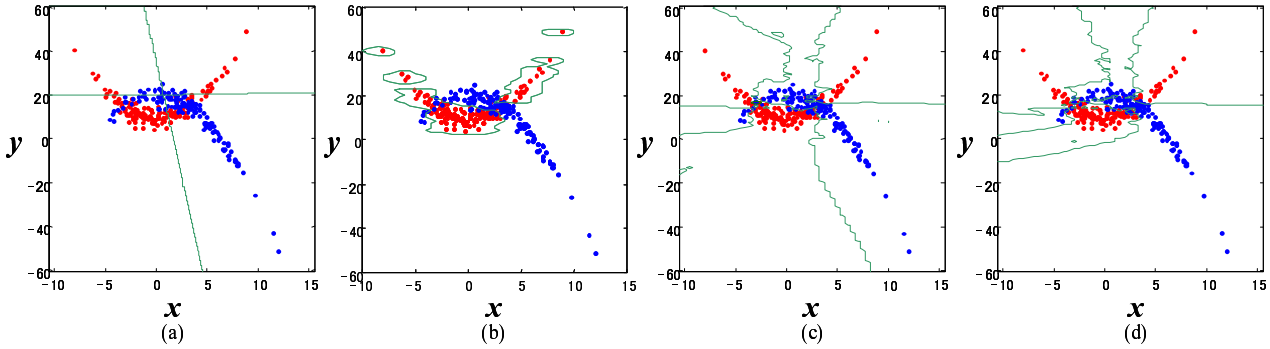


図 11　識別境界の一例．(a)：PDM．(b)：KPDM．(c)：LSC. (d) LSC+.

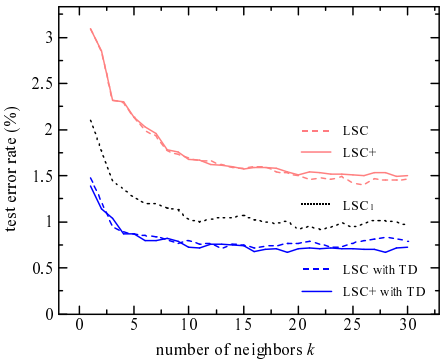


図 12　MNIST における k とエラー率の関係.

表 1　MNIST におけるテストパターンのエラー率

method	test error rate [%]
Nearest neighbor with Euclidean Distance	3.09
Nearest neighbor with TD ($\sigma = 0.9$)	1.49
PDM ($d_l = 26$)	4.1
LSC ($k = 26$)	1.4
LSC+ ($k = 29$)	1.49
Extended TD [13]	1.0
LeNet5 [20]	0.9
Virtual polynominal SVM [14]	0.8
LSC with TD ($k = 13, \sigma = 0.7$) **this work**	**0.71**
LSC+ with TD ($k = 19, \sigma = 0.9$) **this work**	**0.67**
Feature-based SVM [22]	0.60
Virtual SVM with 2pix VSVx [21]	0.56
SVC-rbf [23]	0.42
VSVM[b] [22]	0.38

できていないことが分かる．一方，KPDM は境界が滑らかでありクラスを上手く分離できているのが分かる．LSC や LSC+ は KPDM ほど境界は滑らかではないが，複雑な識別境界を形成しており，PDM よりも識別性能が向上すると予想できる．

5.2　手書き数字パターンを用いた識別性能の比較

次に，公開手書き数字データ MNIST [18] と USPS [19] に提案手法を適用した実験結果を示す．はじめに MNIST を用いて提案手法の性質を調べることにする．MNIST の内訳は訓練パターン数が 60000，未知パターン数が 10000 である．実験では，28×28 ピクセルの手書き数字画像の画素値を文字部が白，背景部が黒となるように白黒を反転させた．

まず，LSC と提案手法におけるパラメータ k とエラー率との関係を調べた．図 12 に学習パターン数 60000 でのエラー率とパラメータ k の関係を示す．なお，未知パターンを固定した片側接距離を用いた提案手法 (LSC_1) の結果も示す．LSC，LSC+ および LSC with TD，LSC+ with TD では k が 10 から 15 程度まではエラー率が低下していき，それ以上になるとエラー率はほとんど変化しなくなる．また，LSC と LSC+ では顕著な違いは見られない，すなわち，LSC で凸集合の外側での距離評価はほとんど行われていないと予想できる．また，LSC_1 と比べて両側接距離を用いた LSC with TD の方がエラー率が小さいことが分かる．

表 1 に MNIST におけるテストパターンに対するエラー率を示す．表中の PDM, TD, LSC, LSC+, LSC with TD 及び LSC+ with TD では最も誤識別が小さくなったパラメータでの結果を示している．PDM の d_l は部分空間の次元数であり，すべてのクラスで共通としている．また，TD の結果は文献 [11] と異なっているが，これは用いた画像のサイズが異なっているためと考えられる．表から提案手法である LSC with TD 及び LSC+ with TD では，LSC や TD を単独で用いた場合よりもエラー率が減少しており，両者を組合わせて使うことが有効であるのが分かる．非常に強力な Support Vector Machine (SVM) を用いた方法 [21], [22] と比較すると，提案手法は特徴抽出やカーネル法を用いることなく同程度のエラー率を達成できているのが分かる．

次に，学習パターン数が少なく，MNIST より識別が困難な USPS を用いてエラー率を調べた．USPS では訓練パターン数が 7291，未知パターン数が 2007 である．表 2 に USPS におけるテストパターンに対するエラー率を示す．MNIST の場合と同様に，表中の PDM, TD, LSC, LSC+, LSC with TD 及び LSC+ with TD では最も誤識別が小さくなったパラメータでの結果を示している．また，TD の結果は文献 [11] と異なっているが，これは用いた学習パターンの数が文献 [11] の方が多いためと考えられる．USPS においても，提案手法のエラー率は LSC や TD を単独で用いた場合よりも減少した．提案手法はこれまでに報告されている USPS の最も低いエラー率 1.9% [26] に近い結果を出すことができた．ちなみに LSC_1 における最低のエラー率は 3.24% となり，USPS においても両側接距離を用いた方がエラー率が小さくなることを確認した．

表 2　USPS におけるテストパターンのエラー率

method	test error rate [%]
Nearest neighbor with Euclidean Distance	5.53
Nearest neighbor with TD ($\sigma = 0.7$)	3.24
PDM ($d_l = 30$)	5.2
LSC ($k = 11$)	3.9
LSC+ ($k = 14$)	4.0
Virtual SVM, local kernel [24]	3.0
Extended TD [13]	2.4
Feature-based virtual SVM [22]	2.34
LSC with TD ($k = 7, \sigma = 1.1$) **this work**	**2.14**
LSC+ with TD ($k = 6, \sigma = 1.1$) **this work**	**2.14**
Combination of TV and local representation [25]	2.0
P2DHMDM with 3-nearest neighbor [26]	1.9
Human error rate [27]	2.5
Human error rate [28]	1.51

6. ま と め

本稿では，線型分離が困難なデータに対して有効な部分空間法の一種である局所部分空間法 (Local Subspace Classifier) を紹介し，それに凸射影を組入れた LSC+ も紹介した．また，LSC と接距離を併用することで高精度に画像パターンを識別する方法を提案した．具体的には，各クラスで未知画像パターンの k 近傍学習パターンを接距離で求めた後，接平面上の近似変形未知パターンの平均パターンと各近傍近似変形学習パターンを用いて LSC による識別を行った．手書き数字データ MNIST と USPS を用いた実験により，LSC や接距離を単体で用いる場合よりもエラー率が低くなったのを確認した．

なお，LSC で使用する主な記憶容量はデータ数を N，次元数を d とすると $N \times d$ であり，$d \ll N$ の場合には，カーネル法が必要とする記憶容量 $N \times N$ より小さくすることができる．一方，LSC の計算コストは未知パターンが入力されてから識別部を設計するので大きくなるが，部分空間法の特長であるクラス毎に独立に識別部を設計できるため並列化を行えば高速化できると期待できる．また，提案手法の計算時間は接距離を高速に計算する方法 [11], [29] を利用することで短縮することが可能である．今後の課題として，学習規則を取り入れて更なる識別精度の向上を図ることや，画像以外のパターンへの適用，処理の高速化が挙げられる．

文　献

[1] S. Watanabe, P.F. Lambert, C.A. Kulikowski, J.L. Buxton, and R. Walker, “Evaluation and selection of variables in pattern recognition,” Computer and Information Sciences, vol. 2, (Julius Tou, ed.). New York: Academic Press, pp. 91–122, 1967.

[2] T. Iijima, H. Genchi, and K. Mori, “A theory of character recognition by pattern matching method,” Proc. of 1st Int. J. Conf. on Pattern Recognition, pp. 50–56, 1973.

[3] 池田正幸, 田中英彦, 元岡 達, “手書き文字認識における投影距離法,” 情処学論, vol. 24, no. 1, pp. 106–112, 1983.

[4] 石井健一郎, 上田修功, 前田英作, 村瀬 洋, “わかりやすいパターン認識,” オーム社, Aug. 1998.

[5] E. Oja, “Subspace methods of pattern recognition,” Research Studies Press, 1983. (小川英光，佐藤 誠訳，パターン認識と部分空間法，産業図書，1986)

[6] 津田 宏治, “ヒルベルト空間における部分空間法,” 信学論, vol. J82-D-II, no. 4, pp. 592–599, Apr. 1999.

[7] 前田 英作, 村瀬 洋, “カーネル非線形部分空間法によるパターン認識,” 信学論, vol. J82-D-II, no. 4, pp. 600–612, Apr. 1999.

[8] J. Laaksonen, “Local subspace classifier,” Proc. of ICANN’97, pp. 637–642, 1997.

[9] J. Laaksonen, “Subspace classifiers in recognition of handwritten digits,” PhD thesis, Helsinki University of Technology, 1997.

[10] P.Y. Simard, Y. LeCun, and J.S. Denker, “Efficient pattern recognition using a new transformation distance,” Proc. of NIPS, vol. 5, pp. 50–58, 1993.

[11] P.Y. Simard, Y. LeCun, J.S. Denker, and B. Victorri, “Transformation invariance in pattern recognition – Tangent distance and tangent propagation,” Int’l J. of Imaging Systems and Technology, vol. 11, no. 3, 2001.

[12] R.O. Duda, P.E. Hart, and D.G. Stork, “Pattern classification,” 2nd edition, John Wiley & Sons, 2001.

[13] D. Keysers, J. Dahmen, T. Theiner, and H. Ney, “Experiments with an extended tangent distance,” Proc. of ICPR, pp. 38–42, 2000.

[14] B. Schölkopf, P. Simard, A. Smola, and V. Vapnik, “Prior knowledge in support vector kernels,” Proc. of NIPS, vol. 10, pp. 640–646, 1998.

[15] B. Haasdonk and D. Keysers, “Tangent distance kernels for support vector machines,” Proc. of ICPR, vol. 2, pp. 864–868, 2002.

[16] A. Pozdnoukhov and S. Bengio, “Tangent vector kernels for invariant image classification with SVMs,” Proc. of ICPR, vol. 3, pp. 486–489, 2004.

[17] K. Fukunaga, “Bias of nearest neighbour error estimation,” IEEE Trans. on Pattern Anal. Mach. Intell., vol. 9, no. 1, pp. 103–112, 1987.

[18] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Intell. Signal Process., pp. 306–351, 2001.

[19] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, and L.D. Jackel, “Backpropagation applied to handwritten zip code recognition,” Neural Computation, vol.1, no.4, pp.541–551, 1989.

[20] Y. LeCun, L.D. Jackel, L. Bottou, J.S. Denker, H. Drucker, I. Guyon, U.A. Muller, E. Sackinger, P. Simard, and V.N. Vapnik, “Comparison of learning algorithms for handwritten digit recognition,” Proc. of Int’l Conf. Artificial Neural Networks, pp. 53–60, 1995.

[21] D. DeCoste and B. Schölkopf, “Training invariant support vector machines,” Machine Learning, vol. 46, no. 1-3, pp. 161–190, 2002.

[22] J.X. Dong, A. Krzyzak, and C.Y. Suen, “Fast SVM training algorithm with decomposition on very large datasets,” PAMI, vol. 27, no. 4, pp. 603–618, 2005.

[23] C.-L. Liu, K. Nakashima, H. Sako, and H. Fujisawa, “Handwritten digit recognition: Benchmarking of state-of-the-art techniques,” Patt. Recog., vol. 36, no. 10, pp. 2271–2285, 2003.

[24] B. Schölkopf, “Support vector learning”, Ph.D. thesis, R. Oldenbourg Verlag, Munchen., Technical University of Berlin, 1997.

[25] D. Keysers, R. Paredes, H. Ney, and E. Vidal, “Combination of tangent vectors and local representation for handwritten digit recognition,” Proc. of SPR2002, vol. LNCS 2396, pp. 538–547, 2002.

[26] D. Keysers, C. Gollan, and H. Ney, “Local context in non-linear deformation models for handwritten character recognition,” Proc. of ICPR2004, vol. 4, pp. 511–514, 2004.

[27] J. Bromley and E. Sackinger, “Neural-network and k-nearest-neighbor classifiers,” Tech. Rep. 11359–910819-16TM, AT&T, 1991.

[28] J.X. Dong, “Statistical results of human performance on USPS database,” Report. CENPARMI, Concordia University, 2001.

[29] A. Sperduti and D.G. Stork, “A rapid graph-based method for arbitrary transformation-invariant pattern classification,” Proc. of NIPS, vol. 7, pp. 665–672, 1994.

A Nonlinear Subspace Method for Pattern Recognition Using Mutli-Layered Perceptrons

Ryo SAEGUSA† and Shuji HASHIMOTO†

† Department of Applied Physics, School of Science and Engineering, Waseda University
Okubo 3-4-1 Shinjuku-ku Tokyo, 169-8555 Japan
E-mail: ryos@ieee.org, shuji@waseda.jp

Abstract Linear subspace method based on principal component analysis has been applied for pattern recognition of high-dimensional data. However, the linear subspace method can not represent nonlinear characteristics of the data-distribution efficiently. In order to overcome this problem, some nonlinear subspace methods have been proposed. In this paper, we propose a novel nonlinear subspace method for pattern recognition using multi-layered perceptrons which can hierarchically construct a nonlinear subspace from the data-distribution. We introduce the optimization scheme of the subspace taking into account the classification. The proposed method achieves high classification accuracy by the optimization scheme and provides the finite subspace to avoid the crossover of the subspaces. We examine its effectiveness through some experiments.

Key words Subspace method, Pattern Recognition, Feature Extraction, Nonlinear PCA, Neural Networks, Eigenspace

1. Introduction

In pattern recognition, it is important to extract the appropriate features from the multi-dimensional data and classify the data based on the features. The subspace method has been one of the conventional methods in the pattern recognition of categorical data [1].

The subspace method performs feature extraction and classification simultaneously [2]. In the method, a subspace is constructed for each category with the samples which belong to the category. The base vectors of each subspace are obtained by Principal Component Analysis (PCA) [3]. The obtained base vectors are applied to span the subspace in the order of the contribution to the sample representation. Therefore, the spanned subspace represents the objective category most efficiently. After constructing the subspace for each category, an input sample is classified into the category of which subspace has the greatest similarity.

Subspace-based methods have some varieties regarding the projection method such as the standard, differential and kernel subspaces [4] and is often applied to the image recognition [5]. In these methods, the subspace is usually optimized for the corresponding categorical data without consideration of the other categories. Therefore, the subspace is not always optimal to classification. The learning subspace method proposed in [6] iteratively constructs the subspace to minimize the classification error by rotating the subspace regarding the miss-classified samples.

The above-mentioned methods utilize a linear subspace. However, it is reported that nonlinear characteristics exist in some data sets, such as face images with emotional expressions [7] and the object images in different orientations [8]. When we extract the features and classify these data, a nonlinear subspace method is considered to perform more effectively than these linear subspace methods.

Recently, some methods of Nonlinear PCA (NLPCA) have been developed. The method by Gnanadesikan [9] applies PCA to a vector of which components are polynomial terms generated from the components of an input vector. The method has difficulty with high-dimensional data, since the number of the polynomial terms increases in the polynomial order of the dimension of the input vector.

Kernel PCA (KPCA) by Schölkopf et al. [10] is an effective method of NLPCA which applies linear PCA to the nonlinearly mapped image of an input vector. Studies of KPCA are widely developed [11], [12]. However, KPCA poses problems for practical use regarding dimension reduction and data-reconstruction.

The Sandglass-type Multi-Layered Perceptron (SMLP) by Irie et al. [13] and by DeMers et al. [14] can construct

adequate nonlinear mapping functions to extract a low-dimensional internal representation from given data. In this method, the dimension of the internal representation is fixed.

The training method proposed by Takahashi et al. [15] enables an SMLP to extract the ordered principal components. The authors of the paper mathematically proved that, in a case where the training method is applied to the three layered linear MLP, the outputs of the units in the bottleneck layer converge to the ordered principal components obtained by PCA.

We have proposed a novel method of NLPCA which can hierarchically construct a nonlinear subspace to fit the data-distribution [16], [17]. The hierarchical architecture of the proposed method allows us to select the dimension of the subspace after the optimization and extend the dimension only with the optimization of the additional mapping functions.

In this paper, we propose a novel nonlinear subspace method based on the proposed NLPCA. We introduce new optimization scheme of the subspace taking into account classification. The optimization scheme is expected to achieve high classification accuracy and the nonlinearity of the proposed method will provide the finite subspace to avoid the crossover of the subspaces. We examined its effectiveness through some experiments.

In the section 2., we formulate the proposed NLPCA and a novel nonlinear subspace method. In the section 3., we demonstrate classification regarding two dimensional artificial data. Finally, conclusion and perspectives are given in the section 4.

2. Method

2. 1 Nonlinear principal component analysis

In this section, we formulate the proposed method of NLPCA which provides a nonlinear subspace.

The purpose of NLPCA is to construct the low-dimensional subspace which represents the characteristics of the high-dimensional data.

Let $\vec{x} \in \mathbb{R}^n$, $\vec{y} \in \mathbb{R}^m$ and $\vec{z} \in \mathbb{R}^n$ be the input vector in a high-dimensional space, the extracted vector (principal components) in a low-dimensional space and the reconstructed vector in the original high-dimensional space, respectively, where $n, m \in \mathbb{N}$ $(n \geqq m)$ indicate the dimension of the input vectors and the principal component vectors, respectively. $\mathbb{R}$ and $\mathbb{N}$ represent a set of real numbers and natural numbers.

In the conventional NLPCA, the nonlinear mapping function $\vec{\ }: \mathbb{R}^n \mapsto \mathbb{R}^m$ is defined as

$$\vec{y} = \vec{\ }(\vec{x}), \tag{1}$$

while the nonlinear mapping function $\vec{\psi} : \mathbb{R}^m \mapsto \mathbb{R}^n$ is defined as

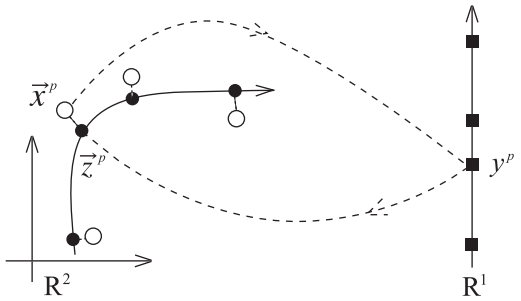


Fig. 1 A concept of NLPCA.

$$\vec{z} = \vec{\psi}\ (\vec{y}). \tag{2}$$

These mapping functions associate the input data with their principal components nonlinearly.

The $\vec{\ }$ and $\vec{\psi}$ are optimized by minimizing the Mean Square Error (MSE):

$$E = \langle ||\vec{x} - \vec{z}||^2 \rangle \tag{3}$$

$$= \langle ||\vec{x} - \vec{\psi}(\vec{\ }(\vec{x}))||^2 \rangle, \tag{4}$$

where $\langle \cdot \rangle$ represents the expectation and $|| \cdot ||$ represents L_2 norm. A smaller MSE indicates the higher fidelity of the reconstruction.

Figure 1 illustrates the concept of the dimension reduction from R^2 to R^1 by NLPCA. In Figure 1, NLPCA nonlinearly maps the data vector $\vec{x} \in R^2$ onto $y \in R^1$, and also nonlinearly maps $y \in R^1$ onto $\vec{z} \in R^2$. If the nonlinear mapping functions are continuous and monotonous, any $\vec{x}$ is mapped onto a curved line in R^2. The MSE of NLPCA corresponds to the average of the squared distance between an input $\vec{x}$ and the mapped $\vec{z}$ onto the curved line.

In the proposed NLPCA, in order to construct principal components: $y_1, \cdots, y_m$ of $\vec{y} = (y_1, \cdots, y_m)$ in the significant order to represent an input vector $\vec{x}$, the nonlinear mapping function $_i : \mathbb{R}^n \mapsto \mathbb{R}^1$ is defined as

$$y_i = \ _i(\vec{x}) \quad \text{for} \quad i = 1, \cdots, m, \tag{5}$$

while the nonlinear mapping function $\vec{\psi}_i : \mathbb{R}^i \mapsto \mathbb{R}^n$ from the product space $(y_1, \cdots, y_i) \in \mathbb{R}^i$ onto the corresponding vector $\vec{z}_i \in \mathbb{R}^n$ is defined as

$$\vec{z}_i = \vec{\psi}_i(y_1, \cdots, y_i) \quad \text{for} \quad i = 1, \cdots, m. \tag{6}$$

The pairs of $\{(\ _i, \vec{\psi}_i)\}_{i=1 \cdots m}$ are adjusted in the increasing order of i. $_i$ and $\vec{\psi}_i$ are optimized with the ith MSE:

$$E_i = \langle ||\vec{x} - \vec{z}_i||^2 \rangle \tag{7}$$

$$= \langle ||\vec{x} - \vec{\psi}_i(y_1, \cdots, y_{i-1}, \ _i(\vec{x}))||^2 \rangle, \tag{8}$$

where $y_1, \cdots, y_{i-1}$ are given. Consequently, the pair of $(\ _k, \vec{\psi}_k)$ is adjusted to perform the best extractor and reconstructor combined with the previous pairs of mapping

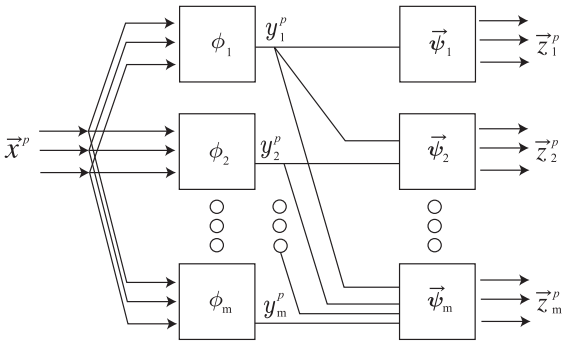


Fig. 2 The diagram of the proposed NLPCA.

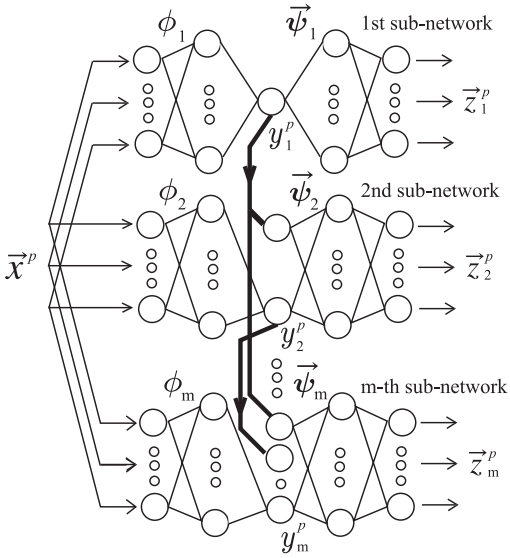


Fig. 3 An implementation of the proposed NLPCA with neural networks. The bold links in the figure are only used for forward propagation. The error signals are not back-propagated through the links.

functions: $\{(\ _i, \vec{\psi}_i)\}_{i=1\ \cdots\ k-1}$. Figure 2 shows the diagram of the proposed NLPCA.

The nonlinear mapping functions of the proposed NLPCA can be implemented with an ensemble of neural networks which have a hierarchical structure as shown in Figure 3. The network is composed of m sub-networks. In the ith sub-network, the first three layers and the last three layers play the role of $_i$ and $\vec{\psi}_i$, respectively. The ith sub-network is given the values of principal components $y_1, \cdots, y_{i-1}$ from the higher 1st, $\cdots$, $(i-1)$th sub-networks. The sub-networks are adjusted in the increasing order of i with the back propagation algorithm.

2.2 Nonlinear subspace method

In this section, we propose a novel nonlinear subspace method based on the NLPCA described in the previous section.

Here, we introduce the d dimensional nonlinear subspace S spanned in the $n(\geqq d)$ dimensional input data space. The d dimensional subspace is constructed by the mapping functions of the proposed NLPCA using d principal components. As shown in Figure 4(a), the mapping functions project the input vector $\vec{x} \in \mathbb{R}^n$ onto $\vec{z} \in \mathbb{R}^n$ of the d dimensional Subspace S spanned in $\mathbb{R}^n$.

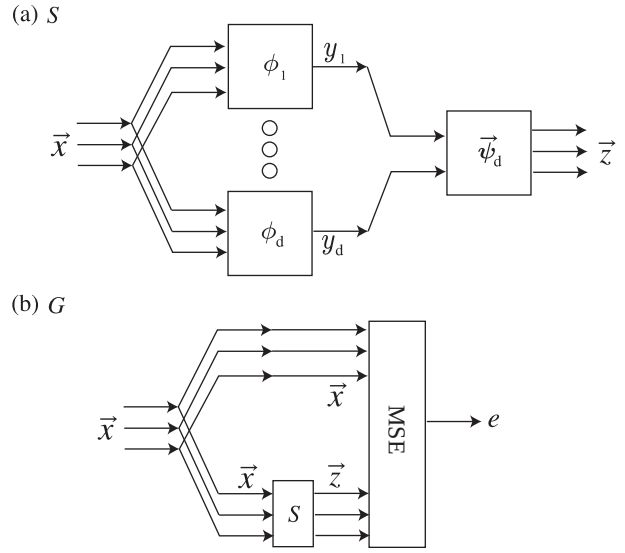


Fig. 4 (a) The mapping functions of a subspace: S to project the input vector $\vec{x}$, (b) Dissimilarity function: $G(\vec{x})$ to measure the dissimilarity of the input vector $\vec{x}$ from the subspace S.

We can measure the dissimilarity of the input $\vec{x}$ from the subspace S by calculating the squared error between $\vec{x}$ and $\vec{z}$ as shown in Figure 4(b). Let us define the dissimilarity function of S with L_2 norm as

$$G(\vec{x}) = ||\vec{x} - \vec{z}||^2. \tag{9}$$

When the G($\vec{x}$) indicates a small value, the $\vec{x}$ is considered to have similarity to the subspace S. If a subspace is optimized enough to represent the samples of a class ω, the function: $G(\vec{x})$ can distinguish the samples of the class from the samples of the other class.

We proposed a novel nonlinear subspace method based on the above-mentioned nonlinear subspaces. In the following, we provide two types of the nonlinear subspace methods (Method I and Method II). We assume the classification of a categorical set $\Omega \in \{\omega_1, \cdots, \omega_c\}$ where ω_i indicates the i-th class and c indicates the number of classes The two methods are different in optimization approaches of the subspace.

The method I is a nonlinear subspace method of which subspace is optimized for each corresponding class in advance of the classification. The $G_i(\vec{x})$ of the subspace S_i is optimized with samples $\vec{x} \in \omega_i$. After the optimization, an unknown-class sample $\vec{x}$ is classified into the class: ω_k as follows,

$$\vec{x} \in \omega_k \ \text{ such that } \min_{i=1\ \cdots\ c} \{G_i(\vec{x})\} = G_k(\vec{x}). \tag{10}$$

Figure 5(a) and (b) shows the optimization process and the

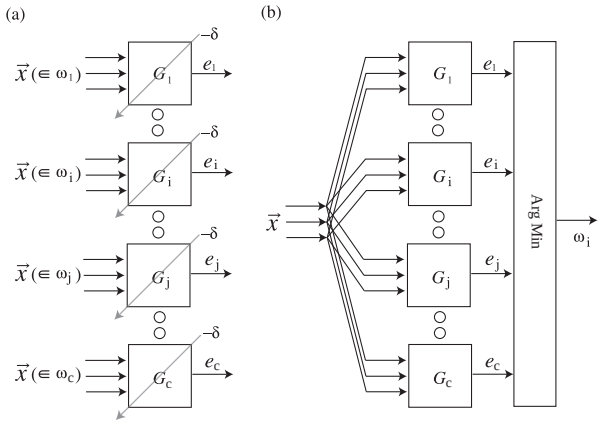


Fig. 5 Nonlinear subspace method (Method I). (a) Optimization process of each subspace, (b) Classification process using the optimized subspaces.

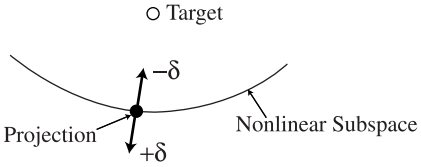


Fig. 6 The optimization of the subspaces

classification process, respectively. In the figure (b), the parameters w of the subspace's mapping functions are adjusted to minimize the error: e by

$$\Delta w = -\eta \frac{\partial e}{\partial w}. \tag{11}$$

We describe this error minimization as $-\delta$ optimization in the bellow.

The concept of the optimization is illustrated in Figure 6. In the figure, the target point is projected onto the point on a nonlinear subspace. By the $-\delta$ optimization, the nonlinear subspace is attracted to the target point.

The each subspace of the Method I is optimized only for the corresponding class in no consideration of the other classes. Therefore, the classification of the samples which belong to the similar categories is expected to fail. We then reflect the classification result to the subspace optimization in the Method II defined in the following.

The Method II is a nonlinear subspace method of which subspace is optimized for the corresponding class with keeping away from the samples of other similar classes. In the Method II, the optimization process and classification process are combined as shown in Figure 7.

In the optimization process, the parameters w of the subspace's mapping functions are adjusted to minimize the error as in the case of Method I. Additionally, if the result of the classification is incorrect, the subspace of the incorrect class which gives the minimum error is adjusted to maximized the error to force the subspace away from the miss-classified

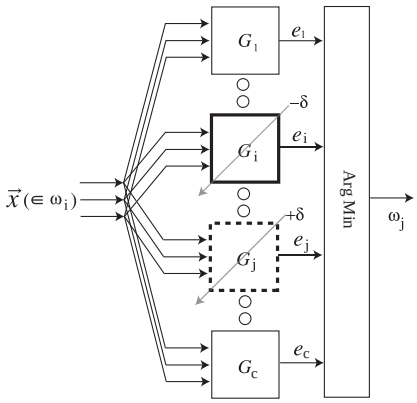


Fig. 7 Nonlinear subspace method (Method II).

sample.

The parameters w of the subspace's mapping functions are adjusted by

$$\Delta w = + \frac{\partial E}{\partial w}. \tag{12}$$

We describe this error maximization as $+\delta$ optimization in the bellow. The concept of the optimization is illustrated in Figure 6. By the $+\delta$ optimization, the nonlinear subspace goes away from the target point.

The Method II is considered to provide higher classification accuracy than Method I due to the additional $+\delta$ optimization, however the representation ability for the corresponding class may be slightly lost. The $+\delta$ optimization and $-\delta$ optimization are in the relation of tradeoff.

In the proposed method, we can iteratively extend the dimension: d of the subspaces in the optimization process taking into account the classification results. This advantage is derived from the proposed NLPCA which can construct the subspaces hierarchically in the order from the low dimension.

3. Experiment

We tried the classification of two categorical data $\Omega_1 = \{\omega_1,\ \omega_2\}$ as shown in Figure 8 (a). The sample vectors $\vec{x} = (x_1, x_2) \in \mathbb{R}^2$ of the ω_1 are distributed around the curved segment:

$$-1.8\ (x_1^2 + 0.3)^2 + \ x_2 + 1.0 = 0, \quad x_1 \in [-1.0,\ 0.4], \tag{13}$$

while the sample vectors of the ω_2 are distributed around the curved segment:

$$1.8\ (x_1^2 - 0.3)^2 + \ x_2 - 1.0 = 0, \quad x_1 \in [-0.4,\ 1.0]. \tag{14}$$

The samples of each category were produced by adding Gaussian noise to the points on the curved segment, respectively. The numbers of training samples of ω_1 and ω_2 were set to 1,000, respectively. The numbers of test samples of ω_1 and ω_2 were also set to 1,000, respectively. We optimized the

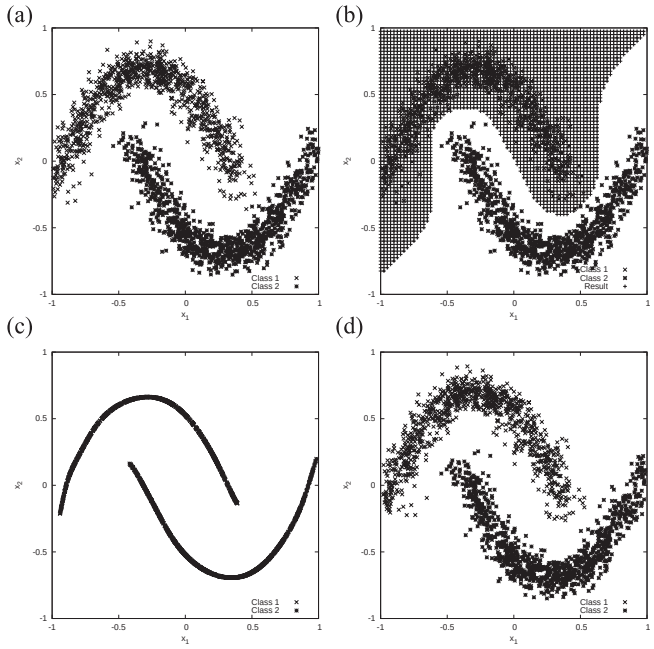


Fig. 8 The results of two categorical data: Ω_1 by Method II. (a) Input samples, (b) Classification results, (c) One dimensional projection, (d) Two dimensional projection.

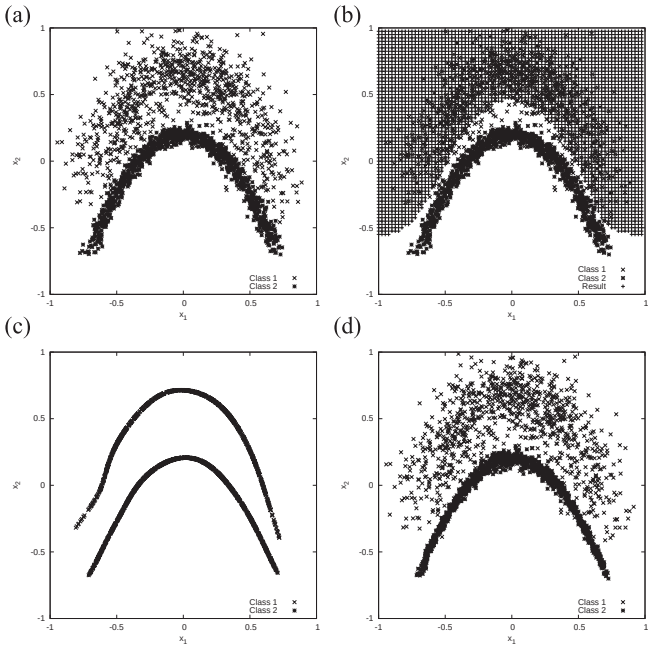


Fig. 9 The results of two categorical data: Ω_2 by Method I. (a) Input samples, (b) Classification results, (c) One dimensional projection, (d) Two dimensional projection.

parameter	value
principal components	2
hidden neurons	30
learning rate (η)	0.008
learning rate (θ)	0.008 3.5
nonlinearity (T)	0.1
training iterations (per a class)	200,000
initial weights	[-0.03, 0.03]

Table 1 Parameters of NLPCA.

Method II with the parameters shown in Table 1 and performed the classification.

Figure 8 shows the input samples (a), classification results (b), one dimensional projection (c), two dimensional projection (d), respectively.

In the figure (b) of the classification results, the area classified into ω_1 is illustrated with crosses. The other area was classified into ω_2. We can find that the nonlinear border provides good classification.

The bottom figures of (c) and (d) show the samples projected onto the one and two dimensional subspaces, respectively. In the figure (c), we can find that the U-shaped nonlinear axes which effectively represent the data in $\mathbb{R}^2$. These figures suggest that the nonlinear subspace provides not only the nonlinear projection but also the projection onto the finite area. This finiteness is effective to avoid the crossover of the subspaces spanned by the different classes. The linear subspace method cannot avoid this problem, since two linear subspaces spread infinitely and crossover unless the subspaces are parallel.

Next, we tried the classification of two categorical data $\Omega_2 = \{\omega_1, \omega_2\}$ as shown in Figure 9 (a) and Figure 10 (a). The sample vectors $\vec{x} = (x_1, x_2) \in \mathbb{R}^2$ of the ω_1 are distributed around the curved segment:

$$1.8\ x_1^2 + \ x_2 + 0.7 = 0, \quad x_1 \in [-0.7,\ 0.7], \tag{15}$$

while the sample vectors of the ω_2 are distributed around the curved segment:

$$1.8\ x_1^2 + \ x_2 + 0.2 = 0, \quad x_1 \in [-0.7,\ 0.7]. \tag{16}$$

The samples of each category were produced by adding Gaussian noise to the points on the curved segment, respectively. The noise of the ω_1 was set larger than the ω_2. The other settings were same as the previous experiment. We optimized the proposed Method I and Method II with the training samples.

Figure 9 and Figure 10 show the input samples (a), classification results (b), one dimensional projection (c), two dimensional projection (d), respectively.

Comparing the classification results of Method I and Method II, we can find that the nonlinear border of the Method II provides better classification than that of the Method I. The border of the Method I is attracted to the widely-distributed ω_1, while the border of the Method II divides two categorical distributions effectively. Table 2 shows the effectiveness of the Method II for the classification.

4. Conclusion

In this paper, we proposed a novel nonlinear subspace

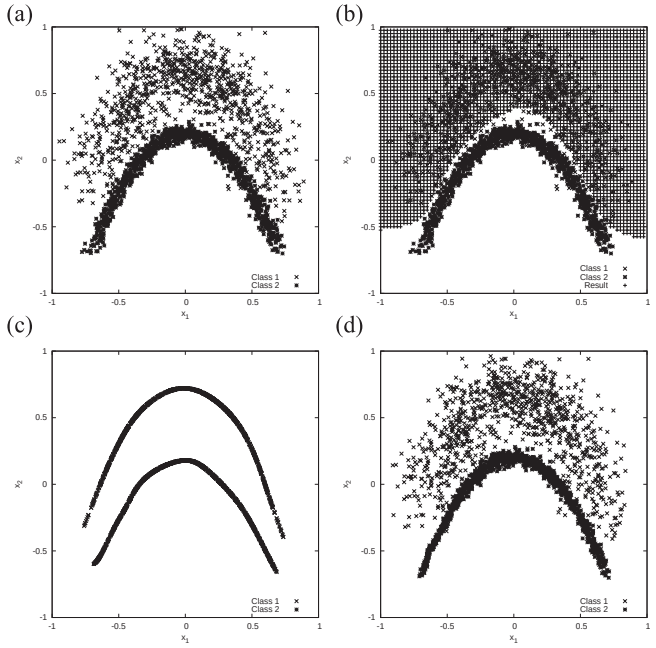


Fig. 10 The results of two class data: Ω_2 by Method II. (a) Input samples, (b) Classification results, (c) One dimensional projection, (d) Two dimensional projection.

Method	1D subspace	2D subspace
Method I	0.9505	0.9000
Method II	0.9995	0.9950

Table 2 Recognition Rate.

method for pattern recognition using multi-layered perceptrons which can hierarchically extract a nonlinear subspace from the data-distribution. We introduced the optimization scheme of the subspace taking into account the classification. The proposed method achieved high classification accuracy by the optimization scheme and provided the finite subspace to avoid the crossover of the subspaces. We examined the effectiveness of the proposed method through the experiments with two dimensional artificial data.

In the proposed method, we can choose any nonlinear mapping functions and optimization approach to construct the subspace. We should examine not only the neural networks but also the various mapping functions for practical problems of pattern recognition. We are now applying the proposed method to the recognition of high-dimensional images such as human faces and handwritten numerals.

Acknowledgements

This work was supported in part by the following organizations: (i) "Information compression using nonlinear principal component analysis," Grant-in-Aid for Young Scientists (B), the Ministry of Education, Culture, Sports, Science and Technology Japan, (ii) "Establishment of consolidated research institute for advanced science and medical care," Encouraging Development Strategic Research Centers Program, the Special Coordination Funds for Promoting Science and Technology, Ministry of Education, Culture, Sports, Science and Technology, Japan, (iii) "The innovative research on symbiosis technologies for human and robots in the elderly dominated society," 21st Century Center of Excellence (COE) Program, Japan Society for the Promotion of Science, and (iv) the Grant-in-Aid for the WABOT-HOUSE Project, Gifu Prefecture, Japan.

References

[1] R. Duda, P. Hart and D. Stork, Pattern Classification 2ed., Springer-Verlag, 2000.

[2] Watanabe, S. Knowing & Guessing -Quantitative Study of Inference and Information, J. Wiley, 1969.

[3] H. Hotelling, "Analysis of complex statistical variables into principal components," Journal of Educational Psychology, Vol.24, pp.417–441, pp.498-520, 1933.

[4] J. Ruiz-del-Solar and P. Navarrete,Eigenspace-based face recognition: a competitive study of different approaches, IEEE Trans. Syst., Man, Cybern., C, vol.35, no.3, pp.315–325, 2005.

[5] M.Turk and A.Pentland, "Eigenfaces for recognition," Journal of Cognitive Neuroscience, Vol.3, No.1, pp.71-86, 1991.

[6] Erkki Oja, Subspace Methods of Pattern Recognition, Research Studies Press and J. Wiley, 1983.

[7] T. Suenaga, A. Sato and H. Sakano, "Cluster discriminant analysis for feature space visualization," Proc. of IEE Int. Conf. on KES, pp.146-150, 2002 (in Japanese).

[8] H. Murase and S. Nayar, "3D object recognition from appearance -Parametric Eigenspace Method-," Journal of IEICE, Vol.J77-D-II, No.11, pp2179-2187, 1994 (in Japanese).

[9] R. Gnanadesikan, Methods for Statistical Data Analysis of Multivariate Observations, John Wiley & Sons Inc, 1977.

[10] B. Schölkopf, A. Smola, and K. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," Neural computation, Vol.10, No.5, pp.1299-1319, 1998.

[11] S. Mika, B. Schölkopf, A. Smola, K.-R. Müller, M. Scholz, and G. Rütsch, "Kernel PCA and de-noising in feature spaces," In Advances in Neural Information Processing Systems 11, 1999.

[12] M. Tipping, "Sparse kernel principal component analysis," In Advances in Neural Information Processing Systems 13, 2001.

[13] B. Irie and M. Kawato, "Acquisition of internal representation by multi-layered perceptron," Journal of IEICE, Vol.J73-D-II No.8, pp.1173-1178, 1990 (in Japanese).

[14] D. DeMers and G. Cottrell, "Nonlinear dimensionality reduction," Advances in Neural Information Processing Systems 5, Morgan Kaufmann, pp.580-587, 1993.

[15] T. Takahashi, R. Tokunaga, and Y. Hirai, "On supervised learning algorithm of three-layer linear perceptron - an extension of Baldi-Hrnik's theorem-," Journal of IEICE, vol.J80-D-II, no.5, pp.1267-1275, 1997 (in Japanese).

[16] R. Seagusa, H. Sakano, S. Hashimoto, "Nonlinear principal component analysis to preserve the order of principal components," Neurocomputing, no.61, pp.57-70, 2004.

[17] Ryo Saegusa, Hitoshi Sakano, Shuji Hashimoto, "A Nonlinear Principal Component Analysis on Image Data," IEICE Trans. Vol.E88-D, No.10, pp.2242-2248, 2005.

[18] D. B. Graham and N. M. Allinson, "Characterizing virtual eigensignatures for general purpose face recognition," Face Recognition From Theory to Applications, ed. H. Wechsler, et al., Springer Verlag, 1998. The UMIST Face Database, http://images.ee.umist.ac.uk/danny/database.html

カーネル部分空間法

鷲沢　嘉一[†]　　山下　幸彦[††]

† (独) 理化学研究所 脳科学総合研究センター　〒 351-0198 埼玉県和光市広沢 2 ー 1
†† 東京工業大学　〒 152-8552 東京都目黒区大岡山 2-12-1
E-mail: †washizawa@brain.riken.jp, ††yamasita@ide.titech.ac.jp

あらまし　カーネル部分空間法とその正則化法，抑制型への拡張について解説をする．
キーワード　カーネル部分空間法，主成分分析，KL 変換，KL 部分空間，CLAFIC 法，相対 KL 変換，マーサーカーネル，サポートベクタマシン，カーネル主成分分析，カーネル標本空間射影，カーネル相対主成分分析，正則化，打切り特異値分解，Tikhonov 正則化，縮小ランクウィーナーフィルタ

A family of kernel subspace classifiers

Yoshikazu WASHIZAWA[†] and Yukihiko YAMASHITA[††]

† Brain Science Institute, RIKEN,　2-1, Hirosawa, Wako-shi, Saitama, 351-0198, Japan
†† Tokyo Institute of Technology,　2-12-1, Ookayama, Meguro-ku, Tokyo 152-8552 Japan
E-mail: †washizawa@brain.riken.jp, ††yamasita@ide.titech.ac.jp

Abstract　We present a family of kernel subspace classifiers and their extensions. There are two kinds of regularization methods and they have a degree of freedom about a null space of the operator in the family. Regularization is important to avoid the over-fitting problem in the machine learning theory. The applicable null space of an operator is able to suppress the effect of the other classes. We describe the details and differences of these classifiers and show experimental results.
Key words　Kernel subspace classifier, class feature information compression (CLAFIC) method, relative Karhunen-Loéve transform, Marcer kernel, support vector machine (SVM), Kernel principal component analysis (KPCA), kernel sample space projection classifier (KSP), kernel relative Karhunen-Loéve transform (KRKLT), kernel relative principal component analysis (KRPCA), truncated singular value decomposition (TSVD), Tikhonov regularization, reduced rank Wiener filter

1. Introduction

Support vector machines have been developed since the early of 1990s and they showed very high generalization capability [1] [2] [3] [4] [5]. Their achievements are supported by the principle that maximizes the minimum margin and the (Marcer) kernel method. In the kernel method, an input vector $f \in \mathbb{R}^d$ is mapped from an input space to a higher or an infinite dimensional space $\mathcal{F}$ called the feature space or the linearization space by a non-linear mapping $\Phi : \mathbb{R}^d \to \mathcal{F}$. Then the classifier is constructed and an unknown input vector is classified in the feature space. Instead of calculating Φ, a function $k(\cdot,\cdot) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ which satisfies the equation

$$k(f_1, f_2) = \langle \Phi(f_1), \Phi(f_2) \rangle, \tag{1}$$

is used. Marcer kernel function satisfies the eq. (1) [6] [7] [8]. Using Marcer kernel function, we can calculate an inner product in $\mathcal{F}$ without care of the form of Φ. In other words, Φ is defined automatically if we define Marcer kernel function $k(\cdot,\cdot)$.

Schölkopf et al. introduced another possibility of Mercer kernel function by solving an eigenvalue problem of a correlation matrix in the feature space [9]. This result was applied to the kernel principal component analysis (KPCA) and used for non-linear representation or image restoration [10]. Kernel methods are also applied to the Fisher discrimination [11] [12], the independent component analysis (ICA) [13] or a Wiener filter [14] [15] [16].

Subspace methods are used widely in an industrial area [17]. A class feature information compression (CLAFIC) method is one of the subspace classifiers. It classifies unknown input pattern by comparing projection norms of an unknown input vector onto Karhunen-Loève (KL) subspaces which are constructed by the samples of each class [18] [19].

Yamashita et al. extended a Karhunen-Loève transform (KLT) [20] [21] which is almost equivalent to a principal component analysis (PCA) [22] [23] to the relative KLT (RKLT) or the relative PCA

(RPCA) [24] [25]. It was also applied to pattern classifier [26] [27] [28]. We call this classifier RKLT method. In RKLT method, features of other classes are considered as noise against an objective class. Under this optimization problem, an optimal subspace for the classification is obtained. We describe details in Section 4.

On the other hand, Tsuda and Maeda et al. applied kernel method to CLAFIC method using results of Schölkopf et al. [29] [30] [31] [32]. We call this method KPCA. KPCA shows much higher accuracy rate than CLAFIC method. Furthermore its computational complexity in the multi-class classification problem is lower than SVM. We describe it in detail in Section 6. 1.

Hikida et al. and Washizawa et al. integrated KPCA and RKLT, and proposed kernel relative KLT (KRKLT) or kernel relative PCA (KRPCA) [33] [34] [35]. Washizawa at al. also proposed a kernel based pattern classification method called kernel sample space projection (KSP) method [36] [37]. An operator of KSP is an orthogonal projection operator onto the kernel sample space which is spanned by samples mapped to high dimensional feature space $\mathcal{F}$. They also introduced some regularization to KSP. KPCA can be interpreted as the regularized version of KSP by truncated singular value decomposition (TSVD). Furthermore they extended KSP to suppressed KSP (SKSP) whose operator which extracts the feature of a class is an oblique projection along to the kernel sample space of the other classes onto kernel sample space of the objective class [38] [39]. KRPCA can be also interpreted as the regularized version of SKSP. We summarize these methods in Table 1.

Table 1 Summary of Kernel Subspace Methods

	Input space	Feature space	
Regularization	TSVD	TSVD	Tikhonov
Orthogonal Projection	CLAFIC	KPCA	KSP
Non-orthogonal transform	RKLT	KRKLT	SKSP

In Section 8., we show some experimental results of kernel subspace methods.

Table 2 denotes notations and symbols used in this paper. Except as otherwise noted, capital alphabets denote matrices and operators, lower-case alphabets denote vectors and functions, and Greek alphabet characters denote scalars.

2. Pattern recognition and discriminant function

The objective of pattern recognition is to classify unknown input pattern $f_x \in \mathbb{R}^d$ correctly, i.e, obtain applicable function $y(f_x)$ maps from input space $\mathbb{R}^d$ to a set of class (or category) labels. Let $\mathcal{K}$ be a set of class labels. We often use discriminant functions $d_k(f_x)$ $(k = 1, \ldots, n \in \mathcal{K})$ to represent function $y(f_x)$.

Definition 1 (Discriminant function). *Functions $d_k(f_x)$ $(k = 1, \ldots, n \in \mathcal{K})$ which measure the similarity between a class k and unknown input pattern f_x are called discriminant functions. The class label $y(f_x)$ is obtained as*

Table 2 Notation and symbols

Φ	Non-linear mapping to the feature space $\mathbb{R}^d \to \mathcal{F}$
$k(\cdot,\cdot)$	Mercer kernel function
$d_k(\cdot)$	Discriminant function of class k
A^*	Adjoint operator of A
$A^\dagger$	Moore-Penrose generalized inverse of A
$\langle\cdot,\cdot\rangle$	inner product
$a \otimes \overline{b}$	Neumann-Shatten product, $(a \otimes \overline{b})c = c\langle b, a\rangle$
e_i	i-th natural basis, $(e_i)_j = \begin{cases} 0 & (i \neq j) \\ 1 & (i = j) \end{cases}$
$\mathcal{B}(S_1, S_2)$	Set of bounded linear operator from S_1 to S_2
Ω_i	Set of samples of class i
f_n^i	n-th sample of Ω_i
$\lvert\Omega_i\rvert$	Number of samples in Ω_i
Ψ_i	Set of samples should be suppressed except class i
g_n^i	n-th sample of Ψ_i
$\lvert\Psi_i\rvert$	Number of samples in Ψ_i
S_i	$S_i = \sum_{j=1}^{\lvert\Omega_i\rvert} \Phi(f_j^i) \otimes \overline{e_j} \in \mathcal{B}(\mathbb{R}^{\lvert\Omega_i\rvert}, \mathcal{F}),\ \ e_j \in \mathbb{R}^{\lvert\Omega_i\rvert}$
T_i	$T_i = \sum_{j=1}^{\lvert\Psi_i\rvert} \Phi(g_j^i) \otimes \overline{e_j} \in \mathcal{B}(\mathbb{R}^{\lvert\Psi_i\rvert}, \mathcal{F}),\ \ e_j \in \mathbb{R}^{\lvert\Psi_i\rvert}$
U_i	$U_i = \sum_{j=1}^{\lvert\Omega_i\rvert} \Phi(f_j^i) \otimes \overline{e_j} + \sum_{k=1}^{\lvert\Psi_i\rvert} \Phi(g_k^i) \otimes \overline{e_{\lvert\Omega_i\rvert+k}}$
K_{S_i}	Kernel Gram matrix of class i, $K_{S_i} = S_i^* S_i$
K_{U_i}	Kernel Gram matrix of all classes $K_{U_i} = U_i^* U_i$
f_x	Unknown input vector
$\mathcal{R}(A), \mathcal{N}(A)$	Range and null space of A respectively
$P_{\mathcal{R}(A)}$	Orthogonal projection operator onto $\mathcal{R}(A)$
$\lVert f\rVert$	l_2 norm $\lVert f\rVert = \sqrt{\langle f, f\rangle}$
$\lVert A\rVert_{\mathrm{lub}}$	Operator norm $\lVert A\rVert_{lub} = \sup_{f\neq 0} \frac{\lVert Af\rVert}{\lVert f\rVert}$
$\lVert A\rVert_F$	Frobenius norm $\lVert A\rVert_2 = \sqrt{\mathrm{tr}(A^*A)}$
I	identity matrix, identity operator

$$y(f_x) = \underset{k\in\mathcal{K}}{\operatorname{argmax}}\ d_k(f_x). \tag{2}$$

Then the objective is to obtain applicable functions $d_k(f_x)$ $(k = 1, \ldots, n)$ that extract the intrinsic feature of classes. In some cases, argmax in the eq. (2) is replaced by argmin. Then we can express with argmax by adding a negative sign. The simple example of the discriminant function is $d_k(f_x) = -\lVert f_k - f_x\rVert$, where f_k is a prototype of class k.

Note that this kind of discriminant function differs from it of binary classifiers such as SVM or Fisher discriminant. In those cases, its sign denotes the class.

3. CLAFIC method

Let $\{f_1^i, f_2^i, \ldots f_{\lvert\Omega_i\rvert}^i\} \in \Omega_i$ be a set of samples of class i.

Definition 2 (CLAFIC [18] [17]). *The discriminant function of CLAFIC method is defined as*

$$d_k(f_x) = \lVert P_k f_x\rVert^2 \tag{3}$$

$$P_k = \underset{X;\mathrm{rank}(X)\leqq\eta}{\operatorname{argmin}} \sum_{f\in\Omega_k} \lVert f - Xf\rVert^2. \tag{4}$$

Let a sample correlation matrix of class k be R_k, and its eigenvalue decomposition is given as

$$R_k = \sum_{f\in\Omega_k} f \otimes \overline{f} = \sum_{l=1}^{d} \lambda_l^k (u_l^k \otimes \overline{u_l^k}), \tag{5}$$

where the operator $(\cdot \otimes \bar{\cdot})$ is Neumann-Shatten product defined by $(a \otimes b)c = \langle c, b \rangle a$. It is equivalent as $ab^\top$ in a real finite dimensional space, however we use this notation consistently because we have to deal with infinite space when we introduce the kernel method.

Theorem 1. *Suppose that eigenvalues λ_l^k $l = 1, 2, \ldots, d$ are sorted in descending order in eq. (5). The solution of eq. (4) is a projection matrix [17] [19] [18]*

$$P_k = \sum_{j=1}^{\eta^k} (u_j^k \otimes \overline{u_j^k}). \tag{6}$$

The parameter η is obtained using cumulative proportion. However its accuracy is sensitive against η. If η is too small, feature of a class cannot extract enough. If η is too large, overlap of classes decreases its accuracy.

4. Relative Karhunen-Loève Transform (RKLT) method

A feature extraction operator of CLAFIC method P extracts feature of a class. However if plural classes have similar feature, the operator extracts both feature and they cannot be discriminated. Let Ψ_k be a set of sample which should be suppressed with respect to class k. Relative Karhunen-Loève Transform (RKLT) method overcome this problem using following optimization problem [24] [25]:

Definition 3. *The operator of RKLT is defined as*

$$B_k = \underset{X;\mathrm{rand}(X) \leq \eta}{\mathrm{argmin}} \sum_{f \in \Omega_k} \|f - Xf\|^2 + \alpha \sum_{g \in \Psi_k} \|Xg\|^2, \tag{7}$$

where α is a parameter which control the effect of the second term.

The first term of eq. (7) is the same as CLAFIC method. The second term suppresses the feature of other classes. RKLT is equivalent to CLAFIC when $\alpha = 0$, and equivalent to reduced rank Wiener filter when $\alpha = 1$ and f, g have no cross correlation [40].

Theorem 2. *Let $R_k = \sum_{f \in \Omega_k} (f \otimes \overline{f})$, $Q_k = \sum_{g \in \Psi_k} (g \otimes \overline{g})$. $A^\dagger$ denotes a Moore-Penrose generalized inverse of the operator A [41]. Suppose that eigenvalue decomposition of $R_k(R_k + Q_k)^\dagger R_k$ is expressed as*

$$R_k(R_k + Q_k)^\dagger R_k = \sum_{j=1}^{d} \lambda_j^k (u_j^k \otimes \overline{u_j^k}), \tag{8}$$

and λ_j^k is sorted in descending order. Then the solution of RKLT is given as

$$B_k = \sum_{j=1}^{\eta} (u_j^k \otimes \overline{u_j^k}) R(R + Q)^\dagger + W(I - (R + Q)(R + Q)^\dagger), \tag{9}$$

where W is an arbitrary matrix and I is an identity matrix.

Note that if $(R + Q)$ is full rank matrix, the second term is zero.

Since the feature extraction operator of RKLT B_k is not orthogonal projector, there are two kinds of discriminant functions:

$$d_k^1(f_x) = \|B_k f_x\|^2 \tag{10}$$

$$d_k^2(f_x) = -\|f_x - B_k f_x\|^2. \tag{11}$$

In the case of CLAFIC method, these are equivalent. Eq. (11) shows higher accuracy rate at an experiment in [28].

5. Kernel sample space projection

Here, we introduce the kernel sample projection classifier (KSP) to describe kernel subspace method systematically. All of kernel subspace methods can be interpreted as the extension of KSP.

Let $\{f_1^i, f_2^i, \ldots, f_{|\Omega_i|}^i\} \in \Omega_i$ be a set of samples of class i, $k(\cdot, \cdot)$ be a Mercer kernel function, and $\Phi : \mathbb{R}^d \to \mathcal{F}$ be a mapping led from the Mercer kernel function. At first, we introduce a kernel sample space which is spanned by samples mapped to feature space $\mathcal{F}$. Let an operator

$$S_i = \sum_{j=1}^{|\Omega_i|} \Phi(f_j^i) \otimes \overline{e_j}, \tag{12}$$

where $e_j \in \mathbb{R}^{|\Omega_i|}$ is a natural basis which has only one '1' element in j-th row and the rest elements are zero. If $\Phi(f_j^i)$ is a column vector in a finite dimensional space, S_i is a matrix expressed as $S_i = [\Phi(f_1^i)\ \Phi(f_2^i) \ldots \Phi(f_{|\Omega_i|}^i)]$. A kernel sample space of class i is expressed as $\mathcal{R}(S_i)$, where $\mathcal{R}(A)$ denotes a range of A.

An operator $S_i^* \Phi(\cdot) : \mathbb{R}^d \to \mathbb{R}^{|\Omega_i|}$ so-called the empirical kernel map [7] is reduced as

$$\begin{aligned} S_i^* \Phi(f) &= \sum_{j=1}^{|\Omega_i|} (e_j \otimes \overline{\Phi(f_j^i)}) \Phi(f) \\ &= \sum_{j=1}^{|\Omega_i|} \langle \Phi(f), \Phi(f_j^i) \rangle e_j \\ &= (k(f, f_1^i)\ k(f, f_2^i)\ \ldots k(f, f_{|\Omega_i|}^i))^\top, \end{aligned} \tag{13}$$

where A^* denotes the adjoint operator of A.

A (kernel) Gram matrix K_{S_i} is defined as

$$K_{S_i} = S_i^* S_i = \begin{bmatrix} k(f_1^i, f_1^i) & \ldots & k(f_1^i, f_{|\Omega_i|}^i) \\ \vdots & \ddots & \vdots \\ k(f_{|\Omega_i|}^i, f_1^i) & \ldots & k(f_{|\Omega_i|}^i, f_{|\Omega_i|}^i). \end{bmatrix} \tag{14}$$

Definition 4 (Kernel sample projection [37] [36]). *The discriminant function of KSP is expressed as*

$$d_i(f_x) = \|P_{\mathcal{R}(S_i)} \Phi(f_x)\|^2 = \langle S_i^* \Phi(f_x), K_{S_i}^\dagger S_i^* \Phi(f_x) \rangle, \tag{15}$$

where $P_{\mathcal{R}(S_i)} = S_i K_{S_i}^\dagger S_i^*$ is an orthogonal projector onto kernel sample space $\mathcal{R}(S_i)$.

Thus the similarity between class i and unknown input patter f_x is measured by the projection norm onto the kernel sample space $\mathcal{R}(S_i)$.

In order to extend KSP, we redefine the feature extraction operator of KSP:

Definition 5 (Kernel sample space projection). *The operator of KSP is defined by following optimization problem.*

$$\begin{aligned} \min_{X_i}: \quad & J[X_i] = \frac{1}{|\Omega_i|}\sum_{f\in\Omega_i} \|\Phi(f) - X_i\Phi(f)\|^2 \\ \text{subject to:} \quad & \mathcal{N}(X_i) \supset \mathcal{R}(S_i)^\perp, \end{aligned} \tag{16}$$

where $\mathcal{N}(A)$ denotes the null space of A and $^\perp$ denotes an orthogonal complement space.

The constraint means that components which is not in samples are projected to the kernel sample space by X_i. Actually, in the CLAFIC or RKLT methods are also needed this constraint. However in almost cases, since samples span whole space, this constraint has no effect.

Proposition 1. *The solution of the optimization problem (16) equals to $P_{\mathcal{R}(S_i)}$.*

Proposition 1 can be proved from results of the Appendix easily.

6. Regularization of KSP

KSP method can classify all train samples correctly if the kernel Gram matrix K_{S_i} is not singular. If we use mapping Φ that maps to an infinite functional space, e.g., Gaussian kernel function $k(f_1, f_2) = \exp(-\|f_1 - f_2\|^2/c)$, kernel Gram matrix K_{S_i} is always nonsingular unless the same samples exist. However, since there are noisy samples or outliers in a sample set, the generalization error is large because of the over-fitting problem. In order to avoid the over-fitting problem, we can introduce the regularization technique, e.g., soft margin techniques for the SVM or the Ada-Boost [42], weight decay parameter for the neural networks [43], or the ridge regression. In a field of the linear inverse problems, the regularization has been discussed for long time. The truncated singular decomposition (TSVD) and Tikhonov regularization are major techniques of the regularization [44] [45] [46] [47].

6.1 Kernel principal component analysis

KPCA can be interpreted as KSP introduced TSVD. We characterize its operator by an optimization problem.

Definition 6 (KPCA [9] [10]). *The operator of KPCA is defined by following optimization problem.*

$$\begin{aligned} \min_{X_i}: \quad & J[X_i] = \frac{1}{|\Omega_i|}\sum_{f\in\Omega_i} \|\Phi(f) - X_i\Phi(f)\|^2 \\ \text{subject to:} \quad & \mathcal{N}(X_i) \supset \mathcal{R}(S_i)^\perp, \ \ \operatorname{rank}(X_i) \leqq \eta \end{aligned} \tag{17}$$

6.2 Regularized KSP

Another major regularization technique is Tikhonov regularization which is used in ridge regression in the area of the multivariate analysis [45] [44]. The over-fitting problem is caused by ill-condition of a feature extraction operator. One of the measure of the condition of operator is an operator (or a spectral) norm which is defined by

$$\|A\|_{\text{lub}} = \sup_{\|x\|\neq 0} \frac{\|Ax\|}{\|x\|} = \sup_{\|x\|=1} \|Ax\|. \tag{18}$$

If the operator norm of a feature extraction operator is large, small amount of components which have the specific direction dominate the value of the feature. Then the decision boundary become complex and the over-fitting problem appears. However, since it is difficult to suppress the operator norm in eq. (16), we use Frobenius norm defined by $\|A\|_F = \sqrt{\operatorname{tr}(A^*A)} \leqq \|A\|_{\text{lub}}$ instead of using operator norm.

Definition 7 (Regularized KSP [39] [38]). *Regularized KSP is defined as the solution of following optimization problem:*

$$\begin{aligned} \min_{X_i}: \quad & J[X_i] = \frac{1}{|\Omega_i|}\sum_{f\in\Omega_i} \|\Phi(f) - X_i\Phi(f)\|^2 + \epsilon\|X_i\|_F^2 \\ \text{subject to:} \quad & \mathcal{N}(X_i) \supset \mathcal{R}(S_i)^\perp, \end{aligned} \tag{19}$$

where $\epsilon > 0$ is a regularization parameter which controls strength of regularization.

Theorem 3 (Solution of regularized KSP). *The solution of optimization problem (19) is*

$$P^{\mu}_{\mathcal{R}(S_i)} = S_i(K_{S_i} + \mu I)^{-1}S_i^*, \tag{20}$$

where $u = \epsilon|\Omega_i|$.

The proof of the theorem is in Appendix. We call regularized KSP just KSP from now.

6.3 Toy example

Here, we show a toy example in order to show the effect of regularization. Figure 1 shows the problem. Training vectors are 200 two dimensional vectors generated by uniform distribution in $[0, 500] \times [0, 500]$. Actual decision boundaries which are concentric circles shown in the figure discriminate samples to two classes shown by red circles and green crosses. The objective is the reconstructing the decision boundaries from samples.

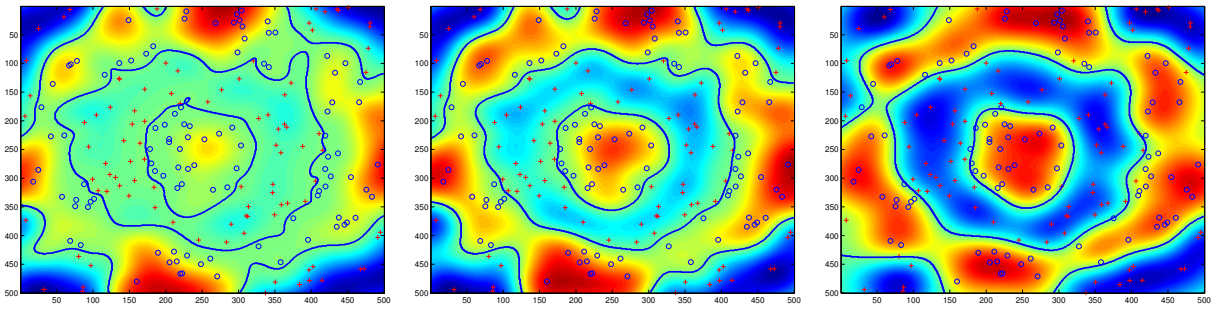
Results are shown in Figure 2. We used Gaussian kernel function $k(x, y) = \exp(-\frac{\|x-y\|^2}{2\cdot 50^2})$. In the case that regularization is not used i.e., $\mu = 0$ or the rank of KPCA is maximum, decision boundary is complex, but all train samples can be classified correctly. On the other hand, in the case that $\mu = 1$ or rank equals to 20, decision boundaries are smooth, but some train samples are misclassified. GE denotes generalization error measured by percentage of pixels which are out of the actual decision boundary because samples are distributed uniformly. In both KPCA and KSP, applicable regularization decrease the generalization error even if some training samples are misclassified.

7. Suppression of the effect of other classes

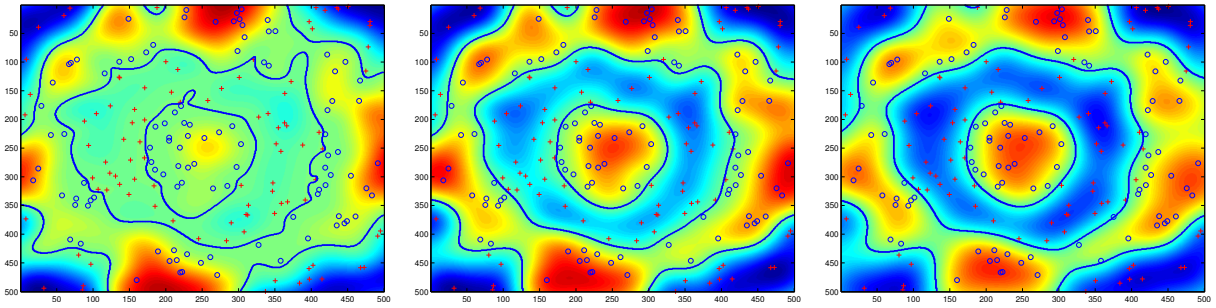
KSP and KPCA can be extended like RKLT. Let

$$U_i = \sum_{j=1}^{|\Omega_i|} \Phi(f_j^i) \otimes \overline{e_j} + \sum_{k=1}^{|\Psi_i|} \Phi(g_k^i) \otimes \overline{e_{|\Omega_i|+k}}. \tag{21}$$

Then definitions of Kernel RKLT and Suppressed KSP is given as follows.



KPCA - Rank=96, GE=14.22% | Rank=50, GE=11.47% | Rank=20, GE=13.76%



KSP $\mu = 0$, GE=14.62% | $\mu = 0.1$, GE=11.78% | $\mu = 1$, GE=12.44%

Figure 2 Results of toy example problem; Upper column: KPCA, Lower column: KSP. Blue line denotes estimated decision boundary and base color shows the value of feature. GE is generalization error i.e., percentage of pixels which are out of the actual decision boundary because samples are distributed uniformly.

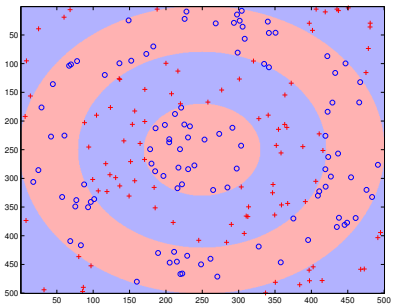


Figure 1 Train vectors of toy example problem: 200 train vectors are uniformly distributed. Actual decision boundaries are concentric circles.

Definition 8 (Kernel RKLT (KRKLT), Kernel RPCA (KRPCA) [35] [33] [34]). *KRKLT is defined by the solution of following optimization problem:*

$$\begin{aligned} \min_{X_i}: \quad & J[X_i] = \frac{1}{|\Omega_i|}\sum_{f\in\Omega_i}\|\Phi(f) - X_i\Phi(f)\|^2 \\ & + \frac{\alpha}{|\Psi_i|}\sum_{g\in\Psi_i}\|X_i\Phi(g)\|^2 \qquad (22) \\ \text{subject to:} \quad & \mathcal{N}(X_i) \supset \mathcal{R}(U_i)^\perp,\ \mathrm{rank}(X_i) \leqq \eta, \end{aligned}$$

where α is a parameter which controls the strength of suppression.

Definition 9 (Suppressed KSP (SKSP) [39] [38]). *SKSP is defined by the solution of following optimization problem:*

$$\begin{aligned} \min_{X_i}: \quad & J[X_i] = \frac{1}{|\Omega_i|}\sum_{f\in\Omega_i}\|\Phi(f) - X_i\Phi(f)\|^2 \\ & + \frac{\alpha}{|\Psi_i|}\sum_{g\in\Psi_i}\|X_i\Phi(g)\|^2 + \mu\|X_i\|_F^2 \qquad (23) \\ \text{subject to:} \quad & \mathcal{N}(X_i) \supset \mathcal{R}(U_i)^\perp, \end{aligned}$$

Theorem 4 (Solution of KRKLT). *We omit an index i which denotes class label to simplify expression. Let $\Lambda_0 = \begin{bmatrix} I_{|\Omega|} & \mathbf{0}_{|\Omega||\Psi|} \\ \mathbf{0}_{|\Psi||\Omega|} & \mathbf{0}_{|\Psi|} \end{bmatrix}$, $\Lambda = \begin{bmatrix} \frac{1}{\sqrt{|\Omega|}} I_{|\Omega|} & \mathbf{0}_{|\Omega||\Psi|} \\ \mathbf{0}_{|\Psi||\Omega|} & \sqrt{\frac{\alpha}{|\Psi|}} I_{|\Psi|} \end{bmatrix}$, $K_U = U^*U$, $A = K_U^{1/2}\Lambda_0\Lambda$, and v_i be i-th eigen vector of $A^\top A$, where suppose that corresponding eigenvalues are sorted in descending order. Then one of the solutions of the optimization problem (22) is given as*

$$X^{KRPCA} = U(K_U^{1/2})^\dagger A \sum_{i=1}^{\eta}(v_i \otimes \overline{v_i})\Lambda^{-1}K_U^\dagger U^*. \qquad (24)$$

Proposition 2. *If K_U is nonsingular, X^{KRPCA} is a projector.*

The proofs of these theorem and proposition appear in Appendix.

Theorem 5 (Solution of SKSP). *The solution of the optimization problem (23) is given as*

$$\tilde{P}^{\mu}_{\mathcal{R}(S)} = U\Lambda_0(K_U + \mu\Lambda^{-2})^{-1}U^*. \qquad (25)$$

Proposition 3. *If K_U is nonsingular and $\mu = 0$, $\tilde{P}^{\mu}_{\mathcal{R}(S)}$ is a projector onto $\mathcal{R}(S_i)$.*

Proposition 4. *If K_U is nonsingular and $\mu = 0$, we have*

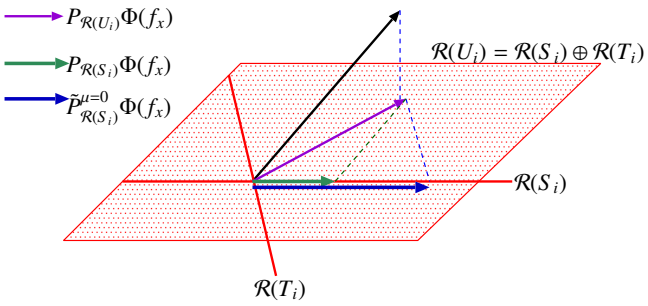


Figure 3 Suppressed kernel sample space projection

Table 3 Results of the handwritten digits classification problem

Method	Parameter	Error rate[%]
KSP with Polynomial kernel	$d = 8$	2.29
KSP with Gaussian kernel	$\sigma = 7$	2.25
KPCA with Polynomial kernel	$d = 7$, rank = 460	2.36
KPCA with Gaussian kernel	$\sigma = 10$, rank = 500	2.32
SKSP with Gaussian kernel	$\sigma = 8$	1.79
3-NN	–	2.4
Polynomial SVM	$d = 4$	1.1

$$\tilde{P}^{\mu}_{\mathcal{R}(S)} P_{\mathcal{R}(U)} = \tilde{P}^{\mu}_{\mathcal{R}(S)} \tag{26}$$

$$P_{\mathcal{R}(U)} \tilde{P}^{\mu}_{\mathcal{R}(S)} = \tilde{P}^{\mu}_{\mathcal{R}(S)}, \tag{27}$$

where $P_{\mathcal{R}(U)}$ is the orthogonal projector onto $\mathcal{R}(U)$.

Proposition 5. *If K_U is nonsingular and $\mu = 0$, $\tilde{P}^{\mu}_{\mathcal{R}(S)} v = 0$ for all $v \in \mathcal{R}(T)$, where $T = \sum_{k=1}^{|\Psi|} \Phi(g_k) \otimes \overline{e_k}$.*

The proofs of these theorem and propositions appear in Appendix.

We show the sketch of SKSP in Fig. 3. From Propositions 3, 4, and 5, $\tilde{P}^{\mu}_{\mathcal{R}(S)}\Phi(f_x)$ can be considered as follows. At first, $\Phi(f_x)$ is orthogonally projected onto $\mathcal{R}(U)$, then it is projected onto $\mathcal{R}(S)$ along $\mathcal{R}(T)$. The similarity between f_x and Ω against Ψ is given as $\|\tilde{P}^{\mu}_{\mathcal{R}(S)}\Phi(f_x)\|$. If $\Psi = \phi$, $\tilde{P}^{\mu}_{\mathcal{R}(S)} = P_{\mathcal{R}(S)}$. Thus SKSP is an extension of KSP.

8. Computational Simulation

8.1 Handwritten digits classification

In order to compare abilities of kernel subspace classifiers, we used a handwritten digit database 'MNIST' provided by the U.S. National Institute of Standard and Technology. It is consisted by 60,000 characters for training and 10,000 characters for testing. Each character is 28x28 pixels and 256 gray scale image.

We show the classification error rate with the parameters which show the lowest error rate in Table 3. The results of the 3-NN (3-Nearest Neighborhood) and the Polynomial SVM is referred from [48] and [5] respectively.

8.2 Binary classification problem

We employ several practical data sets used in [42], [11] and [49](*1). All the data sets we use here are binary classification problems and consist of 100 or 20 realizations.

We use Gaussian kernel function as a kernel function. The parameter of kernel function and the regularization is fixed for all sets. If there are identical samples, we added only one of them to learning set. We use all samples in the other class for Ψ_i since the learning set is not large.

The mean test error rates and their standard deviations are described in Table 4. The results except for KSP and SKSP are referred from the papers above.

Table 4 Mean test error rates and their standards deviations (AB Reg: Regularized AdaBoost, KFD: kernel fisher discriminant). The best method is written in bold face and the second best is emphasized.

dataset	SKSP	KSP	AB Reg	SVM	KFD
Banana	**10.4 ± 0.5**	**10.4 ± 0.5**	10.9 ± 0.4	11.5 ± 0.7	10.8 ± 0.5
Breast-cancer	*26.0 ± 4.6*	29.7 ± 4.5	26.5 ± 4.5	*26.0 ± 4.7*	**24.5 ± 4.6**
Diabetis	**23.0 ± 1.6**	24.5 ± 1.9	23.8 ± 1.8	23.5 ± 1.73	*23.2 ± 1.6*
Flare-solar	37.2 ± 4.5	39.1 ± 2.4	34.2 ± 2.2	**32.4 ± 1.8**	*33.2 ± 1.7*
German	**23.4 ± 2.1**	31.3 ± 2.5	24.7 ± 2.4	*23.6 ± 2.1*	23.7 ± 2.2
Heart	*15.8 ± 3.1*	**15.4 ± 3.3**	16.5 ± 3.5	16.0 ± 3.3	16.1 ± 3.4
Image	*2.8 ± 0.4*	2.9 ± 0.5	**2.7 ± 0.6**	3.0 ± 0.6	4.8 ± 0.6
Ringnorm	18.0 ± 2.3	19.9 ± 1.8	*1.6 ± 0.1*	1.7 ± 0.1	**1.5 ± 0.1**
Splice	11.2 ± 0.7	12.6 ± 0.7	**9.5 ± 0.7**	10.9 ± 0.7	*10.5 ± 0.6*
Thyroid	**4.0 ± 2.3**	*4.2 ± 2.3*	4.6 ± 2.2	4.8 ± 2.2	*4.2 ± 2.1*
Titanic	29.4 ± 10.3	28.3 ± 9.4	*22.6 ± 1.2*	**22.4 ± 1.0**	23.3 ± 2.1
Twonorm	*2.4 ± 0.1*	**2.3 ± 0.1**	2.7 ± 0.2	3.0 ± 0.2	2.6 ± 0.2
Waveform	**9.6 ± 0.4**	11.2 ± 0.6	*9.8 ± 0.8*	9.9 ± 0.4	9.9 ± 0.4
# of bold	5	3	2	2	2
# of emph.	4	1	3	2	3

Consequently, SKSP classifier shows the lowest error rates among those methods in many problems, and SKSP outperformed KSP in most of problems. We can say SKSP can suppress the effect of features in the other class, and can extract important features.

9. Discussion

9.1 Comparison of Regularization

As mentioned above, KPCA and KRPCA use the TSVD, and KSP and SKSP use the Tikhonov regularization. We summarize the difference between the TSVD and the Tikhonov regularization in Table 5.

Table 5 Comparison of regularization

	TSVD	Tikhonov
Parameter	Discrete	Continuous
Computational complexity in classification stage	Low	High
Computational complexity in constructing stage	High	Low
Additive Learning	△	○

Parameters of KPCA and KRPCA is a rank of the operator. If the number of samples are finite, the rank of the operator is discrete.

(*1) : All the data sets are downloaded from 'http://ida.first.fraunhofer.de/projects/bench/benchmarks.htm'.

While regularization parameters of KSP and SKSP is continuous. If the problem is sensitive against these parameter, the Tikhonov regularization is better to control the degree of regularization. For example, if samples are quite few, since the ranks of KPCA or KRPCA equal to the number of samples, it is difficult to control the degree of the regularization.

Since KPCA and KRPCA can reduce the rank of operator, computational complexity in classification stage is lower than Tikhonov methods. While computational complexity of KPCA and KPCA in constructing stage is high because KPCA and KRPCA require eigenvalue decomposition in constructing stage. KSP and SKSP require the inverse operation which is lower computational complexity than the eigenvalue decomposition.

We can apply additive learning to KSP and SKSP easily by using Sherman-Morrison-Woodbury formula and its extension [50].

Proposition 6 (Additive learning of KSP). *Let* $\{f_1, \ldots, f_N, f_{N+1}\}$ *be samples,* $S = \sum_{i=1}^{N} \Phi(f_i) \otimes \overline{e_i}$, $S' = \sum_{i=1}^{N+1} \Phi(f_i) \otimes \overline{e'_i}$, *where* e_i *and* e'_j *are natural basis in* $\mathbb{R}^N$ *and* $\mathbb{R}^{N+1}$ *respectively. Suppose that an inverse matrix of kernel Gram matrix of* S, $K_S^{-1} = (S^*S)^{-1}$ *is known and new sample* f_{N+1} *is given. Then an inverse of new kernel Gram matrix* $K_{S'}^{-1}$ *is given as*

$$K_{S'}^{-1} = \begin{bmatrix} K_S^{-1} & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{bmatrix} + \frac{1}{\tau} t t^\top, \tag{28}$$

where

$$t = \begin{bmatrix} K_S^{-1} S^* \Phi(f_{N+1}) \\ -1 \end{bmatrix}$$

$$\tau = k(f_{N+1}, f_{N+1}) - \langle S^* \Phi(f_{N+1}), K_S^{-1} S^* \Phi(f_{N+1}) \rangle.$$

Furthermore we can introduce the Gaussian elimination to solve an inverse operation. The computational complexity of Gaussian elimination is extremely low. However we cannot store the inverse matrix. Thus it is useful in the case that the number of classification is low e.g., cross validation or leave one out.

9.2 Suppression of other classes effects

From the computational simulations and experiments in [35], SKSP and KRPCA show higher accuracy rate than KSP and KPCA. Thus suppression of effects of other classes improves the accuracy rate. However suppression increases computational complexity simultaneously. Using Sherman-Morrison-Woodbury formula, SKSP requires inverse operations of $|\Omega| \times |\Omega|$ matrix and $|\Psi| \times |\Psi|$ matrix, and several matrix products.

SKSP and KRPCA differ from ordinary binary classifiers e.g., SVM or Fisher discriminant, in that they not need to use all samples of other classes because KSP and KPCA can extract feature of other classes itself. In actual problems, KSP and KPCA can extract the almost enough features. Thus we do not have to use all samples of other classes. Only samples which are similar to Ω_i have to be included in Ψ_i. Since the similarity of an input vector is evaluated by the projection norm onto $\mathcal{R}(S_i)$ in KSP, it sufficient that samples of which projection norms are large are included in Ψ_i. For example, we can use following criterion to choose suppression set:

$$t(g_i) = \|P_{\mathcal{R}(S_i)} \Phi(g_i)\|^2, \tag{29}$$

where $P_{\mathcal{R}(S_i)}$ is an operator of KSP. In this case, samples which have large t should be used.

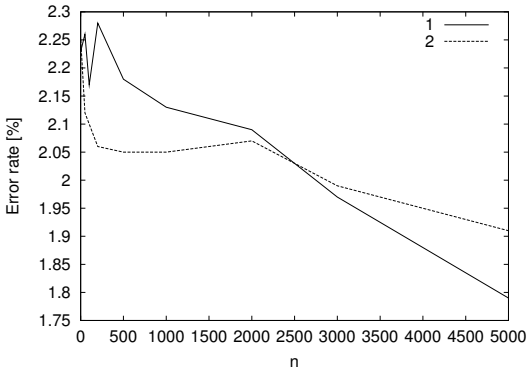


Figure 4 Error rate of SKSP: vertical axis: error rate [%], hirizontal axis: number of suppression samples

Unlike KSP and KPCA which are orthogonal projectors, SKSP and KRPCA are not orthogonal projector. Thus there are two kinds of discriminant function:

$$d_1(f_x) = \|X\Phi(f_x)\|^2 \tag{30}$$

$$d_2(f_x) = \|\Phi(f_x) - X\Phi(f_x)\|^2. \tag{31}$$

Figure 4 shows the relation between error rates of handwritten digit classification problem and the number of suppression samples. '1' and '2' mean eq. (30) and eq. (31) respectively. Suppressing samples are chosen by the criterion (29).

From the figure, when the number of suppression samples is small, discriminant function (31) shows better accuracy. While when the number of suppression samples is large, discriminant function (30) shows better accuracy. However there is no theoretical evidence that which is better.

9.3 Comparison with other classification method

The remarkable difference between SKSP and SVM or KFD is that SKSP is a quadratic discriminant function, while SVM and KFD are linear discriminant functions in the feature space. In the case of classification in the input space (not using a kernel method), generally quadratic classifiers shows higher performance than linear ones, because quadratic classifiers have more degree of freedom than linear ones. Thus if they are extended by kernel methods, quadratic discriminants will show higher performance in feature space.

In SVM, a separating hyperplane is determined by a few samples called support vectors (SVs). The separating hyperplane depends only on samples around boundary and does not depend on other samples or its distribution. If there are noisy samples or outliers in learning samples, a separating hyperplane is deteriorated by

them because they become SVs in high probability. Thus SVM is not robust fundamentally, even if the regularizing methods e.g. soft margin ([5]) or ν-SVM ([51]) are used.

In general, a classifier has trade off between robustness and sparseness about the number of samples. The computational cost of SVM in recognition stage is low because its solution is sparse. Let k and s be computational costs of calculating a kernel function and a multiplication respectively. Let L and L_{SV} be the number of learning samples and SVs respectively. Then main computational cost in recognition stage are given as

$$\begin{aligned} \text{SVM} &: (k+s)L_{SV} \\ \text{KFD} &: (k+s)L \\ \text{SKSP} &: sL^2 + (k+s)L. \end{aligned}$$

Note that generally, $s < k$ and $L_{SV} < L$. Only SKSP has a 2nd-order term with respect to L, because it is a quadratic discriminant in feature space. But as mentioned above, all samples belonging to other classes do not have to be included. Thus we can decrease L and computational cost.

In learning stage, SVM costs a lot of time because it requires to solve a quadratic optimization problem. KFD also requires to solve it ([52]). On the other hand the solution of SKSP is given as a closed form with an inverse operation and multiplications. Moreover as stated above, all samples belonging to other classes do not have to be used. The calculation steps for inverse of matrices and multiplication are $O(L^3)$. Generally, inverse problems are easier than quadratic optimization problems. Thus the computational cost of KSP or SKSP is lower than SVM in learning stage.

Moreover in the case that there are a large number of leaning samples, we can introduce "multi-templates method" to SKSP or KSP easily, because they are not a two-class classifier. In multi-templates method, sub-classes in a class is prepared and an input vector is classified to a sub-class. It is no use for two-class classifiers (e.g. SVM or KFD) to realize this method, because they require to use all samples of all classes.

10. Conclusion

We exposited a family of kernel subspace classifiers and their extensions. They show adequate performance in computational simulations. Especially in the case of multi-class classification problems, they will be useful because it is high computational complexity to construct binary classifiers.

However there are still open problems that are how to obtain the parameter, how to choose suppression samples, and discriminant function of SKSP or KRPCA.

This works have been suppoted by JSPS Grand-in-Aid for Scientific Research 18300057.

References

[1] V. N. Vapnik: "Statistical Learning Theory", Wiley, New-York (1998).

[2] V. N. Vapnik: "The Nature of Statistical Learning Theory", Springer (1995).

[3] B. Schökopf, C. J. C. Burges and A. J. Smola: "Advances in kernel methods: Support vector learning", MIT Press (1999).

[4] C. J. C. Burges: "A tutorial on support vector machines for pattern recognition", Data Mining and Knowledge Discovery, **2**, pp. 121–167 (1998).

[5] C. Cortes and V. Vapnik: "Support-Vector Networks", Machine Learning, **20**, 3, pp. 273–297 (1995).

[6] J. Mercer: "Functions of positive and negative type, and their connection with the theory of integral equations.", Trans. Lond. Phil. Soc. (A), **209**, pp. 415–446 (1909).

[7] B. Schölkopf, S. Mika, C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch and A. Smola: "Input space vs. feature space in kernel-based methods", IEEE Transactions on Neural Networks, **10**, 5, pp. 1000–1017 (1999).

[8] B. Schökopf and A. J. Smola: "Learning with kernels", MIT Press (2002).

[9] B. Schölkopf, A. Smola and K.-R. Müller: "Nonlinear component analysis as a kernel eigenvalue problem", Neural Computation, **10**, 5, pp. 1299–1319 (1998).

[10] S. Mika, B. Schölkopf, A. J. Smola, K.-R. Müller, M. Scholz and G. Rätsch: "Kernel PCA and de–noising in feature spaces", Advances in Neural Information Processing Systems 11 (Eds. by M. S. Kearns, S. A. Solla and D. A. Cohn), MIT Press (1999).

[11] S. Mika, G. Rätsch, J. Weston, B. Schölkopf and K.-R. Müller: "Fisher discriminant analysis with kernels", Neural Networks for Signal Processing IX (Eds. by Y.-H. Hu, J. Larsen, E. Wilson and S. Douglas), IEEE, pp. 41–48 (1999).

[12] S. Mika, A. J. Smola and B. Schölkopf: "An improved training algorithm for kernel fisher discriminants", Proceedings AISTATS 2001 (Eds. by T. Jaakkola and T. Richardson), Morgan Kaufmann, pp. 98–104 (2001).

[13] F. R. Bach and M. I. Jordan: "Kernel independent component analysis", Journal of Machine Learning Research, **3**, pp. 1–48 (2002).

[14] 鷲沢, 山下："カーネルウィーナーフィルタ", 第 6 回情報論的学習理論ワークショップ IBIS2003 予稿集, pp. 143–148 (2003).

[15] 鷲沢, 山下："カーネルウィーナーフィルタ", 電子情報通信学会技術研究報告, NC2003, pp. 73–78 (2004).

[16] Y. Washizawa and Y. Yamashita: "Non-linear Wiener filter in reproducing kernel Hilbert space", Proc. of 18th International Conference on Pattern Recognition (ICPR 2006) (2006 to appear).

[17] E. Oja: "Subspace methods of pattern recognition", Wiley, New-York (1983).

[18] S. Watanabe and N. Pakvasa: "Subspace method in pattern recognition", Proc. 1st Int. J. Conf on Pattern Recognition, Washington DC, pp. 25–32 (1973).

[19] H. Ogawa: "Karhunen-Loève subspace", Proc. 11th IAPR Int. Conf. Patt. Recogn., Vol. 2, The Hague, The Netherlands, pp. 75–78 (1992).

[20] K. Karhunen: "Ueber lineare methoden in der wahrscheinlichkeitsrechnung", Annalyes Academiae Scientiarum Fennicae, Series A1: Mathematica-Physica, **37**, pp. 3–79 (1947).

[21] M. Loève: "Probability Theory", Van Nostrand, New-York (1963).

[22] K. Pearson: "On lines and planes of closest fit to systems of points in space", Philosophical Magazine, **2**, 6, pp. 559–572 (1901).

[23] H. Hotelling: "Analysis of a complex of statistical variables into principal components", Journal of the American Statistical Association, **24**, p. 441 (1933).

[24] Y. Yamashita and H. Ogawa: "Relative Karhunen-Loève transform", IEEE Transactions on signal processing, **44**, 2, pp. 1031–1033 (1996).

[25] Y. Yamashita: "Optimum sampling vectors for Wiener filter noise reduction", IEEE Trans. of signal processing, **50**, 1, pp. 58–68 (2002).

[26] Y. Yamashita, Y. Ikeno and H. Ogawa: "Relative karhunen-loeve transform method for pattern recognition", Proc. of International Conference on Pattern Recognition (ICPR 1998), pp. 1007–1010 (1998).

[27] 池野, 山下, 小川："相対 KL 変換法によるパターン認識", 信学論

(D-II), **J80-D-II**, 2, pp. 541–547 (1997).
[28] 鷲沢, 疋田, 田中, 山下：“パターン認識のための相対 KL 変換法の高精度化”, 情報科学技術フォーラム 2002 一般講演論文集,I-48, pp. 95–96 (2002).
[29] 前田, 村瀬：“カーネル非線形部分空間法によるパターン認識”, 信学論 (D-II), **J82-D-II**, 4, pp. 600–612 (1999).
[30] 津田：“ヒルベルト空間における部分空間法”, 信学論 (D-II), **J82-D-II**, 4, pp. 592–599 (1999).
[31] K. Tsuda: “Subspace classifier in the Hilbert space”, Pattern Recognition Letters, **20**, 5, pp. 513–519 (1999).
[32] E. Maeda and H. Murase: “Multi-category classification by kernel based nonlinear subspace method”, IEEE International Conference On Acoustics, speech, and signal processing (ICASSP), Vol. 2, IEEE press., pp. 1025–1028 (1999).
[33] 疋田, 山下：“カーネル相対 KL 変換法によるパターン認識”, 信学技報, PRMU2001 (2002).
[34] 鷲沢, 疋田, 田中, 山下：“カーネル相対主成分分析による多クラスパターン認識”, 第二回 FIT（情報科学技術フォーラム）情報レターズ, pp. 207–208 (2003).
[35] Y. Washizawa, K. Hikida, T. Tanaka and Y. Yamashita: “Kernel relative principal component analysis for pattern recognition”, Proc. of Joint IAPR Iinternational Workshops on Syntactical and Structural Pattern Recognition and Statistical Pattern Recognition (SSPR/SPR 2004), pp. 1105–1113 (2004).
[36] 鷲沢, 山下：“カーネル標本空間射影法によるパターン認識”, 第 6 回情報論的学習理論ワークショップ IBIS2003 予稿集, pp. 149–154 (2003).
[37] Y. Washizawa and Y. Yamashita: “Kernel sample space projection classifier for pattern recognition”, 17th International Conference on Pattern Recognition (ICPR 2004), Vol. 2, pp. 435–438 (2004).
[38] 鷲沢, 山下：“抑制型カーネル標本空間射影法によるパターン認識”, 電子情報通信学会技術研究報告, iPRMU2003, pp. 85–90 (2004).
[39] Y. Washizawa and Y. Yamashita: “Kernel projection classifiers with suppressing features of other classes”, Neural Computation, **18**, 8, pp. 1932–1950 (2006).
[40] L. L. Scharf: “The SVD and reduced rank signal processing”, Signal Processing, **25**, 2, pp. 113–133 (1991).
[41] A. B. Israel and T. N. E. Greville: “Generalized Inverse: Theory and Applications”, John Wiley and Sons (1974).
[42] G. Rätsch, T. Onoda and K.-R. Müller: “Soft margins for AdaBoost”, Machine Learning, **42**, 3, pp. 287–320 (2001). also NeuroCOLT Technical Report NC-TR-1998-021.
[43] C. Bishop: “Neural Networks for Pattern Recognition”, Oxford Univ. Press (1995).
[44] C. W. Groetsch: “Inverse problems in the mathematical sciences”, Vieweg (1993).
[45] A. N. Tikhonov and V. Y. Arsenin: “Solution of Ill-posed problems”, V. H. Winston and Sons (1977).
[46] 武者, 岡本：“逆問題とその解き方”, オーム社 (1992).
[47] チャールズ W., 金子, 山本, 滝口：“数理科学における逆問題”, 別冊数理科学, サイエンス社 (1996).
[48] Y. LeCun, L. D. Jackel, L. Bottou, A. Brunot, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, U. A. Muller, E. Sackinger, P. Simard and V. Vapnik: “Comparison of learning algorithms for handwritten digit recognition”, International Conference on Artificial Neural Networks (Eds. by F. Fogelman and P. Gallinari), Paris, EC2 & Cie, pp. 53–60 (1995).
[49] G. Rätsch: “Robust Boosting via Convex Optimization”, PhD thesis, University of Potsdam, Neues Palais 10, 14469 Potsdam, Germany (2001).
[50] C. A. Rohde: “Generalized inverses of partitioned matrices”, Journal of Soc. Indust. Appl. Math., **13**, pp. 1033–1035 (1965).
[51] B. Schölkopf, P. Bartlett, B. Smola and R. Williamson: “Shrinking the tube: A new support vector regression algorithm”, Advances in Neural Information Processing Systems, Vol. 11, Cambridge, MA. MIT Press. (1999).
[52] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, A. Smola and K. Müller: “Invariant feature extraction and classification in kernel spaces”, Advances in Neural Information Processing Systems 12, MIT Press, pp. 526–532 (2000).

Appendix

Lemma 1 (Operator equation [41]). *Let $\mathcal{H}_1$, $\mathcal{H}_2$, $\mathcal{H}_3$ and $\mathcal{H}_4$ be Hilbert spaces and $A \in \mathcal{B}(\mathcal{H}_3, \mathcal{H}_4)$, $B \in \mathcal{B}(\mathcal{H}_1, \mathcal{H}_2)$, $C \in \mathcal{B}(\mathcal{H}_1, \mathcal{H}_4)$, where $\mathcal{B}(\mathcal{H}, \mathcal{H}')$ is bounded linear operator from $\mathcal{H}$ to $\mathcal{H}'$. Assume that $\mathcal{R}(A)$, $\mathcal{R}(B)$ and $\mathcal{R}(C)$ are closed. Then operator equation*

$$AXB = C \tag{32}$$

has a solution $X \in \mathcal{B}(\mathcal{H}_2, \mathcal{H}_3)$ when $\mathcal{R}(A) \supset \mathcal{R}(C)$ and $\mathcal{N}(B) \subset \mathcal{N}(C)$. A general form of a solution is given by

$$X = A^\dagger C B^\dagger + Y - A^\dagger A Y B B^\dagger, \tag{33}$$

where Y is an arbitrary operator in $\mathcal{B}(\mathcal{H}_2, \mathcal{H}_3)$.

Corollary 1. *Let $A \in \mathcal{B}(\mathcal{H}_1, \mathcal{H}_2)$, $B \in \mathcal{B}(\mathcal{H}_1, \mathcal{H}_3)$. If $\mathcal{R}(A)$ and $\mathcal{R}(B)$ are closed, an operator equation*

$$A = XB \tag{34}$$

has a solution when

$$\mathcal{N}(B) \subset \mathcal{N}(A). \tag{35}$$

Proofs of Theorems 3, 4 and 5

Here, we omit the symbol i for a class i for briefness. If we let $\alpha = 0$ in eq.(23), it is reduced to eq. (19). Thus we consider eqs. (22), (23).

Since $\|u - P_{\mathcal{R}(U)} v\| \leqq \|u - v\|$ for $^\forall u \in \mathcal{R}(U), ^\forall v \in \mathcal{H}$, X can be expressed as $X = UB$. From Corollary 1, a solution X in eqs. (19) and eqs. (23) can be expressed as $X = CU^*$. Then we can let

$$X = UAU^*, \tag{36}$$

where A is a real matrix of which size is $(|\Omega| + |\Psi|)$. Eq.(23) yields that

$$\begin{aligned}
J &= \frac{1}{|\Omega|}\sum_{s=1}^{|\Omega|} \|\Phi(f_s) - UAU^*\Phi(f_s)\|^2 \\
&\quad + \frac{1}{|\Psi|}\sum_{t=1}^{|\Psi|} \|UAU^*\Phi(g_t)\|^2 + \mu \|UAU^*\|_2^2 \\
&= \frac{1}{|\Omega|}\sum_{s=1}^{|\Omega|} \{k(f_s, f_s) - 2\langle \Phi(f_s), UAU^*\Phi(f_s)\rangle \\
&\quad + \langle UAU^*\Phi(f_s), UAU^*\Phi(f_s)\rangle\} \\
&\quad + \frac{1}{|\Psi|}\sum_{t=1}^{|\Psi|} \langle UAU^*\Phi(g_t), UAU^*\Phi(g_t)\rangle + \mu \mathrm{tr}[UA^*U^*UAU^*] \\
&= \frac{1}{|\Omega|}\sum_{s=1}^{|\Omega|} \{k(f_s, f_s) - 2\langle U^*\Phi(f_s), AU^*\Phi(f_s)\rangle \\
&\quad + \langle AU^*\Phi(f_s), K_U AU^*\Phi(f_s)\rangle\} \\
&\quad + \frac{1}{|\Psi|}\sum_{t=1}^{|\Psi|} \langle AU^*\Phi(g_t), K_U AU^*\Phi(g_t)\rangle + \mu \mathrm{tr}[A^* K_U A K_U].
\end{aligned}$$

Note that $U^*\Phi(f_s)$ and $U^*\Phi(g_t)$ are the s-th and $(|\Omega|+t)$-th column of K_U, respectively. Let $\tilde{D}=\Lambda^{-2}$. Then J is expressed as

$$\begin{aligned}J &= \frac{1}{|\Omega|}\mathrm{tr}[K_U\Lambda_0 - 2(K_U\Lambda_0)^*AK_U\Lambda_0 + (K_U\Lambda_0)^*A^*K_UAK_U\Lambda_0]\\ &\quad+\frac{1}{|\Psi|}\mathrm{tr}[(K_U\Lambda_0)^*A^*K_UAK_U\Lambda_0] + \mu\mathrm{tr}[A^*K_UAK_U]\\ &= \mathrm{tr}[\frac{1}{|\Omega|}K_U\Lambda_0 - \frac{2}{|\Omega|}\Lambda_0K_UAK_U\Lambda_0\\ &\quad+K_UA^*K_UAK_U\tilde{D}^{-1} + \mu A^*K_UAK_U], \end{aligned}\tag{37}$$

since $\Lambda_0=\Lambda_0^*$, $\tilde{D}^*=\tilde{D}$ and $K_U^*=K_U$.

(i) Solution of KRPCA

When $\mu=0$, J is reduced to

$$\begin{aligned}J &= \mathrm{tr}[(\Lambda K_UA^* - \Lambda\Lambda_0)K_U(AK_U\Lambda - \Lambda_0\Lambda)] + \psi\\ &= \|K_U^{1/2}(AK_U\Lambda - \Lambda_0\Lambda)\|_F^2 + \psi,\end{aligned}$$

where ψ is a term which is independent from A, $K_U^{1/2}$ is a square root matrix of K_U. Let singular value decomposition of $K_U^{1/2}\Lambda_0\Lambda$ is given as

$$K_U^{1/2}\Lambda_0\Lambda = \sum_{i=1}^{\gamma}\sqrt{\lambda_i}(u_i\otimes\overline{v_i}).\tag{38}$$

Then if a rank of $K_U^{1/2}\Lambda^{-1}K_UA^*$ is less than η, J is minimum when

$$K_U^{1/2}AK_U\Lambda = \sum_{i=1}^{\eta}\sqrt{\lambda_i}(u_i\otimes\overline{v_i}).\tag{39}$$

Since Lamma 1, and a property of singular value decomposition, $u_i=\frac{1}{\sqrt{\lambda_i}}(K_U^{1/2}\Lambda_0\Lambda)v_i$, one of the solutions is given as

$$\begin{aligned}A &= (K_U^{1/2})^\dagger K_U^{1/2}\Lambda_0\Lambda\sum_{i=1}^{\eta}(v_i\otimes\overline{v_i})\Lambda^{-1}K_U^\dagger\\ X &= UAU^*\\ &= U(K_U^{1/2})^\dagger K_U^{1/2}\Lambda_0\Lambda\sum_{i=1}^{\eta}(v_i\otimes\overline{v_i})\Lambda^{-1}K_U^\dagger U^*\end{aligned}\tag{40}$$

(ii) Solution of SKSP

The variation of J with respect to A in eq.(37) is given as

$$\begin{aligned}\delta J &= \mathrm{tr}[K_U(\delta A)^*K_UAK_U\tilde{D}^{-1} + K_UA^*K_U(\delta A)K_U\tilde{D}^{-1}\\ &\quad-\frac{2}{|\Omega|}\Lambda_0K_U(\delta A)K_U\Lambda_0 + \mu(\delta A)^*K_UAK_U + \mu(\delta A)K_UA^*K_U]\\ &= \mathrm{tr}[(\delta A)^*(K_UAK_U\tilde{D}^{-1}K_U - \frac{1}{|\Omega|}K_U\Lambda_0K_U + \mu K_UAK_U)\\ &\quad+(\delta A)(K_U\tilde{D}^{-1}K_UAK_U - \frac{1}{|\Omega|}K_U\Lambda_0K_U + \mu K_UA^*K_U)]\\ &= 2\mathrm{tr}[(\delta A)^*(K_UAK_U\tilde{D}^{-1}K_U - \frac{1}{|\Omega|}K_U\Lambda_0K_U + \mu K_UAK_U)]\end{aligned}\tag{41}$$

J is minimum when

$$K_UA(K_U+\mu\tilde{D})\tilde{D}^{-1}K_U = \frac{1}{|\Omega|}K_U\Lambda_0K_U.\tag{42}$$

In the case of $\mu>0$, from Lemma 1,

$$\begin{aligned}A(K_U+\mu\tilde{D})\tilde{D}^{-1} &= \frac{1}{|\Omega|}K_U^\dagger K_U\Lambda_0K_UK_U^\dagger + W - K_U^\dagger K_UWK_UK_U^\dagger\\ &= W + K_U^\dagger K_U(\frac{1}{|\Omega|}\Lambda_0 - W)K_UK_U^\dagger,\end{aligned}\tag{43}$$

where W is arbitrary operator. Let $W'=W-\frac{1}{|\Omega|}\Lambda_0$, it follows that

$$A(K_U+\mu\tilde{D})\tilde{D}^{-1} = \frac{1}{|\Omega|}\Lambda_0 + W' - K_U^\dagger K_UW'K_UK_U^\dagger.$$

Then we have

$$\begin{aligned}A &= \frac{1}{|\Omega|}\Lambda_0\tilde{D}(K_U+\mu\tilde{D})^{-1} + W'\tilde{D}(K_U+\mu\tilde{D})^{-1}\\ &\quad-K_U^\dagger K_UW'K_UK_U^\dagger\tilde{D}(K_U+\mu\tilde{D})^{-1}.\end{aligned}\tag{44}$$

Since $\frac{1}{|\Omega|}\Lambda_0\tilde{D}=\Lambda_0$, X which minimizes J is given as

$$\begin{aligned}X &= UAU^*\\ &= U\Lambda_0(K_U+\mu\tilde{D})^{-1}U^* + UW'\tilde{D}(K_U+\mu\tilde{D})^{-1}U^*\\ &\quad-UK_U^\dagger K_UW'K_UK_U^\dagger\tilde{D}(K_U+\mu\tilde{D})^{-1}U^*.\end{aligned}$$

Since $K_U=U^*U$, $\mathcal{R}(K_U)=\mathcal{R}(U^*)$. Then we have

$$\begin{aligned}K_UK_U^\dagger U^* &= U^*,\\ UK_U^\dagger K_U &= U.\end{aligned}$$

It is clear that

$$U^*(U\tilde{D}^{-1}U^*+\mu I) = (U^*U+\mu\tilde{D})\tilde{D}^{-1}U^*,$$

so that

$$(K_U+\mu\tilde{D})^{-1}U^* = \tilde{D}^{-1}U^*(U\tilde{D}^{-1}U^*+\mu I)^{-1}.\tag{45}$$

Then we have

$$\begin{aligned}K_UK_U^\dagger\tilde{D}(K_U+\mu\tilde{D})^{-1}U^* &= K_UK_U^\dagger U^*(U\tilde{D}^{-1}U^*+\mu I)^{-1}\\ &= U^*(U\tilde{D}^{-1}U^*+\mu I)^{-1}\\ &= \tilde{D}(K_U+\mu\tilde{D})^{-1}U^*.\end{aligned}$$

Hence, eq.(45) yields that

$$\begin{aligned}X &= U\Lambda_0(K_U+\mu\tilde{D})^{-1}U^*\\ &\quad+UW'\tilde{D}(K_U+\mu\tilde{D})^{-1}U^* - UW'\tilde{D}(K_U+\mu\tilde{D})^{-1}U^*\\ &= U\Lambda_0(K_U+\mu\tilde{D})^{-1}U^*.\end{aligned}\tag{46}$$

In the case of $\mu=0$, if K_U is nonsingular, eq.(42) yields

$$A = \Lambda_0K_U^{-1},\tag{47}$$

$$X = U\Lambda_0K_U^{-1}U^*.\tag{48}$$

Proof of Proposition 2

An operator A is a projector if and only if $AA=A$. Since eq. (40),

$$\begin{aligned}XX &= UAU^*UAU^*\\ &= UAK_UAU^*.\end{aligned}$$

If K_U is nonsingular,

$$\begin{aligned}XX &= U\Lambda_0\Lambda\sum_{i=1}^{\eta}(v_i\otimes\overline{v_i})\Lambda^{-1}K_U^{-1}K_UU^*\\ &= UAU^* = X.\end{aligned}$$

Proof of Proposition 3

If K_U is nonsingular and $\mu = 0$,

$$\begin{aligned}\tilde{P}^{\mu}_{\mathcal{R}(S)}\tilde{P}^{u}_{\mathcal{R}(S)} &= U\Lambda_0 K_U^{-1}U^* U\Lambda_0 K_U^{-1}U^* \\ &= U\Lambda_0 K_U^{-1}U^* = \tilde{P}^{u}_{\mathcal{R}(S)}.\end{aligned}$$

Proof of Proposition 4

If K_U is nonsingular and $\mu = 0$,

$$\tilde{P}^{\mu}_{\mathcal{R}(S)}P_{\mathcal{R}(U)} = U\Lambda_0 K_U^{-1}U^* UK_U^{-1}U^* = U\Lambda_0 K_U^{-1}U^* = \tilde{P}^{\mu}_{\mathcal{R}(S)},$$
$$P_{\mathcal{R}(U)}\tilde{P}^{\mu}_{\mathcal{R}(S)} = UK_U^{-1}U^* U\Lambda_0 K_U^{-1}U^* = U\Lambda_0 K_U^{-1}U^* = \tilde{P}^{\mu}_{\mathcal{R}(S)}.$$

Proof of Proposition 5

If K_U is nonsingular and $\mu = 0$, since $\mathcal{R}(T) = \mathcal{R}(UT)$

$$\tilde{P}^{\mu}_{\mathcal{R}(S)}UT = U\Lambda_0 K_U^{-1}U^* UT = U\Lambda_0 T = 0.$$

Subspace2006 部分空間法入門

黒沢　由明†

† 東芝ソリューション, 府中エンジニアリングセンター
〒 183-8532 東京都府中市武蔵台 1-1-15

あらまし　部分空間法は 1970 年前後に登場し，現在も使われているパターン認識の重要な基礎技術で，本稿はその基礎的な入門編である．本稿では，部分空間法，重み付き部分空間法，部分空間を求める手段である主成分分析，その主成分分析とベイズ識別などで用いられる主成分分析との関係，部分空間法とベイズ識別の関係，部分空間のもう 1 つの求め方である逐次最適化手法を紹介している．最後に研究課題として「部分空間の求め方」「類似度の構成法」「意外な対象，意外な適用」について述べている．

キーワード　部分空間，主成分分析，複合類似度，CLAFIC, 重み

The Engineers's Guide to the Subspace Method

Yoshiaki KUROSAWA†

† Toshiba Solutions Corporation, Fuchu Engineering Center
1–15, Musahidai, 1-Chome, Fuchu-shi, Tokyo, 183–8532 Japan

Abstract　The subspace method has been developed around 1970, and is now used as important basic technology. This report is written as a guide to this field. The subspace method itself, weighted subspace method, principal component analysis (PCA) as the tool for fixing a subspace, it's relationship to PCA used in bayes discrimination, the relationship between subspace method and bayes discrimination, and the iterative optimization method as another fixing method of subspace are introduced here. Finally, "how to fixing the subspace", "how to construct a similarity", "surprised object, surprised application" are described.

Key words　subspace, principle component analysis, multiple similarity, CLAFIC, weight

1. はじめに

本稿は部分空間法に関して，初心者向けに書いたもので，式の変形についても詳細に記述した資料である．なお，部分空間法を学ぶための文献としては [1] [2] がある．

本稿では，まず部分空間法の定義の説明，その最初の歴史について触れた後，重み付き部分空間法の説明を行い，部分空間の決め方の代表的な手法である主成分分析を，ベイズ識別などで用いられる通常の主成分分析と関連付けて説明している．次に，部分空間法とベイズ識別の関係を述べ，部分空間のもう 1 つの求め方である逐次最適化手法を紹介した後，最後に研究課題として 3 つの方向性について簡単な紹介を行っている．

部分空間法は 1970 年前後に登場し，現在まで，姿を替え，形を替え，様々に応用されて来ている．パターンを線形空間の点であると考える限り，パターンがなすクラスタを部分空間として捉えることは自然である．これは，より厳密には多様体なのであろうけれども，そこに至る前に部分空間として分析して行く考え方は自然である．

ニューラルネットによる認識系も結局は線形空間の中を区切る超曲面を構成していることを考えると，線形空間の中のベクトルの塊をどう分類していくか，区分けしていくかという事が認識の基本であるようにも思える．その意味でも部分空間法の考え方は重要である．

2. 部分空間法とは

部分空間法の定義は以下の通りである．パターン空間の次元，すなわちパターンベクトルの要素の数を N，正規直交ベクトルである辞書ベクトルを φ_i，辞書ベクトルの数を r， 入力ベクトルを x とする時，類似度を S を，

$$S = \sum_{i=0}^{r-1} (x, \varphi_i)^2. \tag{1}$$

で定義する．この式で辞書ベクトルはカテゴリごとに定義され，S もカテゴリごとに計算される．各カテゴリごとに計算された類似度について，最大類似度をとるカテゴリを認識結果とする．ここで r は φ_i の張る部分空間の次元で， N と同様，次

元と呼ばれるが，N と混同しやすいので注意が必要．次元 N はパターン空間全体の次元，r は部分空間の次元である．

φ_i, x が正規化されていない時は，

$$S = \sum_{i=0}^{r-1} (x, \varphi_i)^2 / \|x\|^2 \|\varphi_i\|^2. \tag{2}$$

と書き，こちらの方が一般的である．なお，式の計算やコーディング時に，式 (1) と式 (2) を混同すると間違いが起きやすいので注意が必要である．

この方式はカテゴリを代表する辞書として辞書ベクトルが張る部分空間を設定し，入力パターンのこの部分空間への正射影の長さをもって類似度とする方式である．式 (1) で $r = 1$ としたものは単純類似度と呼ばれる．プロジェクション P を

$$P = \sum_{i=0}^{r-1} \varphi_i \varphi_i^T. \tag{3}$$

で定義すれば，内積を $(a, b) = a^T b$ と書くことにより，

$$S = x^T P x. \tag{4}$$

とも記述できる．

部分空間法の意味は，線形空間内のパターン分布の広がりを部分空間として捉え，この部分空間でパターンを代表させる考え方である．これに対して，単純類似度 (部分空間法の 1 次元版) の考え方は，パターンの広がりとしては 1 次元の空間を考慮したものである．

この空間を決める手法としては主成分分析が代表的な手法で，学習パターン集合に対して，その平均的な射影成分が最大となる部分空間を得る手法である．この主成分分析の手法は部分空間法が考案された当初から導入されている [3] [7].

部分空間法は 1970 年前後にハワイ大の S. Watanabe によって発表されたものである [6] [7] [8] [9]．最初に，すべてのカテゴリのパターン全部を使って主成分分析を行うことによって特徴選択を行う認識システム：SELFIC [10] の提案があり，それに引き続いて部分空間法：CLAFIC の提案がなされた．

一方，国内では，1970 年以前から高速高性能な活字読取 OCR である ASPET70 および 71 が開発されていた．このプロジェクトの前半は国家プロジェクトとして，後半は電総研と東芝の共同開発として実施されたものである．この装置の認識システムとして，電総研の飯島により開発された複合類似度法 [3] [4] [5] が用いられていた．

この方式は活字文字認識に始まり現在に至るまで手書文字認識，漢字認識，音声認識等，様々な場面に応用されている．これは関数解析的な考え方から導かれたもので, マルチスケールの考え方を持つボケの理論や画像の性質を記述した方程式を含む理論体系で，その中の類似度計算部分が重み付き部分空間法と見なせるものである．また，実用面から言うと，重みが全て 1 の場合の複合類似度法の類似度計算は部分空間法のそれと同じであり，その意味で複合類似度の開発を部分空間法の最初の研究開発の 1 つであると考えることができる．

以上述べたように，1970 年前後にハワイ大と日本で部分空間法の研究開発がなされていたのである．

なお，複合類似度法の特徴抽出ではボカシを用いた濃度特徴が使われているが，これが開発された当時はボカシの重要性について，あまり気づかれていなかった．当時は文字はできるだけ鮮明に観測したほうが良いという考え方が主流であって，ボカシは認識性能に悪影響があると考えられていた．しかしながら，その後，特に統計的手法ではボカシの重要性が認められるようになってきて，特徴をボカす方式が多く使われるようになって来ただけでなく，文字認識以外でもその重要性が認められている．

3. 重み付き部分空間法とは

部分空間法の重要な拡張の方法の 1 つとして重み付き部分空間法がある．ここでは，それについて説明していく．

重み付き部分空間法の定義は，μ_i を重みとして，

$$S = \sum_{i=0}^{r-1} \mu_i (x, \varphi_i)^2, \tag{5}$$

で定義する．この重みの与え方で認識系の性質が変わって来る．

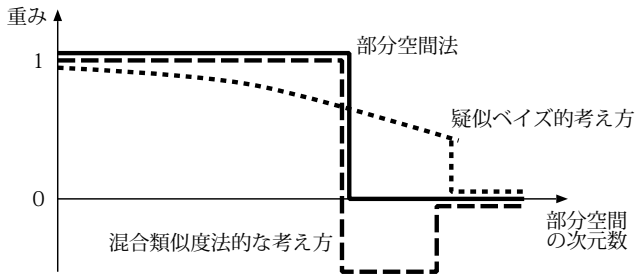


図 1　重み付き部分空間法の重みの例

Fig. 1　Example of weight in weighted subspace method.

図 1 の「部分空間の次元数」の軸上の番号は主成分分析で求めた固有ベクトルの番号であり，この番号は対応する固有値の大きいものから順に 0 オリジンの整数で付けられているとする．この図に示す通り，部分空間法では，ある次元数まで重み 1 で，そこから後は 0 となる．一方，疑似ベイズ的な考え方に対応する重み付き部分空間法の例では，重み 1 を起点として，ある次元数まで重みがダラダラと下がり，そこから後が 0 となる．また，混合類似度法のような考え方 [11] では，ある次元で重みがマイナスになるように設定されている．なお，このようなマイナスの重みを付ける場合には，ベクトルの番号の振り方は，必ずしも固有値の大きい順とはならない．

疑似ベイズのように固有値番号が上がるに従って，重みが下がるというのは，固有値の小さい固有ベクトルの成分については，類似度に対する寄与度を下げるという考え方である．

混合類似度的な考え方とは，認識誤りしやすいカテゴリのパターンを代表する部分空間を，正解のカテゴリを代表する部分空間に直交するように作成して，その空間の正規直交基底も辞

書ベクトルに加え，その空間への射影の長さを類似度から引くように設定したものである．

なお，疑似ベイズ識別は部分空間法では無いが，その関係については後で述べる．

4. 主成分分析

部分空間法では，辞書ベクトルの求め方が問題となるが，一般に部分空間法という言葉は主成分分析を用いて固有ベクトルを求め，これをもって辞書ベクトルとする手法のことを意味する場合が多い．

通常の主成分分析は平均ベクトルを中心にして分布の主成分を探す方式を指すが，部分空間法の辞書ベクトル φ_i を求めるのに用いる主成分分析は平均ベクトルを用いない方式で，通常の主成分分析とは若干異なる．

ここでは，最初に通常の主成分分析について説明する．

以下，主成分分析 (Principal Component Analysis : PCA) の考え方を簡単に紹介する．

学習対象ベクトルの 1 つ 1 つに名前を付けて，それを α と書くこととし，学習ベクトルを x_α と記述する．その平均ベクトルを m とする．この時，$x_\alpha - m$ のベクトル集合の方向をもっとも良く表す単位ベクトルを φ_0 とする．ここで言う「ベクトル集合の方向をもっとも良く表す」という事を，数学的には「$x_\alpha - m$ と φ_0 の内積の自乗和が最大になる事」と考えることにする．$\|\varphi_0\| = 1$ の条件が付くので，この条件を含める為に未定乗数 λ_0 を使って，以下のように評価式を設定する．この評価式の J を最大にする φ_0 を求めるのが主成分分析である．

$$J = \sum_\alpha (x_\alpha - m, \varphi_0)^2 - \lambda(\|\varphi_0\|^2 - 1)$$

$$= \sum_\alpha \{\sum_i (x_{\alpha i} - m_i)\varphi_{0i}\}^2 - \lambda_0(\sum_i \varphi_{0i}^2 - 1). \qquad (6)$$

この式から φ_0 を求めるために，J を φ_0 の要素 φ_{0i} で偏微分し，それが 0 になるとして φ_{i0} を求める．

$$\frac{\partial J}{\partial \varphi_{0i}} =$$

$$2\sum_\alpha (x_\alpha - m, \varphi_0)(x_{\alpha i} - m_i) - \lambda_0(2\varphi_{0i}) = 0. \qquad (7)$$

この式で，φ_{0i}，m_i と $x_{\alpha i}$ を縦に並べて φ_0，m と x_α を作ることによって以下の式が得られる．

$$2\sum_\alpha (x_\alpha - m, \varphi_0)(x_\alpha - m) - \lambda_0(2\varphi_0) = 0. \qquad (8)$$

これは，さらに次のように書き換えられる．

$$\sum_\alpha (x_\alpha - m)(x_\alpha - m)^T \varphi_0 = \lambda_0 \varphi_0. \qquad (9)$$

$$K = \sum_\alpha (x_\alpha - m)(x_\alpha - m)^T, \qquad (10)$$

と書くことにより，式 (9) は

$$K\varphi_0 = \lambda_0 \varphi_0, \qquad (11)$$

となる．すなわち φ_0 は共分散行列 K の固有ベクトルとなる．固有ベクトルは次元数分，存在し，それぞれが直交する．

5. 超接平面上の主成分分析

部分空間法における主成分分析は

$$J = \sum_\alpha {x_\alpha}^T P x_\alpha, \qquad (12)$$

を最大化する P を求める事であるが，

$$P = \sum_{i=0}^{r-1} \varphi_i {\varphi_i}^T, \quad \|\varphi_i\|^2 = 1, \qquad (13)$$

なので，未定定数 λ_i を導入して，

$$J = \sum_\alpha \sum_{i=0}^{r-1} \{(x_\alpha, \varphi_i) - \lambda_i(\|\varphi_i\|^2 - 1)\}, \qquad (14)$$

を最大にする φ_i を求めるという事でもある．しかしながら，このような説明では一般的な主成分分析との関係が分からない．そこで以下，その関係が分かるような説明を試みる．

部分空間法では入力パターンはノルムが 1 に固定されているのと等価であるから，パターンは超球面上に分布していると考えて良い．ここでは簡単のためにこれを近似的に，超球面では無く単位ベクトル e に直交し，球面に接する超接平面上に分布しているとして議論を進める．図 2 にその様子を示す．

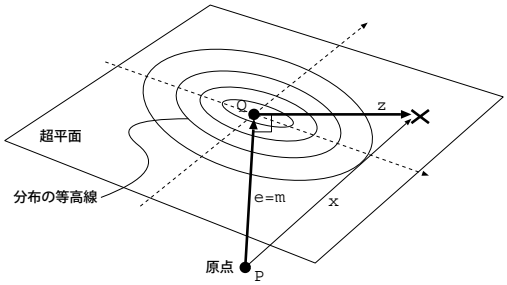


図 2　球面に接する超平面上でのパターン分布の例

Fig. 2　Example of pattern distribution on hyper plane which is tangential to spherical surface.

分布が e の終点 Q を中心とするガウス分布であれば，平均ベクトル m は e に等しくなる．その理由を以下に説明する．超平面上で Q を始点とするベクトル z_α がガウス分布をしているので，それらのベクトルを原点を中心としたベクトル x_α 書き直すと，$x_\alpha = e + z_\alpha$ となる．したがって，この近似を行うと，x_α のノルムは 1 に近いが 1 で無くなる．x_α の平均ベクトルを m と書くと，$\overline{z_\alpha} = 0$ なので，$m = \overline{x_\alpha} = e$ となる．以下，e を用いずに m を用いる．

この超平面内で平均ベクトルを考慮する通常の主成分分析を行うと，平均ベクトルは超平面の原点に等しいので，主成分は

$$K = \sum_{\alpha} z_\alpha z_\alpha^T, \tag{15}$$

の固有ベクトルとなる．

次に，部分空間法で使われる主成分分析での固有ベクトルについて調べることにする．ここで，以下の式の固有ベクトルを考えてみよう．これは部分空間法の主成分分析で使われる共分散行列である．

$$K = \sum_{\alpha} x_\alpha x_\alpha^T. \tag{16}$$

この式は分布の平均ベクトルが m であるにもかかわらず，共分散行列における平均ベクトルを強制的に 0 にした式である．平均ベクトル m と式 (15) の固有ベクトルが，式 (16) の固有ベクトルと同じであることを次に見て行く．

（性質 1）

ベクトル m は式 (16) の固有ベクトルになる．

（証明）

超平面と m は直交するので $(z_\alpha, m) = 0$ である．また，$x_\alpha = m + z_\alpha$，$\|m\| = \|e\| = 1$ である．

$$\sum_{\alpha} x_\alpha x_\alpha^T m = \sum_{\alpha} x_\alpha (x_\alpha, m)$$

$$= \sum_{\alpha} x_\alpha (m, m) = \sum_{\alpha} x_\alpha = m, \tag{17}$$

なので，m は式 (16) の固有ベクトルになる．（証明終り）

（性質 2）

式 (15) から求められる固有ベクトルを ψ_i とすると， ψ_i は式 (16) の固有ベクトルとなる．

（証明）

ベクトル ψ_i は式 (15) の固有ベクトルであるから，

$$\sum_{\alpha} z_\alpha z_\alpha^T \psi_i = \lambda_i \psi_i, \tag{18}$$

である．また，超平面は m と直交するので超平面内のベクトル ψ_i は m と直交する．ガウス分布の平均ベクトルは超平面の原点なので，$\sum_\alpha z_\alpha = 0$ である．

$$\sum_{\alpha} x_\alpha x_\alpha^T \psi_i = \sum_{\alpha} x_\alpha (x_\alpha, \psi_i)$$

$$= \sum_{\alpha} (m + z_\alpha)(m + z_\alpha, \psi_i)$$

$$= \sum_{\alpha} m(z_\alpha, \psi_i) + \sum_{\alpha} z_\alpha (z_\alpha, \psi_i)$$

$$= \sum_{\alpha} z_\alpha z_\alpha^T \psi_i = \lambda_i \psi_i. \tag{19}$$

ゆえに，ψ_i は式 (16) の固有ベクトルになり，その固有値は λ_i である．（証明終り）

このことから，部分空間法の辞書ベクトルとして式 (16) の固有値を φ_i として，この φ_i を辞書として採用することは式 (15) における m と ψ_i を採用することと同じになる．なお，厳密には $m \neq \varphi_0$，$\|x_\alpha\| \neq 1$ なので，上記のモデルは部分空間法とは異るが，近似的に部分空間法における主成分分析を理解する上で役立つと思う．

式 (16) の K は相関行列，能率行列，モーメント行列と呼ばれるものであるが，混同して共分散行列と呼ぶことも多い．本稿では特に問題ない限り，これも共分散行列と呼ぶことにする．

6. 複合類似度

複合類似度では，主成分分析の結果得られる r 個の固有ベクトルをそのカテゴリの辞書とする．この固有ベクトルを φ_i，固有値を λ_i とした時，複合類似度 S は次式で定義される．

$$S = \sum_{i=0}^{r-1} \frac{\lambda_i}{\lambda_0} \cdot \frac{(x, \varphi_i)^2}{\|x\|^2 \cdot \|\varphi_i\|^2}. \tag{20}$$

正確には複合類似度は式 (20) の $\sqrt{S}$ で定義されている．

この式で r を 全体空間の次元 N とすると式 (20) は，複合類似度法では特性核と呼ばれる共分散行列が

$$K = \sum_{\alpha} x_\alpha x_\alpha^T = \sum_{i=0}^{N-1} \lambda_i \varphi_i {\varphi_i}^T, \tag{21}$$

と書けるので，

$$S = \frac{1}{\lambda_0} x^T K x = \frac{1}{\lambda_0} \sum_{\alpha} (x_\alpha, x)^2 \tag{22}$$

となり，これは式 (4) に対応する．

式 (22) の意味は類似度を入力パターンと学習パターンの内積の自乗の平均で定義しているということで，学習パターンの集合と入力パターンの類似性を計る尺度としては妥当性のある方法であることが分かる．

一般には式 (20) が複合類似度と呼ばれるが，複合類似度法といった場合には，これまでに述べたようにボカシの考え方や画像に関する方程式などを含む理論体系全体を意味する．

7. 正 準 化

複合類似度法では正準化と呼ばれる手法が用いられる．これはパターンの直流成分を取り除いて認識を行おうとするもので，入力パターン・ベクトルの要素を表すビット数が少ない場合，その解像度を有効に使うための手段である．一般に認識精度に対しては良い影響があると予想されるが，厳密にどのような影響があるかは分かっていない．ベクトルの要素が全て 1 のものを $\vec{1}$ と表記し，ベクトルの次元を N として，

$$x' = x - \frac{1}{N}(x, \vec{1})\vec{1}, \tag{23}$$

によって x を x' に変換する操作を正準化と言う．ベクトル $\vec{1}$ の代わりに認識対象カテゴリ全ての平均パターンを用いる考え方もある．なお，正準化したパターンで作った共分散行列の固有ベクトルは正準化されている．

8. ベイズ識別

ベイズ識別は部分空間法では無いが，関係が深いので，その関係について後で説明する．その為の準備として，ここではベイズ識別と疑似ベイズ識別の説明を行う．

まず，カテゴリ A とカテゴリ B を識別するケースを考える．カテゴリ A のパターン・ベクトル x を集めて，x の出現頻度を調べると，x の頻度分布が作れる．この頻度分布を A のパターン数で割ったものは，カテゴリ A の文字が入力された時に，そのパターン・ベクトルが x である確率で $P(x|A)$ である．これは尤度と呼ばれる．ベイズの定理によると，

$$P(A|x) = \frac{P(x|A)P(A)}{P(x)}, \tag{24}$$

によって事後確率 $P(A|x)$ が求まる．なお，事前確率は $P(A)$ である．この $P(A|x)$ をカテゴリ B の事後確率 $P(B|x)$ と比較して大きい方を答えとするのがベイズ識別の基本的な考え方である．ここでは $P(A)$ をカテゴリによらず一定とし，また比較の際は $P(x)$ は共通なので，尤度 $P(x|A)$ を事後確率とみなして認識系を構築することとする．

パターン認識の分野でベイズ識別と言う場合には，パターン分布はガウス分布と仮定されることが多い．すなわち，パターン x の出現確率，すなわち尤度が次の式で定義される．次元を N として，

$$P(x) = \frac{1}{\sqrt{(2\pi)^N |K|}} exp\{-\frac{1}{2}(x-m)^T K^{-1}(x-m)\}. \tag{25}$$

ここで，$(x-m)^T K^{-1}(x-m)$ について説明しておく．これは 2 次形式なので，2 次元のケースで考えると理解しやすい．簡単のため，$m = 0$ として，$x = (x_0, x_1)^T$ と表記し，

$$K^{-1} = \begin{bmatrix} a & c \\ c & b \end{bmatrix}, \tag{26}$$

と置くと，

$$x^T K^{-1} x = ax_0^2 + 2cx_0x_1 + bx_1^2, \tag{27}$$

となる．これで，式 (25) の等高線が楕円になることが分かる．

式 (25) における m は平均ベクトル，K はパターン分布の共分散行列である．ここで，この m と K は各カテゴリごとに定義されるものであり，入力ベクトル x に対して 各カテゴリごとに $P(x)$ が求まる．この $P(x)$ が最大となるカテゴリを答えとするものがベイズ識別である．

カテゴリが 2 個の場合に 2 次元平面上にパターンが分布する様子を図 3 に示す．カテゴリ A のパターン分布，すなわちパターン・ベクトルを x とした時にそれが A である確率を $P(x)$，カテゴリ B である確率を $Q(x)$ とした時に，$P(x) > Q(x)$ が入力パターンがカテゴリ A であると判断されるエリア，$P(x) < Q(x)$ が入力パターンがカテゴリ B であると判断されるエリアとなる．

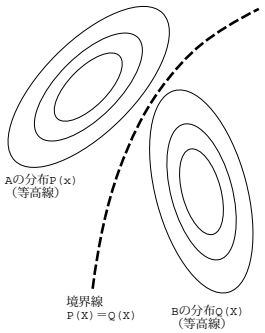


図 3　ベイズ識別での 2 カテゴリ識別

Fig. 3　Tow category discrimination by bayes discriminant function.

$P(x) = Q(x)$ がその中間の境界線である．

さて，式 (25) の両辺の log を取って右辺を若干変更することにより，

$$-2logP(x) - Nlog(2\pi) =$$

$$(x-m)^T K^{-1}(x-m) + log|K|, \tag{28}$$

となり，認識処理としてはこの右辺が最小となるカテゴリを求めれば良いことになる．この式で，後半の $log|K|$ を落としたものはマハラノビス距離と呼ばれる．次に φ_i を K の固有ベクトル，λ_i を固有値，固有ベクトルを番号順に並べた行列を Φ とした時，K を固有ベクトル展開して，

$$K = \Phi \begin{bmatrix} \lambda_0 & 0 & ... & 0 \\ 0 & \lambda_0 & ... & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & ... & \lambda_{N-1} \end{bmatrix} \Phi^T, \tag{29}$$

と置くと，式 (28) の右辺を相違度 D として，

$$D = \sum_{i=0}^{N-1} \frac{1}{\lambda_i}\{(x-m, \varphi_i)^2\} + log(\prod_{i=0}^{N-1} \lambda_i), \tag{30}$$

が得られる．ベイズ識別における実際の計算は，この式を相違度として用いて行われる．もちろん，マハラノビス距離はこの式で log を落した形式のものである．

9. 疑似ベイズ識別

ベイズ識別では，式 (30) からも分かるように，固有値 λ_i の逆数が右辺の計算に使われる．これを重みとして見ると，重みは $1/\lambda_0$ を起点に右肩上がりに上昇して行く．これは入力パターンの中に，主要成分で無いところがどれだけあるかをパターン空間全次元に渡って見る考え方である．このような考え方のものは、高い次元の固有値が極端に低くて，その次元での要素がノイズばかりというシチュエーションには向いていない．なぜなら，λ_i の値が小さい所で相違度が不安定になるからである．

このため，ある r よりも大きい λ_i を強制的に一定として式 (30) を計算する方式が疑似ベイズ識別である [15] [16] .

一定にした固有値を δ として，式 (30) を変形する．具体的には，式 (30) において番号が r 以上の固有値をすべて δ に置き換えると以下のように変形できる．

$$
\begin{aligned}
D &= \sum_{i=0}^{r-1}\frac{1}{\lambda_i}\{(x-m,\varphi_i)^2\}+\sum_{i=r}^{N-1}\frac{1}{\delta}\{(x-m,\varphi_i)^2\}\\
&+log(\prod_{i=0}^{r-1}\lambda_i)+log(\prod_{i=r}^{N-1}\delta)\\
&=\sum_{i=0}^{r-1}\frac{1}{\lambda_i}\{(x-m,\varphi_i)^2\}\\
&+\sum_{i=0}^{N-1}\frac{1}{\delta}\{(x-m,\varphi_i)^2\}-\sum_{i=0}^{r-1}\frac{1}{\delta}\{(x-m,\varphi_i)^2\}\\
&+log(\prod_{i=0}^{r-1}\lambda_i)+log(\prod_{i=r}^{N-1}\delta)\\
&=\frac{1}{\delta}\|x-m\|^2-\sum_{i=0}^{r-1}(\frac{1}{\delta}-\frac{1}{\lambda_i})\{(x-m,\varphi_i)^2\}\\
&+log(\prod_{i=0}^{r-1}\lambda_i)+log(\prod_{i=r}^{N-1}\delta)\\
&=\frac{1}{\delta}\{\|x-m\|^2-\sum_{i=0}^{r-1}(1-\frac{\delta}{\lambda_i})\{(x-m,\varphi_i)^2\}\}\\
&+log(\prod_{i=0}^{r-1}\lambda_i)+log(\prod_{i=r}^{N-1}\delta), \qquad (31)
\end{aligned}
$$

先頭の $1/\delta$ はどのカテゴリにも共通であるので，全体に δ を掛ける事として，最終的に以下のようにして相違度が求まる．

$$
\begin{aligned}
D &= \|x-m\|^2-\sum_{i=0}^{r-1}(1-\frac{\delta}{\lambda_i})(x-m,\varphi_i)^2\\
&+\delta\left\{log(\prod_{i=0}^{r-1}\lambda_i)+log(\prod_{i=r}^{N-1}\delta^2)\right\}. \qquad (32)
\end{aligned}
$$

これが疑似ベイズの相違度である．

これは特徴抽出法の 1 つである「加重方向指数ヒストグラム法」と共に三重大から提案された方式で，両者ともに実際に良く使われている方式である．

10. 投 影 距 離

部分空間法と似た方式に投影距離がある．平均ベクトルを m として，

$$
d = \|x-m\|^2-\sum_{i=0}^{r-1}(x-m,\varphi_i)^2, \qquad (33)
$$

で定義される．これは，部分空間法とは違って，ノルムに制限は無く，辞書ベクトル $\{\varphi_i\}$ で構成される部分空間に含まれない部分について， x と m の距離を計るものである．この方式でも φ_i は主成分分析によって求める場合が多い．

11. 内積型と距離型

ここでは文献 [17] に従って，疑似マハラノビスに対応する重み付き部分空間法を示し，これを疑似球面マハラノビスと呼ぶことにする．これはパターンが球面上に分布しているとして，ベイズ識別と同様な考え方から求められた類似度で，一部の項を除けば重み付き部分空間法の形式となっている．

$$
S = 2\delta\ln(x,\varphi_0)+(x,\varphi_0)^2+\sum_{i=1}^{r-1}(1-\frac{\delta}{\lambda_i})(x,\varphi_i)^2. \qquad (34)
$$

これに対して，通常のガウス分布を対象として求められた疑似ベイズ識別の log の項を省略した形式の疑似マハラノビスの式を次に示す．

$$
D = \|x-m\|^2-\sum_{i=0}^{r-1}(1-\frac{\delta}{\lambda_i})(x-m,\varphi_i)^2, \qquad (35)
$$

この 2 つの式の各固有ベクトルの重みの形が同じであることに注目して欲しい．疑似球面マハラノビスの φ_0 は疑似マハラノビスの m に対応している．

このように，疑似球面マハラノビスが疑似マハラノビスに対応しているのであるが，同様に，疑似ベイズに球面疑似ベイズ，投影距離に部分空間法，ユークリッド距離に単純類似度が対応付いている．

疑似ベイズ系はユークリッド距離ベースの相違度であり，球面系は内積ベースの類似度である．各式における計算方法に着目すると，前者は距離型，後者は内積型となるのである．表 1 にその関係を整理して示す．

表 1　距離型と内積型の対応関係

Table 1　Correspondence between Euclid distance type and inner product type.

距離型	内積型
疑似ベイズ	疑似球面ベイズ
疑似マハラノビス	疑似球面マハラノビス
投影距離	部分空間法
ユークリッド距離	単純類似度

これらの認識性能に関しては，表 1 の横に並んでいる方式同士は良く似た傾向を持つが，縦の関係で比較するとかなり違っている．内積型と距離型では，その重み係数が同じであれば，ほぼ同じような認識性能，認識傾向を持っている．従って，重要なのは距離型か内積型かでは無く，その重み係数をどう設定するかである．

12. 逐次最適化手法で部分空間を求める

ここでは、まず LVQ からスタートして、最後に部分空間の学習法について述べることとする．コホーネンによって提案された LVQ [18] は以下のような学習の枠組である．

入力パターンを x， 辞書ベクトルを m，相違度を D として，

$$
D = ||x-m||^2, \qquad (36)
$$

$$m = m \pm \alpha(x - m), \quad (+ : correct, - : error), \tag{37}$$

で m を更新して行く．この方式は確率降下法 [19] [20] によって導き出される方式でもある．LVQ はパターン認識分野で使われる逐次学習型のベーシックな手法の 1 つであり，しばしばニューラルネットの仲間とされて，そう呼ばれることが多い．一方，確率降下法は様々な認識系に適用できる逐次学習の有力な手段の 1 つであり，LVQ はユークリッド距離に確率降下法を適用したものである．次に，これを単純類似度に適用して見よう．

$$S = (x, \varphi)^2 / \|x\| \|\varphi\|, \tag{38}$$

で，類似度を定義すると，φ の更新式は以下のようになる．

$$\varphi = \varphi \pm \alpha(x, \varphi)\{x - (x, \varphi)\varphi\}, \tag{39}$$

この式は複雑なので，正規化の項を無視した形に確率降下法を適用して見よう．

$$S = (x, \varphi)^2, \tag{40}$$

を類似度とすると，

$$\varphi = \varphi \pm \alpha(x, \varphi)x, \tag{41}$$

と更新式が求まる．

この式では，x, φ は正規化されていることが前提となっているので，実は，その条件を入れずに確率降下法を適用した式 (41) は誤りである．従って，この場合，更新式に何らかの形で正規化の条件を入れなければならない．一番，単純な方法は更新する度に φ を正規化する手段である．式 (39) の代わりに，そういう手段を取ることが可能である．そもそも，式 (39) といえども，計算誤差が累積する可能性が高いので，更新後の正規化は必要となる．

次に，この話を部分空間法に拡張する．

$$S = \sum_{i=0}^{r-1} (x, \varphi_i)^2, \tag{42}$$

で類似度を定義すると，φ_i の更新式は式 (41) と同じになる．しかし，この時 φ_i の正規直交性は保証されていないので，正規直交性の条件を含めて式 (42) に確率降下法を適用しないといけない．しかしながら，この計算はかなり複雑になるので，単純類似度のケースで説明したのと同じように各更新ごとに φ_i の正規直交化を行う方法がある．その方法としてシュミットの直交化を使ったものは，コホーネンによって提案された LSM [21] と同じである．

LSM に似た方式に ALSM [22] [23] がある．これは，式 (16) の能率行列 K に対して

$$K = K \pm \alpha x x^T, \tag{43}$$

で更新する方式である．これは，1980 年に前田が特許として提案した方式 (学習型複合類似度法) [24] と同じものである．

LSM を確率降下法の立場で見ると，正規直交性を考慮せずに確率降下法を適用した形になっているが，これを考慮した方式の 1 つが DMD [25] である．これは部分空間法を定義しているプロジェクションの回転パラメータを更新パラメータとする方式である．

これ以外にも正規直交性に考慮した確率降下法の適用方法が検討されている [26].

13. 部分空間の求め方

13.1 独立成分分析

主成分分析と逐次最適化手法が部分空間法における部分空間の求め方の代表的なものであるが，これ以外に，最近では独立成分分析法が取り上げられるようになった．この手法は辞書として作られるベクトルの直交性を要請するものでは無く，分布密度の濃い方向を辞書ベクトルとして抽出する考え方のものである．

13.2 特 徴 選 択

特徴ベクトルから重要な特徴を選別する特徴選択の考え方は SELFIC が提案された時から研究が継続しており [12] [13] [14]，現在まで研究対象として取り上げられている．この特徴選択も部分空間法の前段部分として重要である．

14. 類似度の構成法

14.1 相互部分空間法

部分空間法がベクトルと部分空間のなす角度を測定しているのに対して，相互部分空間法は部分空間と部分空間のなす角度を測定するもので，1985 年に前田 [27] によって提案された．この方式は動画像として得られる顔画像の認識に力を発揮した [28]．動画像として入力される画像は 1 枚で無く複数枚であり，かつ様々に変動している．このような入力パターンを扱うのに，入力パターンを部分空間として考えるのは有力な手段であり，実際にその有効性が実験的に実証されている．また，最近では顔以外の動画像系の課題にも適用され，様々な分野での応用が期待されている．

14.2 カーネルトリックの導入

SVM の登場で，カーネルトリックという技術が注目されている．カーネルトリックとは，要するに元の次元から超高次元への変換を行い，超高次元空間で認識することにより，識別率を上げていこうという考え方である．低次元よりも高次元の方がパターンはバラバラで認識しやすかろうという考え方である．これは特徴選択などで特徴の次元を落していく従来型の考え方と逆のアプローチである．この考え方も当然，部分空間法に適用でき，その相乗効果が期待される．

14.3 混合類似度的な考え方

混合類似度 [11] そのものの説明は他に譲るとするが，その考え方は，類似している文字 A,B があった時，A の部分空間への射影はポジティブに，B の部分空間への射影はネガティブに扱うことによって，類似文字識別の精度を上げようとするものである．様々なバリエーションが考えられ，文字に限らず，類

似部分の識別に効果があると期待される．

15. 意外な対象，意外な適用

15.1 アイゲン・フェース

アイゲン・フェース [29] は顔画像の固有ベクトルを用いて，顔画像認識を行おうとする手法である．当時，顔認識の通常のアプローチは顔の部品をそれぞれ抽出して，その形や配置を分析して行こうという考え方が多かった．顔画像の中の構造を考えずにそのまま線形空間のベクトルとして捉え，主成分分析を行う考え方は，私にとっては，ずいぶん大胆な考え方であった．このアイゲンフェースの話を聞いた時には，部分空間法が顔まで使えるとは思っていなかったが，その後の展開を見ると部分空間法の力のあるところを感じる．

15.2 パラメトリック固有空間法

パラメトリック固有空間法 [30] は連続する動画像の 1 枚の絵を線形空間上の点として捉え，動画全体を線形空間上の点列と見て解析を進めて行こうという考え方である．通常，複雑な形状を持つ物体の動画像を解析して行こうという時，その形状の分析を行って，特徴点を出して，それをトラッキングしていくようなアプローチが取られる．これに対して，この手法は画像全部をベクトルと見なして，線形空間で見ることによって分析していくというアプローチに可能性，有効性がある事を示したものである．

顔への適用で驚いていた私は，この話を聞いてもっと驚くことになった．部分空間法的なセンスというものはパターン認識においてかなり重要なアイテムである．

16. おわりに

部分空間法のオリジナルの考え方そのものは 40 年近くの時を経た現在では，さすがにその認識能力が高いとは言えない．しかしながら，部分空間法の考え方をベースにした様々な新しいアイディアはいろいろな分野で活躍している．ユークリッド距離や内積，ベクトルの成す角度，ガウス分布などはパターン認識分野における重要な基礎概念であると考えるけれども，部分空間もまた重要な基礎概念の 1 つであると思う．ユークリッド距離や内積に比べると複雑な分だけ部分空間は，研究対象としての課題が数多くあるのだと思う．

この部分空間法の発展の方向として，本稿では「部分空間の求め方」,「類似度の構成法」,「意外な対象，意外な適用」と 3 つに分けて簡単のその内容を紹介した．部分空間法は，今後もパターン認識の分野で重要な位置を占めて行くと思う．

文　献

[1] 石井健一郎，他: ” わかりやすいパターン認識”, オーム社，1998.
[2] エルッキ・オヤ: ” パターン認識と部分空間法”, 産業図書 1986.
[3] 飯島: ” 視覚パターンの特徴抽出に関する基礎理論,” 信学会雑誌, Vol.46, No.11, pp.228-235 1963.
[4] 飯島泰蔵: ” パターン認識,” コロナ社, 1973.
[5] Iijima, T. et al.: “A theory of character recognition by pattern matching method,” Proc. 1st. IJCPR, pp.50-56, 1973.
[6] Watanabe, S: “Knowing and Guessing”, John Wiley & Sons, 1969.
[7] Watanabe, S. et al.: “Evaluation and selection of variables in pattern recognition,” Computer and Information Sciences, Vol.2, Julius Tou, ed., New York: Academic Press, pp.91-122, 1967.
[8] Kulikowski, C. A.: “Pattern recognition approach to medical diagnosis,” IEEE Trans. on Systems Science ans Cybernetics, Vol. SSC-6, pp.173-178, 1970.
[9] Watanabe, S. et al.: “Subspace method of pattern recognition,” Proc. 1st. IJCPR, pp.25-32, 1973.
[10] Watanabe, S.: “Karhunen-Loeve expansion and factor analysis,” Trans. on 4th Prague Conf. on Information Theory, Statistical Decision Functions, Random Processes, Publishing House of the Czechoslovak Academy of Sciences, Prague, pp.635-660, 1967.
[11] 飯島: ” 混合類似度による識別理論” 信学技報, PRL74-24, 1974.
[12] Watanabe, S.: “DIRECLADIS - Dinensionality Reduction for Class Discrimination”, Proc. 7th ICPR, p.1252-1255, 1984.
[13] 若林，他, ” 非線形正規化と特徴量の圧縮による手書き漢字認識の高精度化,” 信学技報, PRL95-1, 1995.
[14] 河村聡典, 新田恒雄, “最小分類誤り学習による特徴選択型文字認識”, 信学論 (D-II), Vol.J81-D-II, No.12, pp.2749-2756, 1998
[15] 栗田，他, ” 加重方向指数ヒストグラムと疑似マハラノビス距離を用いた手書き漢字・ひらがな認識,” 信学技報, PRL82-79, 1982.
[16] Kimura, F. et al.: “Modified quadratic discriminant functions and the application to chinese character recognition”, IEEE Trans. PAMI, Vol.9, No.1, pp.149-153, 1987.
[17] 黒沢: “球面ガウス分布から導出される部分空間法”, 信学論, J81-D-II, No.6, pp.1205-1212, 1998.
[18] Kohonen, T.,“Learning vector quantization,” Helsinki University of Technology, Laboratory of Computer and Information Science, Report TKK-F-A601, 1986.
[19] Amari, S.: “A theory of adaptive pattern classifiers”, IEEE Trans. EC, Vol.16, No.3, pp.299-307, 1967.
[20] Katagiri, S. et al.: “A generalized probabilistic descent method”, ASJ, Fall Conf., 2-p-6, pp.141-142, Nagoya, Japan, 1990.
[21] Kohonen, T. et al.,“Classification of phonemes by learning subspaces,” Helsinki University of Technology, Dept. of Technical Physics, Report TKK-F-A348, 1978.
[22] Kuusela, M. et al.,“The averaged subspace methods for spectral pattern recognition,” Proc. 6th. IJCPR, pp.134-137 1982.
[23] Oja, E. et al.,“The ALSM algorithm - an improved subspace method of classification,” Pattern Recognition, Vol.16, No.4, pp.421-427, 1983
[24] 前田, “パターン認識装置,” 公開特許公報, 昭 56-137483, 1980.
[25] Watanabe, H. et al., “Discriminative metric design for pattern recognition,” ICASSP-95, Vol.5, pp.3439-3442, 1995.
[26] 黒沢, ” 確立降下法を 2 次形式類似度と相違度に応用したパターン認識手法,” 信学技報, PRMU97-181, 1997.
[27] 前田，他: “局所構造を導入したパターン・マッチング法”, 信学論, J68-D, No.3, pp.345-352 1985.
[28] 山口，他:“動画像を用いた顔認識システム”, 信学技報, PRMU97-50, 1997.
[29] Turk, M. et al.,“Face recognition using eigenface,” Proc. of Computer Vision and Pattern Recognition, pp.568-591, 1991.
[30] Murase, H. et al.,“Visual learning and recognition of 3-d objects from appearance,” International Journal of Computer Vision, Vol.14, pp.5-24, 1995.

体験談：部分空間法による画像認識

村瀬　洋

名古屋大学情報学区研究科 〒464-8603 名古屋市千種区不老町

E-mail:　murase@is.nagoya-u.ac.jp

あらまし　部分空間法は，今日ではパターン認識の一手法として良く知られた手法であり，関連した理論研究も数多くなされている．私自身が部分空間法を知ったのは 1977 年大学４年生の時である．それ以降，現在に至るまで部分空間法に関連した研究を，画像認識の応用を中心に行ってきた．本発表では，私の部分空間法との出会い，その後の研究にまつわる体験談や苦労した点などについて述べることにする．

キーワード　部分空間法，体験談，文字認識，画像認識

My Experiences: Image Recognition using the Subspace Method

Hiroshi MURASE

Graduate School of Information Science, Nagoya University　Furo-cho, Chikusa-ku, Nagoya, 464-8603 Japan

E-mail:　murase@is.nagoya-u.ac.jp

Keyword　Subspace method, Experiences, Character recognition, Image recognition

1. はじめに

郵便番号の自動読み取り装置が日本で初めて郵便局で稼働し始めたのは 1968 年のことである．当時中学生だった私はその文字認識の原理にとても興味を持ったが，私とパターン認識との付き合いはそれから始まることになる．その後 1977 年，大学の卒研配属で，文字認識の研究室（名大電気，市川教授，三宅助教授）に入った．当時の研究室では，文字認識の対象を活字から手書き文字に発展させようとしていて，構造解析的な認識手法を研究していた．計算機への日本語入力手段として現在では仮名漢字変換が普及しているが，この時代はそれが実用化される前のことである．日本人にとって計算機への文字入力の主力になるのはキーボードではなく，手書き文字認識であると考えられていた時代であった．ちなみにその翌年に東芝から初めてのカナ漢字変換ワープロが 630 万円で発売された．また当時，東芝で前田賢一氏や黒澤由明氏が部分空間法を研究されていたことも後で知ることになる．

2. 部分空間法との出会い（学部４年）

パターン認識について何も知識がないまま研究室に入り，最初に自分で書いた認識プログラムはテンプレートマッチング法あった．入力パターンとテンプレートとの内積を類似度とした手法（単純類似度法）であり，初心者にとっては自然の発想であったと思う．テンプレートマッチング法は活字文字に対してはある程度の精度で認識できたが，手書き平仮名文字に対しては，変形が大きいため，精度良く認識できなかった．また多少工夫してみても，少しも認識率は上がらなかった．テンプレートマッチングでは変形や変動への対処がいかに重要であるかを，この時学んだ．

当時，研究室の仲間となんとなく勉強したのが飯島泰三先生の複合類似度法であった．研究室の助手であった吉村ミツ先生が前職の電総研で飯島先生のグループにいたことも影響していたのだと思う．しかし，学部学生にとって飯島先生の論文は難しかったため解説記事で勉強した．その時，単純類似度法の式 $S=(\mathbf{x},\mathbf{t})$ ｛$\mathbf{x},\mathbf{t}$ は入力とテンプレート｝と，複合類似度法の式 $S=\sum_i \lambda_i(\mathbf{x},\mathbf{e}_i)^2$ ｛$\lambda_i,\mathbf{e}_i$ は固有値，固有ベクトル｝を見比べて，２つのことに気がついた．１つ目は複数のテンプレートを利用することにより変形に強くなること，２つ目は入力パターンと複数テンプレートとの内積の２乗和は，固有値展開により，少数の固有ベクトルとの内積の固有値の重み付け２乗和と等価であることで

ある．線形代数をある程度知っていれば当然だと思うだろうが，当時は自分自身で式を変形してこの点を納得した時にかなり感動したことを覚えている．

さっそく複合類似度法をプログラムし実験を行ったところ，確かに認識率は向上した．しかし，十分なものではなかった．手書き文字の変形の程度は予想以上に大きい.それでも何度も実験を行っているうちに，あるとき認識率がかなり上がった．しかし，調べてみるとプログラムのバグであった．固有値の重みを付け忘れたのである．これこそが，今普通に言われる部分空間法である．それは，渡辺慧先生の CLAFIC 法[1]や，Oja 先生の Subspace Method[2]そのものであった．当時は，何もわからないまま，この重みが大きな意味があるのではないかと，重みをさまざまに変化させたり，非線形に変化させて闇雲に実験を行ったことを思い出す.これが,私と部分空間法との出会いであった．

3. 変動吸収特性核（修士時代）

修士課程に進学した私は，引き続き手書き平仮名文字認識の研究を発展させた．手書き平仮名文字の変形はかなり大きなものである．部分空間法も一種のテンプレートマッチングと考えられるため，大きな変形を吸収することは難しい．そこで考えたのが，変形を生成して新しいテンプレートを作り部分空間法を適用する手法である．その際に，変形を行列で表現できるようにすれば，定式化が美しくなるのではないかと考えた．更にもしこれが実現できれば，もとの画像の変形を生成しなくても部分空間を直接に計算できるようになるのではないかと発想した（図１）．それが，変動吸収特性核[3,4]の考え方である．当時，自己相関行列のことを特性核と呼んでいたために，ここでは特性核という用語を用いた．これは，多数の学習パターンから自己相関行列を一度作成しておけば，変形した学習パターンの自己相関行列は，もとの学習パターンに戻らなくても計算できるという手法である．これにより様々な変形を変形行列という形で適当に定義するだけで，多数の変形に対応した自己相関行列が作成できるようになる．これは，多数の学習パターンを生成してそれを用いて部分空間法を行うことと等価になる．実は，このような発想に至った理由は，定式化が体系的に扱えるということと同時に，当時使用していた計算機の処理能力やディスク容量が現在と比較すると極端に小さかったために，計算量やメモリー量を減らすために，さまざまな工夫が必要であったためである．

部分空間法では，最初に学習パターンから自己相関行列（特性核）$R=(1/n)\sum \mathbf{x}_j\mathbf{x}_j{}^T$ を作成する．その際に，変形行列 A_i を特性核 R に対して掛けることにより新しい特性核 $\tilde{R}=(1/m)\sum A_i R A_i^T$ を作成する．変形行列 A_i をうまくデザインすれば変形を吸収する部分空間を作成することができる．これは変形したパターンを学習パターンとした特性核となる．この特性核のことを変動吸収特性核と呼んだ．変形行列の例を図２に示す．この研究内容は，私の初めての電子情報通信学会の論文となる．

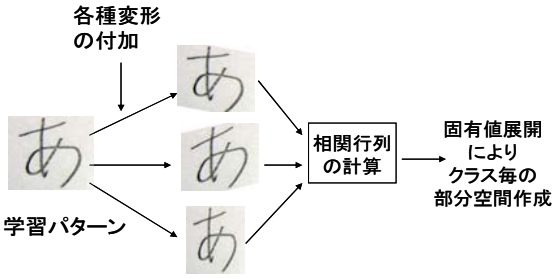


図 1. 部分空間法における変形の付加

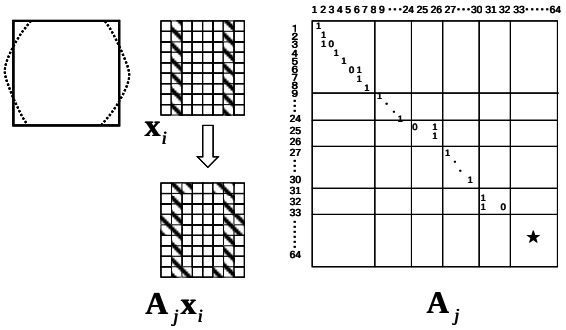


図 2. 変形行列 Ai の例

4. 固有ベクトルの計算（修士時代～33 歳）

当時，大学の計算機で部分空間法を実現するときのネックが固有ベクトルの計算であった．私のいた学科の計算機は中型計算機 FACOM230-38 で，これを数十人で使用していた．混みあってはいたが少なくともこの計算機を利用しているうちは，個別には計算機使用料金はかからなかった．しかしながらＯＳの制約から１プロセスの主記憶のデータ領域が 64Kword（128Kbyte）しかとれなかったのである．

そのために部分空間法の最初の実験では 9x10 画素という今から思えば非常に粗い画素で実験していた．しかし，上記で述べた変動吸収特性核の実験をするためには，ある程度の画素数がないと文字の変形そのものが意味をなさなくなるために，ある程度の画素数が必要となる．部分空間法のアルゴリズムを詳細に分析すると，かなりの部分は使用料金のかからない学科の

計算機で計算できる．しかしどうしても計算できない部分が固有ベクトルの計算部分である．名大の大型計算機センタ(M180)を利用すれば良いこともわかったが計算機使用料金が高い．さまざまにパラメータを変化させて料金を見積もったが，意味のある最小限の実験（それでも 14x15 画素）を１回行うのに，計算機使用量が約５万円かかることがわかった．

これは，当時の私の頂いていた日本育英会の奨学金のほぼ１ヶ月分にあたる．一方で，当時私の所属していた研究室は教授不在になっていたために研究室予算がかなり減らされていることも知っていたので，あまり研究室にも迷惑はかけられない．その中で恐る恐る吉村先生に相談したところ,「すこし高いけど,是非やってみなさい」といわれたが，その一言がとてもうれしかったことを思い出す．但し,「失敗は許されない」ととても緊張して準備をした．何度も学科の計算機でシミュレーションを行い,慎重に手順を確認した結果,幸い大型計算機センターでの計算は１回で済んだ．この時の実験では，５万円のためにずいぶん大変な思いをしたが,この固有ベクトルの計算に苦労したことが,あとになって役に立つことはこの時点ではわからなかった．

1980 年に大学を修了したのちにＮＴＴ研究所に就職し，しばらくは別の仕事を行っていた（文字の切出しと認識を融合させた候補文字ラティス法の提案など)．そのとき，アンダーグラウンドで趣味の研究として楽しんでいたのが，画像の固有ベクトルの高速計算である．大学時代に，固有ベクトルの計算方法でずいぶん苦労したために，その思いが残っていたことと，１Ｍバイトくらいのメモリーが自由に使えるミニコンが当時のＮＴＴの研究室にあり，何か計算しなければバチがあたるというようなケチ根性があったためである．ここで新しく発想したのは，画像をベクトルと考えて固有ベクトルを計算する場合，画像の符号化によりデータ量を圧縮し，圧縮したデータで計算すれば，少ない記憶容量と少ない計算時間で固有ベクトルが計算できるのではないかというものである．その時はこれは学会で発表できるものではないと思っていた．

そんなある時，ＮＴＴ研究所の重役がＡＴ＆Ｔベル研究所に行き，そこの重役と会議するが，ついでに画像関係の研究交流をやろうという話が持ち上がったが,その１名に私が選ばれた．その場で趣味でやっていた固有ベクトル計算の研究を発表するという羽目に陥った．そうなった理由の一つが，ベル研究所の重役の１名がノーベル物理学賞受賞のペンジャス博士（宇宙の背景黒体輻射計測）で，物理学者ならば固有ベクトルが好きではないかというＮＴＴ側の短絡的な発想であった．私は「絶対に違う，これは恥をかく」と思いつつも，とても下手な英語で度胸のみで発表をした．そのとき，ペンジャス博士からは質問はなかったが，ベル研の画像の研究者達からは，初めて聞く発想の研究だと，評価してくれた．アメリカにはこれほどまでにケチ根性で固有ベクトルを計算しようという発想がなかったのだろう．確かに，今の恵まれたＰＣ環境であったら私もこのような発想はしなかったと思う．その反響に気を良くして，IEEE Trans. IP に投稿した[5]．幸い論文の内容は何も修正無しで採録された上に，原稿一式のコピーが添えられていて，そこには全てのページに渡って詳細に英語の表現や文法的な添削がなされていた．これほど親切な査読はその後経験したことがない．その後，同様の考え方を２値画像に特化したものを Pattern Recognition にも投稿した[6].

5. パラメトリック固有空間法（33 歳～40 歳）

1980 年代後半，画像からの３次元物体認識はコンピュータビジョンの一つの主要テーマとなっていた．さらに David Marr 博士の影響だと思うが，画像から幾何学的な特徴を抽出し，それを３次元数理的な手法や，さまざまな仮定や知識を用いて認識しようとするモデルベースの認識手法がいろいろ提案されていた．もともと文字認識をはじめとするパターン認識は，コンピュータビジョンとは別の分野であり，３次元物体をテンプレートマッチングで認識しようというと,「そんな適当な方法ではできるはずがない」と文句を言われそうな時代であった．

ただ，固有ベクトルの計算法で少し気をよくしていたこともあり，手書き文字でも部分空間法である程度の認識精度が得られるのであるから，３次元物体の認識も変形や変動さえうまく表現できれば，テンプレートマッチングの枠組みで認識できるという直感もあった．また，Sirovich 氏や Turk 氏らが顔認識のための固有顔を発表した時代である．彼らの当時の研究は正面顔の認識であったので,これは文字認識と同じであり,過去の経験からうまくいくことは想像の範囲内であった．そこで，それでは自分は３次元の変形に対応してやろうと考えた.手書き文字認識や顔認識の場合には,変形の主な要素は平均の周りの分布であるのに対し,３次元物体認識は，向きというパラメータにより変化する部分であり，逆に変形が扱いやすいのではないかと考えた．そこで着目したのは，パラメータにより補間するという考え方である．物体の向きなどのパラメータで変形をコントロールすれば良い．これがパラメトリック固有空間法[7]の最初の発想である．

パラメトリック固有空間法は２次元画像の照合により３次元物体を認識する手法である．３次元物体は見る方向や照明により見かけが大きく変化する．そこ

で様々に変化する見かけ画像をパラメトリック固有空間表現で表し，認識に用いる．パラメトリック固有空間法では，学習段階では，最初に，認識するべき物体を様々な方向，さまざまな照明条件[7]で体系的に撮影する．それを学習パターンとして主成分分析を行い，固有ベクトル（固有空間）を計算する（図３）．

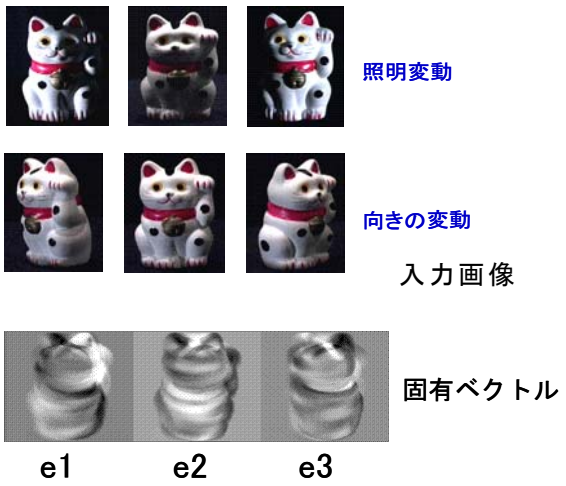


図 3. 入力画像とその固有ベクトル

次に各学習パターンを固有空間に投影し，その特徴点を記憶する．学習パターンのそれぞれの見かけ画像が大きく異なる場合には，特徴点も固有空間上でまばらに分布する点となる．一般的に物体の見かけ画像の連続的な変化は，固有空間上の点の連続的な軌跡になる(図4).つまり特徴点をスプライン関数などで補間することで，中間的な画像の認識を実現することができる．また照明条件についても同様に固有空間上で補間を行い．中間的な照明条件を表現することができる．

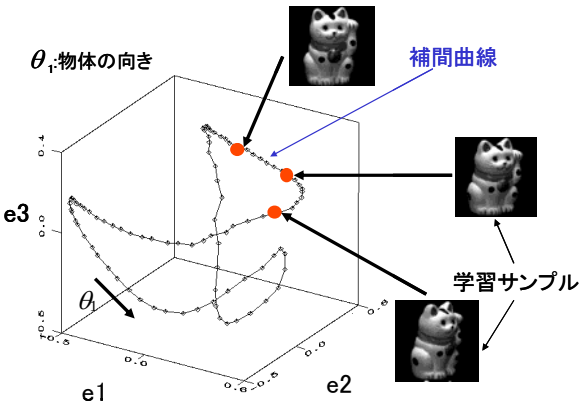


図 4. 学習パターンの特徴空間上での補間

認識段階では，未知の画像をこの固有空間に投影し，その特徴点が予め作成した軌跡の上に存在すれば，その物体とし，そうでなければその物体では無いとする．軌跡は物体の向きや照明条件の情報を持っているので，その情報を抽出することにより，物体の向きの検出も行うことができる．

この考え方は，その後，コロンビア大学の Nayar 先生と共に，照明変動への拡張[8,9],物体の抽出[10,11]，動画像の認識[12]，ビジュアルサーボ[13]，特徴抽出[14]へと応用していった．

6. 部分空間法の応用と生成型学習（40 歳以降）

5.1　カーネル非線形部分空間法

1998 年頃に，識別方法としてサポートベクトルマシン（SVM）が流行していた．SVM の本質は，マージンを最大化する識別境界の設定にあるが，実際上は SVM と同時にそれを支えているカーネルが大きな役割を果たしていることが分かっていた．カーネルにより多次元に投影された空間において単純な識別関数でも，もとの空間では複雑な識別関数を表現できる．NTT の前田英作氏とともに部分空間法とカーネルとを組み合わせたカーネル非線形部分空間法[15]を提案することになった．

その後しばらくはアクティブ探索法などのマルチメディア探索の研究を行っていて，部分空間法の研究からは離れていた．

5.2　雨滴認識への応用

部分空間法は文字認識や物体認識だけでなく，さまざまな応用がある．そこで，雨滴認識への応用について試してみた．具体的には，自動車などの車載カメラからフロントガラス上の雨滴を検出する手法である．現在，自動ワイパーなどを目的に自動車にはレインセンサーなどが取り付けられているが，これはセンサーの設置されている場所がぬれているかどうかをチェックしているだけである．フロントガラス全体にどのように雨滴が付着しているかを認識することができれば，フロントガラスを見て雨滴を検出することによりドライバーと類似した視点での雨滴検出が可能となる[16].

フロントガラス上の雨滴は屈折現象により画像上で表現される．雨滴は一種の凸レンズの効果によりどれも類似したパターンとなる（図５）．つまり類似したパターンを集めてこれから部分空間を作成すれば，部分空間によりフロントガラス上の雨滴を検出することが可能となる．雨滴の例と，それを用いて学習した部分空間法により，雨滴を検出した例を図６に示す．

図 5. 雨滴パターン

図 6.　雨滴を検出した例

5.3　複数フレームを利用した部分空間法

近年,カメラ付き携帯やデジカメの普及は著しい.これらで撮影した画像から文字を認識するニーズは高い．カメラの解像度は年年向上しているが，どれだけカメラの解像度が向上しても，一度にたくさんの文字情報を入力したいという要求はあり，その段階で低解像度の画像を認識したいという問題は発生する．低解像度文字の例を図７に示す.

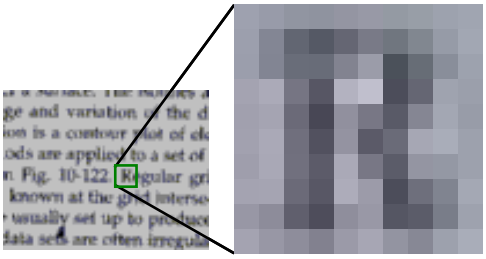


図 7. 低解像度の文字

このようにカメラ付き携帯などについているカメラで少しはなれて撮影したような品質の低い画像の認識を考えてみる．この場合，１枚の画像だけでは高い精度が得られなくても，複数のフレームが得られれば認識精度が向上する．関連研究としては相互部分空間法などもある．われわれは，もう少し単純に１枚１枚を部分空間法で認識しそれをロバスト統計により統合する手法を提案した.その結果,かなり解像度が低く，１枚だけでは精度よく認識できない文字などに対しても認識精度が向上することを確認した[17].

5.4　ボケ関数による生成型学習

低品質な画像を認識する際に，前節のように複数フレームを利用することも考えられるが，もう一つのアプローチは，変形をいかに表現するかということになる．変形さえうまく取り扱えればテンプレートマッチングや部分空間法はある程度の性能が得られるわけである．そこでこれまでの考え方を整理して生成型学習を考えた.

低解像度の文字を認識するためには，例えば超解像のように複数の低解像度の画像を集めることにより解像度の高い画像を生成してから認識する手法もあるが,もう一つの手法が生成型学習のアプローチである．ここでは生成型部分空間法について紹介する．もしある文字が低解像度で観測されるとすれば，それはどのような画像になるであろうかを考え，そのような画像を生成して認識する手法[18]である.

低解像度の画像が得られる過程を考えると，光学的なボケ，撮像系でのボケや画素に対応するセンサーの配置に伴うボケなどさまざまな要因が考えられる．ここでは，第１近似として，これらを一括して生成関数という名前のフィルタで表現する手法を試した．生成関数は，点広がり関数（ＰＳＦ）と画素のサンプリングの配置をモデル化したものである．ボケ関数を適切に推定し，それを用いることにより学習サンプルを生成することにより認識精度が向上することを実験により確認した[19].

5.5　生成型学習を用いた道路標識認識

車載カメラから得られた画像から周囲を認識しドライバーを支援する技術が望まれている．例えば道路標識の認識がある．この場合，ボケだけではなく，傾き，回転，切り出し誤差，などさまざまな変動要因がある．前節で示したボケ関数により生成型学習法をさらに推し進めて，ここでは生成型学習を用いた部分空間法により道路交通標識を認識する手法について紹介する．ここでは,考えられる変動要因を図８に示す６種類に分類し，６種類の生成モデルを提案した．提案手法ではこれらの生成モデルを用いて現実に想定されるであろう劣化をシミュレーションして学習画像を生成する．これを用いて部分空間法により認識する方法である．生成により認識精度が向上することが明らかになった[20].

図 8.　劣化要因の 6 つのモデル

7. むすび

以上，著者自身の部分空間法に関係する体験を述べた．これは，部分空間法に限った話ではないが，困難にあったときにそれに立ち向かい，苦労しながらいろいろ工夫することは次の研究の展開につながる．著者の体験談が，少しでも若い研究者の今後の研究の進め方の参考になれば幸いである．

謝辞

学生時代にご指導いただいた当時名古屋大学電気工学科の三宅康二先生，吉村ミツ先生および研究室の皆様に感謝する．ＮＴＴ時代に本研究をサポートしていただいた内藤誠一郎さんや当時の研究室の皆様，コロンビア大学の Shree Nayar 先生や学生の Sameer Nene 氏に深謝する．5.2 以降の最近の研究は，名古屋大学の研究室のメンバーである井手一郎先生，目加田慶人先生，高橋友和氏，柳詰進介氏，栗畑博幸氏，石田皓之氏と共に行ったものである．

文　献

[1] S. Watanabe and N. Pakvasa, “Subspace method in pattern recognition,” Proc. 1st Int. J. Conf on Pattern Recognition, Washington DC, pp. 25-32, Feb. 1973.

[2] E. Oja, Subspace methods of pattern recognition, Research Studies Press, Hertfordshire, 1983.

[3] 村瀬 洋, 木村 文隆, 吉村 ミツ, 三宅 康二, "パターン整合法における特性核の改良とその手書き平仮名文字認識への応用", 信学論(D), J64-D, 3,pp.276-283, 1981

[4] 村瀬 洋, 木村 文隆, 吉村 ミツ, 三宅 康二, 変動吸収特性核を用いた手書き平仮名文字認識, 信学技報 PRL, PRL79-87 pp.7-12, 1980

[5] H. Murase, M. Lindenbaum, Spatial Temporal Adaptive method for partial eigenstructure decomposition of large images,IEEE Trans. IP, 5, pp.620-629, 1995.

[6] J. B. Roseborough, H. Murase, Partial eigenvalue decomposition for large image sets using run-length encoding, Pattern Recognition, vol. 28, No. 3, pp.421-430, 1995.

[7] 村瀬 洋, S.K.Nayar, ２次元照合による３次元物体認識 - パラメトリック固有空間法-, 信学論、J77-D-II, 11, pp.2179-2187, 1994.

[8] H.Murase, S.K.Nayar, Visual learning and recognition of 3-D objects from appearance , International Journal of Computer Vision, Vol. 14, pp.5-24, 1995.

[9] H.Murase, S.K.Nayar, Illumination Planning for object recognition using parametric eigenspace, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 16, No. 12, pp.1219-1227, 1994.

[10] 村瀬 洋, S.K.Nayar,”多重解像度と固有空間表現による３次元物体のイメージスポッティング”, 情報処理学会論文誌,Vol.36, No.10, pp.2234-2243, 1995.

[11] H.Murase, S.K.Nayar, Appearance-based detection of 3D objects in cluttered scenes, Pattern Recognition Letters, 18, 375-384, 1997.

[12] H.Murase, R.Sakai, Moving object recognition in eigenspace representation: Gait analysis and lip reading, Pattern recognition letters, 17, pp.155-162, 1996.

[13] S.K.Nayar, S.A.Nene, H.Murase, Subspace methods for robot vision, IEEE Trans. Robotics and Automation, Vol.12, No.5, pp.750-758, October, 1996.

[14] Simon Baker, Shree K. Nayar, and Hiroshi Murase, Parametric feature detection, International Journal of Computer Vision, Vol. 21, No.1, pp.27-50, 1998.

[15] 前田 英作, 村瀬 洋, カーネル非線形部分空間法によるパターン認識, 電子情報通信学会論文誌, Vol. J82-DII,No,4, pp.600-612,1999.

[16] H. Kurihata, T. Takahashi, Y. Mekada, I. Ide, H. Murase, Y. Tamatsu, Miyahara:"Rainy Weather Recognition from In-Vehicle Camera Images for Driver Assistance" , IEEE 2005 Intelligent Vehicle Symposium, MPO4, pp.204-209, Jul. 2005

[17] 柳詰進介, 高橋友和, 井手一郎, 目加田慶人, 村瀬洋, 携帯ディジタルカメラにより撮影された動画像からの低解像度文字認識, 電子情報通信学会論文誌, Vol.J89-D No.2 pp.323-331, 2006

[18] 村瀬洋, 画像認識のための生成型学習, 情報処理学会論文誌, Vol.46 No.SIG 15(CVIM12), pp.35-42, 2005

[19] 石田皓之, 柳詰進介, 目加田慶人, 井手一郎, 村瀬洋, 部分空間法による低解像度文字認識のための生成型学習法, 電子情報通信学会, 信学技法 PRMU2004-7, pp.37-42, 2004

[20] 石田皓之, 高橋友和, 井手一郎, 村瀬洋, 榎本光宏, 道路標識認識のための学習データ生成手法の検討, 画像の認識・理解シンポジウム論文集 MIRU2005, pp.989-996, 2005

MIRU2006サテライトワークショップ
部分空間法研究会2006　06.07.18

部分空間法入門

黒沢由明（東芝ソリューション）

1

部分空間法が考案されていた頃、
こんな事を考えている研究者がいました。

昭和45年度電子通信学会全国大会

125　超楕円形識別関数による文字の識別

葛城銑史
（東芝総研）

はしがき　パターンの各類ごとの統計的分布を等固分析により知り，これに基づき識別関数を構成することができる。[1][2] この報告では，計算機シミュレーションによる識別実験の結果を述べる。

識別関数　入力パターンベクトルを x $(x_1, x_2, \cdots, x_N)$ とすると，各類に対する識別関数は，各類の平均パターンを $\bar{x}^k$，各類の分散行列 V_k の個有ベクトルを ϕ_j^k，個有値を $\lambda_j^{(k)}$ とすると，

$$d^k(x,k) = \frac{1}{\lambda_0^{(k)}}(x-\bar{x}^k)^2 - \sum_{j=1}^{M_k}\left(\frac{1}{\lambda_0^{(k)}} - \frac{1}{\lambda_j^{(k)}}\right)(x-\bar{x}^k, \phi_j^k)^2 \quad \cdots\cdots (1)$$

で定義される。ここで，M_k は，個有軸の打切数，$\lambda_0^{(k)}$ は，第 M_k 軸以降の個有値を代表して近似する値である。d^k の等値面が N 次元空間中の超楕円面となるので，超楕円形識別

葛城による疑似ベイズと同じ距離尺度の提案

2

部分空間法とは？

x：入力ベクトル（次元N）

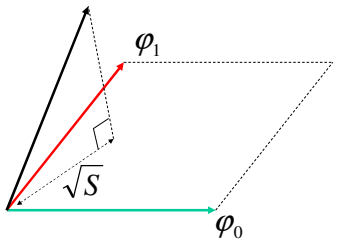
φ_i：カテゴリ i の辞書ベクトル（直交ベクトル）

r： 部分空間の次元

$$S_i = \sum_{i=0}^{r-1}(x,\varphi_i)^2 / \|x\|^2 \cdot \|\varphi_i\|^2$$

3

図示すると、 x



4

行列を使って書くと、

x： 入力ベクトル（次元N）

φ_i： 正規直交ベクトル

$P = \sum_{i=0}^{r-1}\varphi_i\varphi_i^T$ と置いて

$$S_i = x^T P x = \sum_{i=0}^{r-1}(x,\varphi_i)^2$$

（$r=1$の時は単純類似度と呼ぶ）

5

1970前後　部分空間法の誕生

SELFIC　特徴選択
↓
CLAFIC　部分空間法

S. Watanabe
（ハワイ大）

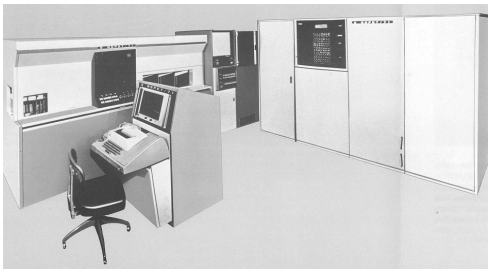
複合類似度法　飯島（電総研）

↓

複合類似度法はボケの理論を含む、認識全体の体系で、その中の類似度システムが重み付き部分空間法と見なせる。

ASPET71の開発（国家プロジェクト）

6



ASPET71の概観　　（電総研、東芝）

英数字活字の高精度読取、2000字/秒、200枚/分
類似度計算はアナログ回路

7

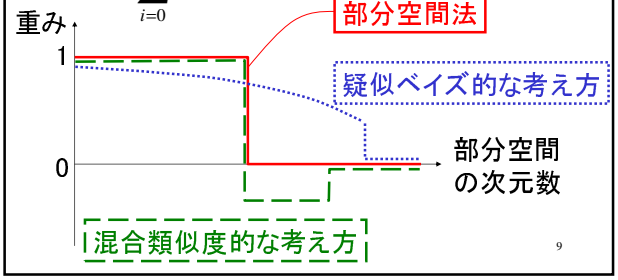
重みを付けると？

8

重み付き部分空間法

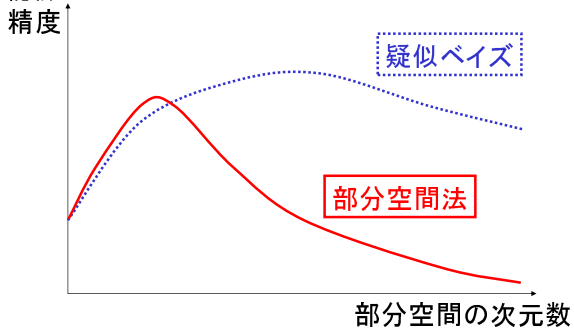
μ_i ： 重み

$$S_i = \sum_{i=0}^{r-1} \mu_i (x, \varphi_i)^2$$



9

重みを付ける効果



10

重み付き部分空間法への展開

複合類似度法（1970前後）　　飯島

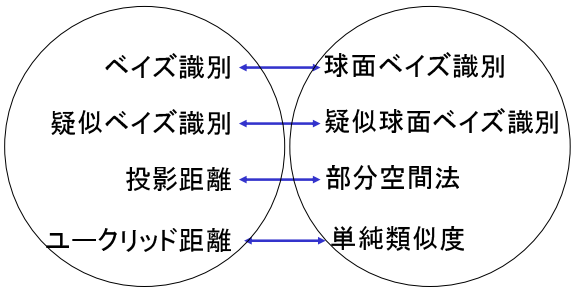
疑似ベイズ識別（1980前半）　　三重大/名大

混合類似度法（1980代）　　飯島

11

疑似ベイズって部分空間法なんですか？

→ ではありませんが、親戚です。



12

式で比べると、

$$S_i = \|x-m\|^2 - \sum_{i=0}^{r-1} \mu_i (x-m, \varphi_i)^2$$

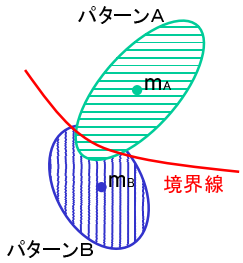
$$S_i = \|x-m\|^2$$ 距離型

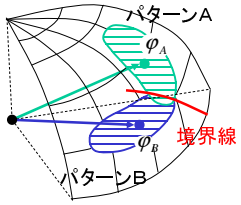
$$S_i = \sum_{i=0}^{r-1} \mu_i (x, \varphi_i)^2$$

$$S_i = (x, \varphi_i)^2$$ 内積型

13

図で比べると、





距離型　　内積型

14

どうやって部分空間を求めるの？

15

主成分分析

$x_\alpha = \{x_{\alpha j}\}$: α番目の学習ベクトル

$\varphi = \{\varphi_j\}, \|\varphi\| = 1$: 求めたい辞書ベクトル

$$J = \sum_{\alpha \in \Omega} (x_\alpha, \varphi)^2 - \lambda(\|\varphi\| - 1)^2$$ Jを最大化

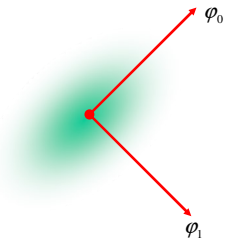
$$\frac{\partial J}{\partial \varphi_j} = \sum_{\alpha \in \Omega} 2(x_\alpha, \varphi) x_{\alpha j} - 2\lambda\varphi_j = 0,$$

$$\sum_{\alpha \in \Omega} x_\alpha x_\alpha{}^T \varphi = \lambda\varphi, \qquad k = \sum_{\alpha \in \Omega} x_\alpha x_\alpha{}^T$$ と置いて、

$$K\varphi = \lambda\varphi$$

16

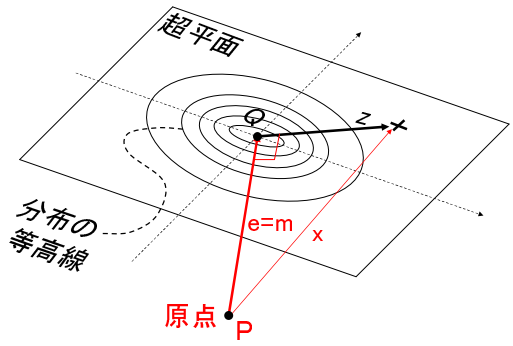
主成分分析結果：



主成分分析はCLAFIC、複合類似度が提案された最初から使われている。

17

球面に接する超平面上でのパターン分布の例：



18

逐次最適化手法によって部分空間を求める

部分空間を決めるパラメータを、認識結果に基づいて少々、変更する。通常、正読したパターンによって、そちらに近づくように、誤読したパターンによって、そちらから遠ざかるように加減算する。

φ：パラメータ、辞書

$\Delta\varphi$：パラメータ、辞書の更新量

α：学習の強さを決める重み

$\varphi = \varphi \pm \alpha\Delta\varphi$

19

逐次最適化手法によって部分空間を求める

LVQ　1980後半、コホーネン（ヘルシンキエ科大）

φ：　辞書ベクトル（相違度はユークリッド距離）

$\varphi = \varphi \pm \alpha(x - \varphi)$　で更新　（単純類似度版も可能）

↓

LSM　1980前後（何故か、こちらが先）、コホーネン

φ_i：　i番目の辞書ベクトル（正規直交）

$\varphi_i = \varphi_i \pm \alpha(x, \varphi_i)x$　で更新、その後、シュミットの直交化

↓

20

逐次最適化手法によって部分空間を求める

ALSM=学習型複合類似度法

1980前後、オヤ（ヘルシンキエ科大）、前田（東芝）

K：能率行列（共分散行列）

$K = K \pm \alpha xx^T$　で更新、その後、主成分分析

DMD　1995、渡辺、山口、片桐（ATR）

P：部分空間へのプロジェクションとして、このPを複数の回転行列の積で表し、各回転行列の回転角度をGPDで求めるもの。

21

部分空間法の研究課題

（3つの方向性）

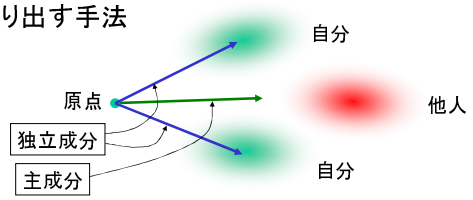
22

1. 部分空間の求め方

・主成分分析　1970～

・独立成分分析　1990代～

分布のピークが複数ある時、それぞれを成分として取り出す手法



・逐次最的化手法　1980代～

LVQ,LSM,ALSM,DMD(GPDの適用)など

23

2. 類似度の構成法

・相互部分空間法　1980前後、前田（東芝）

部分空間と部分空間の類似度を使う

・カーネル・トリックの導入　最近のSVM関連の研究

空間を超高次元空間に射影し、そこで部分空間法を構成する。

・混合類似度的な考え方　最近でも続いている

部分空間法は自分しか見ていないが、ライバルも考慮して類似度を構成しようという考え方。

24

3. 意外な対象、意外な適用

・アイゲン・フェース　1990前後、ペントランド（MIT）
顔画像の固有ベクトルのこと。

・パラメトリック固有空間法 1995前後、村瀬（NTT）
物体の動きを部分空間の中で捉える

25

部分空間法を使ってみましょう

26

辞書の作り方

・数値計算ソフトを入手する。
（自分で作っても良いですが）

redhat系linuxの場合

www.netlib.org>browse>lapack>rpms>lapack*.rpm
をダウンロード、インストールする。

vineの場合、
apt-get install lapack でもインストールできる。
g2cが無い場合は、apt-get install gcc-g77
gccでのリンク例：–llapack –lblas –lm –lg2c

この例のよう、ウェブで捜して持ってくる。

27

辞書の作り方

・ランク落ちしていると、トラブリます。学習パターン数が少ない時は要注意。出来たベクトル、固有値などをチェックしておいた方が良い。

・整数化する場合、直交性や正規性の崩れがどの程度か見ておく必要がある。

28

類似度計算の方法

・整数化する時はオーバーフローに注意。意外と簡単にオーバーフローします。

・正規化の省略

$$S_i = \sum_{i=0}^{r-1} (x, \varphi_i)^2 \quad / \|x\|^2 \cdot \|\varphi_i\|^2$$

x が正規化されていれば、不要。
類似度を閾値比較しなければ、不要。

辞書が正規化されていれば不要

29

類似度計算の方法

・高速化（実験の規模が大きい時）

専用ハード　（昔）
↓
アセンブラ・高速化コーディング　（昔）
↓
cでの高速化コーディング　（昔）
↓
並列計算機能（SSEなど）の利用（昔～今）
↓
マルチコア化対応　（今～将来）

30

類似度計算の方法

・正準化（複合類似度法の中で提案されている）

$$x = x - (x,h)h, \quad \|h\| = 1$$

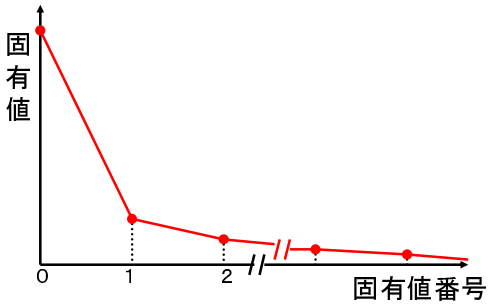
h成分を除去する操作。hとして要素がすべて1のベクトルを正規化して使うと、直流成分を除くことができる。hとして全カテゴリのパターンから作る平均パターンを使うという考え方もある。

整数化した場合にダイナミック･レンジの拡大の効果がある。性能向上の効果もあるようだが、その程度については、はっきりしていない。

31

確認しておきたいこと

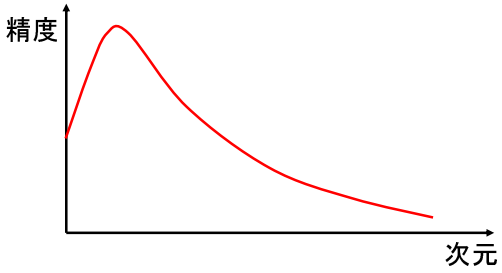
・辞書の直交性、辞書の正規性（ノルム＝1）、固有値の出方



32

確認しておきたいこと

・辞書ベクトルの数（部分空間の次元）と精度の関係

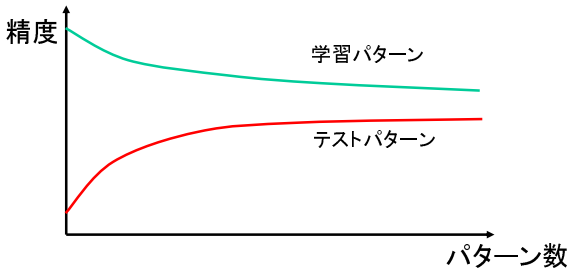


次元数はそろえた方が良いのか、
そろえない方が良いのか。

33

確認しておきたいこと

パターン数と精度の関係



34

終

35

部分空間法研究会 Subspace2006 委員会

実行委員会

実行委員長	福井 和広	(筑波大)
実行委員	内田 誠一	(九大)
	大町 真一郎	(東北大)
	坂野 鋭	(NTT データ)
	佐藤 敦	(NEC)
	佐藤 真一	(国立情報学研究所)
	佐藤 洋一	(東大)
	牧 淳人	(京大)
会場担当	大町 方子	(東北文化学園大)
顧問	前田賢一	(東芝)

部分空間法研究会 (Subspace2006) 予稿集

編集　部分空間法研究会 実行委員会
発行日　2006 年 7 月 11 日