怪奇!! 次元の呪い

ー識別問題，パターン認識，データマイニングの初心者のためにー

（前編）

（株）NTTデータ　坂野　鋭　sakano@rd.nttdata.co.jp

NEC　山田 敬嗣　yamada@ccm.cl.nec.co.jp

■序章ー迫りくる修論発表会ー■

S君は焦っていた．修論発表会が来月に迫っているのに，実験も何もできていないのである．何とかして今月中に新しいアイディアを思いついて，実験を済ませて論文まではともかくプレゼンの準備まではしなくてはならない．

S君のテーマは手書き文字の認識．なんといってもよく分からない情報理論なんかの研究より結果が目で見て分かる文字認識の研究を選んだわけだ．研究室では，他にはデータマイニングとかバイオインフォマティックスの研究をやっているらしいが，S君にはなぜ同じ研究室でこんな研究をやっているかは分からない．あまり，学問には熱心ではないのだ．

だから，というわけでもないのだろうがM1の時は遊びまくった．S君の所属する研究室の基本方針は放任主義であったので，3カ月に論文の1本も読んでおけば誰にも何も言われずに済んでいたのだ．

企業に就職した先輩たちからも人生で一番遊べるのはM1の時でM2になったら修士論文を仕上げるために頑張らなくてはいけないと聞いていたので気合を入れて遊びまくった．勉強はM2になってからすればいいと思っていた．

しかし，M2になると，就職活動とかで意外と時間がとられることになった．

そんなわけで，今は1カ月で1年分を取り返さなくてはならない．とにかく，手っ取り早く実験を終了させたい．

まず，認識の対象は手書き数字に限定することにする．漢字は字種数が多いので避ける．先輩の修論をひっくり返すと，手書き数字のデータベースIPTP CDROM1☆1というものがあることが分かった．これで95%くらいの認識率が出れば全国大会では発表できて，何とか修論にはなりそうだ．

実験のために，まずデータを用意した．卒業した先輩のディレクトリをあさるとIPTP CDROM1の整理されたデータと取り扱うためのプログラムが見つかった．

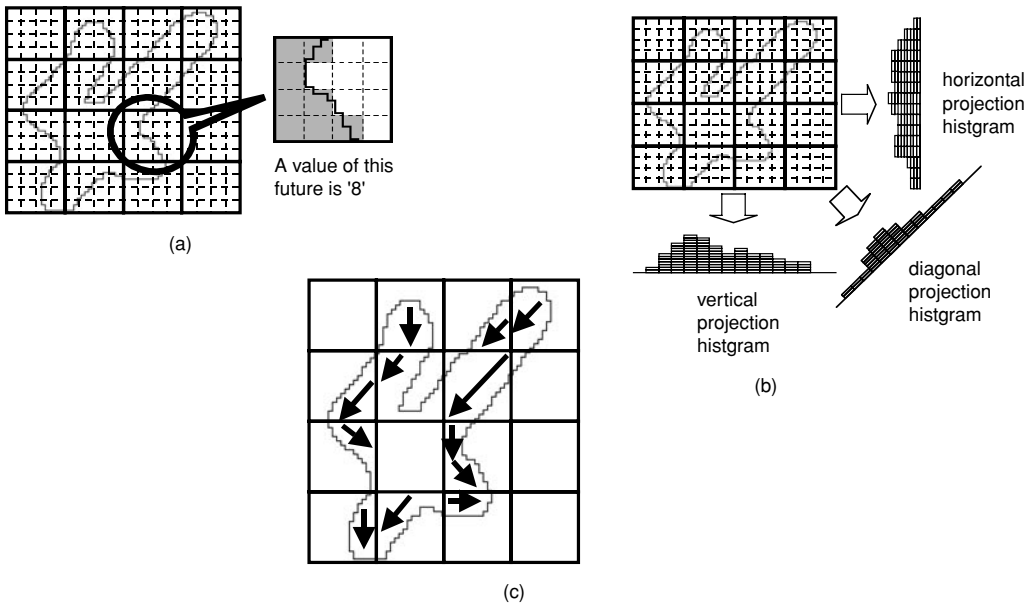
文字認識のアルゴリズムは，文字画像の大きさや位置を揃え，ごま塩ノイズ等の雑音除去を行う前処理部，文字画像に対してさまざまな画像処理を施して特徴ベクトルと呼ばれる数値に変換する特徴抽出部，特徴ベクトルがどの文字かを判定する識別部などからなる．先輩のディレクトリにはすべてのソースが揃っていた．

次は，研究のアイディアである．できるだけ簡単に実験できるものがいい．そこで，問題の先輩の修論を見ると，メッシュ特徴，方向線素特徴などいろいろな特徴抽出系について評価を行い，新しい特徴を提案している．しかし，経験のないS君にはどうしてその新しい特徴が有効なのかは分からなかった．

このとき，S君の頭にアイディアが閃いた．先輩が評価している個々の特徴を並べて，性能が向上すれば新しい文字認識方法といえるじゃないか！　これならうまくいきそうだし，実験も先輩の作ったプログラムを走らせるだけだから簡単だ．S君は早速実験にとりかかった．

まず，データを用意する．S君はIPTP CDROM1から各字種あたり200個を取り出し，小さなデータセットを作った．IPTP CDROM1には1字種あたり3,000個以上

☆1 IPTP CDROM1は，郵政研究所が提供する手書き数字データベース．



(a) メッシュ特徴．文字画像を荒いメッシュで区切りメッシュの中の黒画素数を特徴値として使う．文字線の位置変動に強いといわれている．
(b) 周辺分布．4方向から射影したヒストグラムを特徴値として用いる．文字線の方向性を反映する．
(c) 方向性特徴．いろいろなバリエーションがあるが，文字線の方向を直接計測し，何らかの表現で特徴値とする．文字認識においては最も効果がある特徴抽出系である．

図-1　文字認識のための特徴のいろいろ

のデータがあるが実験を早くまわすためにはサンプルは少ない方がいい．それに以前に読んだ手書き漢字認識の論文では学習データ，テストデータとして字種あたり100個のデータを使っていたから，数字でもそれでよかろうと考えた．

さて，実験である．手始めに簡単な特徴としてメッシュ特徴（**図-1**（a））と周辺分布（**図-1**（b））を試した．

最初に64次元のメッシュ特徴と，16次元の周辺分布での認識率を個々に調べ，次に統合して80次元にした場合の認識率を計測する．識別部には字種ごとの特徴ベクトルを平均したベクトルからサンプルまでのユークリッド距離が，最も近かった字種を認識結果とするアルゴリズムを用いることにした．

実験は簡単にできた．メッシュ特徴は値のレンジが0から64，周辺分布特徴は0～1,024なので周辺分布特徴の値を16で割るという工夫も凝らした．メッシュ特徴，周辺分布特徴ではそれぞれ学習データでは70％，テストデータでは60％程度だった認識率が統合して80次元にしたことによってテストデータで80％程度に改善した．凄い改善である．実験の準備に1週間ほどかかったが，無駄ではなかった．どうせプレゼン用のスライドは前の晩に作るのだから，あと3週間のうちにありったけの特徴を統合していけば目標の精度は達成できるに違いない．

次の1週間はいろいろな特徴抽出系についての個別の認識率を出していくことで過ぎた．文字認識の世界には実にいろいろな特徴があるものだ．全体の特徴は1,000次元を超える勢いであった．

さて，ついに目標の実験開始だ．これまでに試した周辺分布，メッシュ特徴，モーメント特徴，方向性特徴など，合わせて1,000次元を超える特徴のそれぞれのレンジを0から64に合わせ，認識率を出す．1,000次元もあるので多少時間がかかる．まず学習データでの認識率を出した．92.34％!!　今までの最高値だ．S君は俄然自信を持った．自分の考えは正しかった．これで修論発表会は乗り切れる．

いよいよテストデータの実験である．これも少し時間がかかって，さて，認識率が出た．82％？　確かに大半の特徴単独の場合より高くなっているが，方向性特徴（**図-1**（c））単独の場合の86％より低い．どうしてだ？

■ニューラルネットに頼ってみよう■

S君は悩んだ．修論発表会まで，あと2週間しかないのだ．S君は研究室の後輩に相談した．後輩はS君と違って勉強熱心な奴なのだ．

その後輩によると，手書き文字認識にはニューラルネットが有効ということらしい．ニューラルネットは人間の脳を真似たアルゴリズムで，代表的なものは入力層，中間層，出力層の3つの層を持ち，入力層のユニ

ット数が特徴の次元数，出力層のユニット数が字種の数で構成される．なんでも，中間層のユニット数を増やすほど複雑な識別境界を実現できる識別器なのだそうだ．S君は早速ソースをもらい，試してみることにした．

調整するパラメータは主として中間層のユニット数だが，とりあえず思い切り多くすることにして1,000個にしてみた．なんといっても多い方が複雑な識別面を作ってくれるということなのだから多少学習に時間がかかることになるが，後からパラメータを振ってみたりすることを考えると，この方が得に思えた．

数時間後，学習が終了した．学習データでの認識率は100％!!　今までの最高だ，さすがはニューラルネットである．じゃあ，テストデータでの実験をやってみよう．多少の計算のあと，計算機が認識率を出してきた．68％？　なんだこれは，ずいぶんと性能が悪いじゃないか．S君は何か間違いがあるはずだと思い，ニューラルネットの解説を探し始めた．

まず，電子情報通信学会誌に5回連載の解説[1]を見つけたが，長いのでやめておく．次々に文献をさかのぼっていくと90年代前半にたくさん解説が見つかった．パソコン雑誌みたいな雑誌にもたくさんでている．S君はできるだけやさしそうな解説を選び，その中でも式の少なそうなものを読み始めた．

S君の関心を引いたのは過学習という現象だった．なんでも学習データについて100％になるまで学習すると，テストデータでの認識率が落ちていくということらしい[2]．なるほど！とS君は思った．うまくいかないのはこれが原因だったのか！

S君は新たな実験に入った．学習データの認識率が90％くらいのところで止めてみることにする．しかし，認識率は72％くらい．しかし，少しは上がったのだ．希望はある．S君は実験を繰り返した．解説には学習を途中で止めるとよいとは書いてあったが，どこで止めるとよいかは書かれていなかった．実験を繰り返すしかないのだ．

実験を繰り返すうちに1週間が経過してしまった．S君はかなり焦っていた．いくら実験を繰り返してもニューラルネットは80％以上の認識率を出してこないのだ．このままではM3になってしまう．

■そうか！ 最近はサポートベクトルマシンが■
■流行りなのか■

思い余ったS君はニューラルネット以外に何かないか探し始めた．ふと，「情報処理」を見ると最近はサポートベクトルマシン（Support Vector Machine, 以下SVM）というものが強力なパターン認識技術として使われているらしい[3]．流行の方法なら，うまくいくに違いない．S君は最後の希望をサポートベクトルマシンにかけることにした．

解説論文の前半を読んで大体のやり方を理解したと思ったS君はWWWでSVMのソースを探した．すると`http://www.kernel-machines.org`というサイトにたくさんのソースがあった．どれがいいのか分からないから，適当にいくつかダウンロードして，makeが通ったやつで実験を始めた．

ところが，いつまでたっても学習が終わらない．最新型のPCでないにしても，ちょっと時間がかかりすぎる．結局学習が済むのに2日かかった．

これだけ時間がかかったのだから，大丈夫だろうと考えてテストデータの認識率を算出した．しかし，82％程度である．S君はもう一度解説を読み直した．注意深く読むと，SVMにはカーネルパラメータとソフトマージンパラメータという2つのパラメータがあり，これを調整しないとうまくいかないということであった．

S君の目の前は真っ暗になった．あと，5日しかないのに学習に2日かかるアルゴリズムをまわしつづけなくてはならないのか!!

どうすればいいのだろう？　途方にくれるS君の脳裏に「M3」の2文字が明滅した．就職だって苦労して決めたのに.....

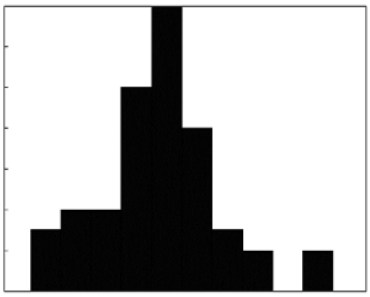
（なお，このお話はほとんど著者らの創作であり，実在の人物，研究室，大学，企業とはほとんど関係がありません）

■次元は呪う－識別問題に潜む罠－■

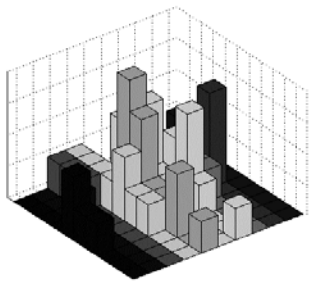
S君の研究はなぜうまくいかなかったのだろう？　M1の時も遊んでいてはいけない，というしごくまっとうなご意見はとりあえずおいておくとして，S君のアプローチ自体は，そんなに違和感なく受け入れられる方が多いのではないだろうか？

こうした認識の問題を考える場合に，処理が重ければ精度が高いアルゴリズムが使えるとか，情報が多ければ多いほど性能が高くなるといった感覚は，普通に持たれる方が多い．しかし，識別問題を考える場合にはこうした「感覚」は通用しない．処理量の低い識別方法や情報が少ない場合の方が識別精度が高い場合はあまり珍しくはない．

この，やや直感から外れた現実はなぜ現れるのだろうか？　パターン認識問題に限らず，データマイニン



(a) 1次元の場合



(b) 2次元の場合

図-2　正規乱数のヒストグラム

グやバイオインフォマティックスで現れる識別問題には，共通する特徴がある．それは，数十から数百，時には数千次元という多変数のデータを扱わなくてはならないということである．すなわち，識別問題を解くべき舞台は日常感覚では考えられない高次元の空間であり，そもそも，日常的な直感による理解は不可能なのである．

こうした高次元性から現れる独特の問題は「次元の呪い」と呼ばれている．次元の呪いは，具体的には学習機械の過適応，サンプル数の問題として現れる．

この問題を理解するために，ヒストグラムによる密度関数推定を考えてみよう．正規乱数から発生させた1変数のデータが100個あり，これを区間数10のヒストグラムで表現する．すると，**図-2** (a) に示すように多少雑音の乗ったガウス分布が得られる．では同じ100個のデータについて，2次元のヒストグラムを書いてみると，**図-2** (b) のようになる．この図からはガウス分布であることは分からない．データが少なすぎて確率密度関数を再現できないのである．

なんで，こんなことになってしまうのだろう？　答えはヒストグラムの分割数にある．1次元から2次元では次元数は2倍になっただけだが，ヒストグラムの分割数でみると10×10で100になっている．つまり，100個のデータでは分布の表現が困難になってくる．これが3次元では10^3個，一般化してN次元では10^N個の箱を埋めるのに十分なだけのサンプルが必要なことになる．つまり，特徴の次元数を増やせば増やすほど，必要なサンプルの数が指数関数的に増大する．S君が試したカテゴリあたり100個というサンプル数がいかに少ないものになっているか分かるだろう．数百次元の空間で識別問題を考えるときには，100個とか1,000個とかの学習サンプルを用意しても，1，2次元で数個のサンプルを用意したのと変わらないと考えていいのである．これが次元の呪いの恐怖である．

すなわち，パターン認識などに現れる識別問題の本当の定義は，

高次元の空間で少数のサンプルを用いて，最適な識別機械を設計すること．

なのである．

前置きがずいぶん長くなったが，これからS君の研究がなぜうまくいかなかったかを振り返りながら，このことを踏まえた，識別問題を解くための検討過程の一例を説明する．

■識別問題の解き方■

たとえば文字認識などの従来から存在する問題の改良を考える場合や突然与えられた新しい識別問題に関して研究を始める場合について考える．

識別問題を解くにあたっては通常，

データ収集→前処理→特徴抽出→次元削減→識別→評価

の順番で検討を行う．これらの手順はこの通りにいくことは少なく，次の検討に入った段階で前の段階の問題が明らかになることがあるが，とりあえず頭に入れておくとよい順番に挙げた．

以下，順次説明していく．なお，本稿で扱う特徴ベクトルは数値を要素として持つものに限定する．いわゆる質的データの取扱いについては文献4) を参照されたい．

■データ収集■

検討を開始するにあたっては，まずデータを集めなくてはならない．このとき，取り組むべき問題の性質により，どのようなデータを集めるのかを考えることがきわめて重要である．たとえば，「姿勢変動に頑健な

顔認識アルゴリズム」の研究のために正面顔の画像ばかりを集めても研究が始まらないことは明らかであろう．

また，現実の現象を網羅できるだけの十分な量のデータを集めることも重要である．データ収集には必ずコストがかかるものではあるが，自分のリソースの許す限り，できるだけ多量のデータを用意することをお勧めする．

新しい問題に取り組むのではなければ既存のデータベースを利用するのもお勧めである．このときには前の2つの条件を満たすデータを選ぶとともに，そのデータを使った過去の研究と実験条件を合わせることも大事なことである．そうしなくては，手法の公平な比較はできないし，一方で条件を合わせておけば比較実験を省くことができるというメリットもある．

S君の失敗の第1歩がここにあった．IPTP CDROM1を用いた実験では学習，テストデータとも1,000個単位で評価が行われている．このことに気づいていれば，実は彼は失敗を避けられたかもしれない．

■前　処　理■

前処理は，データから雑音等の本質的ではない情報を除去する作業であるが，文字認識やテキスト分類のように特徴ベクトルが顕に与えられない問題の場合とデータマイニングのような数値が直接に与えられる問題では意味合いが異なってくる．

S君の研究テーマである文字認識の問題では，文字の位置や大きさの正規化，ごま塩雑音の除去などが主要な前処理技術として用いられている．

一方，多変量解析やデータマイニングの問題では欠損値，異常値を除去するプロセスであったり，数値の正規化を行うところであったりする．多くの場合，この過程は人手で行われ，多大な時間を必要とする．たとえば，CRM（Customer Relationship Management）と呼ばれる顧客データからの戦略決定の業務では，顧客から得たデータからの欠損値，異常値除去だけで解析時間の過半を消費するという．

どちらの場合でも，ここで手を抜くと後段の検討過程でとんでもない現象を引き起こす可能性があるので決して手を抜かないことを強くお勧めする．

■特徴抽出■

特徴抽出はデータを数値（特徴ベクトル）に変換する過程である．当然のことながら画像認識やテキスト分類では本質的な問題になるが，数値がダイレクトに与えられるデータマイニングなどでは不必要な過程である．特徴抽出は問題の性質に大きく依存するために，一般論を語ることは難しい．あえて一般的な注意を語るとすると，

1．対象の性質を深く研究すること
2．過去の研究を広くサーベイし，従来用いられている特徴抽出器の性質をきちんと理解しておくこと

の2点である．たとえば，文字認識などの画像認識の問題では，画素値をそのまま入力すると非常に大きな次元数になるが，対象に性質に基づいた特徴を用いれば，非常に少ない次元数の特徴で対象をうまく表現することができる．逆に，無思慮に特徴を増やしても，冗長な特徴が増えるばかりで，むやみに特徴次元数を増やし，自ら次元の呪いを呼び込むことになる．

S君の失敗の原因の1つはこの点にある．実は周辺分布と方向線素特徴は双方とも文字線の方向を表現する特徴であって，この2つを組み合わせてもさほど情報は増えていない．S君は特徴の組合せがお手軽研究だと思って始めたわけだが実はそれほどお手軽な研究ではなかったわけである．

■次元削減■

入力された特徴ベクトルの次元数を削減するプロセスである．主として，行列を用いて特徴ベクトルを変換する次元圧縮技術[5]と，テーブルを用いて特徴を選択する特徴選択の技術に分類される．現時点では次元の呪いに対抗するほとんど唯一の技術であり，真剣に検討することをお勧めする．

性能向上を要求するのであれば，主成分分析や判別分析のような次元圧縮技術の方が優れているが，データマイニングの場合のように現象を説明する規則を発見するという意味合いでは，変数を単純に減少させる特徴選択の技術がより本質的である．

次元圧縮技術では主成分分析と判別分析が重要な技術である．主成分分析は，高次元の特徴空間での分布形状を線形の範囲内で最適に近似する方法であり，単純に変数の減少を狙うためには最も素直な方法である．この方法は単に分布を近似して低次元に写像する方法であるため，理論的には識別精度の向上は起こり得ない．しかし，次元の呪いが緩和されるため，現実には識別精度の向上がみられることは珍しくない．

一方，判別分析は積極的に識別に有効な空間に写像する方法であるために，理論的にも実験的にも識別精度は向上する．しかし，セキュリティのための顔識別

問題などのように，学習していないカテゴリの信号が入力されることが想定される場合には，予測不能な挙動を起こすことがあるので注意が必要である．

特徴選択の技術には，基本的には総当たり以外にはよい方法がないことが知られている[6]．識別問題との関連では判別分析に基づく変数選択法[7]が提案されており，簡単なわりには有効であるので試してみることをお勧めする．この場合にも，識別性能が向上することは珍しくない[8]．

S君の失敗の原因のもう1つは，そもそも次元削減を検討しなかったことである．大量の特徴をくっつけることが彼のアイディアであったわけだが，主成分分析や判別分析の方法で冗長な次元を削減できれば彼のアイディアは活きたかもしれない．

■識別器の選び方■

識別器の選択，設計は識別問題の中核部分であり，よりよい識別器を求めて，世界中の研究者が激しい争いを展開している．理論的にも実験的にも興味のある謎がたくさんあり，挑戦すべき課題には枚挙の暇のない領域でもある．

研究的な検討では，(1) 新たな識別器の提案，(2) 最適な識別器の選択，の2つの方向の検討が考えられるが，ここでは初心者の検討で多くみられる (2) について指針を示すことにしよう．

識別器の選定をするときに絶対にやってはいけないことはS君がやったように，「流行の識別器を選択する」ということである．この世界では，新しい方法を使っていないと新しい論文にみえないというような，悪しき習慣があるが，こと識別問題に関しては，こうした態度は致命的な問題を引き起こすことがある．

たとえば，いわゆるニューラルネットやSVMはパターン認識のみならず機械学習の世界でも応用され，これを使えば，簡単に高い性能が得られると期待されがちであるが，これらのように処理量が多く，多数のパラメータを含む識別器を初期段階で用いるのは本質的に誤っている．

少なくとも検討初期には，識別の前処理の正当性の検討も必要となるため，試行錯誤的な実験の繰り返しになる可能性が高い．したがって，処理が軽く，パラメータが少なく，挙動の理解しやすい識別器を使うのが正しい．

著者らの推薦する検討の進め方は，前段の処理の正当性が保証されるまでは，k-最近傍法（以下k-NN法）を用いることである．k-NN法は，学習データのすべてを記憶し，入力データから，すべての学習データまでの距離を計算し，上位k個の中に最も多かったカテゴリを正解とするものである．

この方法は簡単であるため，コーディングが楽で，パラメータも少なく，さらに，データに予期しない非線形性があった場合でも，それなりに性能を出してくる．

前段の処理の正当性が保証されている場合には，分布の正規性を仮定した判別分析や二次識別器のような識別器を簡単な順に用いていく．これらの挙動によりデータの分布に対して，ある程度の感触がつかめる場合がある．

もしも，これらの方法により目標が達成できないことがあれば，そのときがニューラルネットやSVMに関する検討を開始する段階である．このときも，分布の非線形性について何らかの知見が得られない場合には用いない方がよい．苦労しても性能が向上しないことが珍しくないからである．非線形性の根拠が得られない場合には，前段の処理に戻って検討を繰り返す方が着実な成果につながることが多い．

もし，これらを採用する場合には，少なくとも「なぜニューラルネットやSVMを使ったのか？」という質問には答えられるようにしていただきたいものである．

S君の間違いは，識別器の性質に関する考察なしに，人の噂を信じて識別器を採用したことである．彼の誤解は，ニューラルネットでは複雑な識別面を作ることができるから，高い識別性能を達成できると考えたところにある．現実の問題では，学習に利用できるサンプル数にも限りがあり，不必要な複雑さはかえって識別性能を低下させる．

ここまで書いたところで紙数が尽きた．後編では，識別器の選択法について，もう少し語った上で，識別器の評価の仕方と注意すべき点を解説し，その上で人工データと実データを用いた実験で実例を示していくことにしよう．

参考文献
1) 山田敬嗣, 佐藤　敦: ニューラルネットによるパターン認識, 電子情報通信学会誌, Vol.82, pp.852-859, pp.977-984, pp.1046-1053, pp.1248-1255, Vol.83, pp.50-56 (1999-2000).
2) 具体的な文献名はあげないが，90年前後は過学習現象は，学習サンプルの提示し過ぎによる，学習結果の劣化と理解されていた．
3) 前田英作: 痛快!サポートベクトルマシン, 情報処理, Vol.42, No.7 (July 2001).
4) 坂本慶行: カテゴリカルデータのモデル分析, 共立出版 (1985).
5) 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).
6) Jain, A. K., Duin, R. and Mao, J.: Statistical Pattern Recognition: A Review, IEEE, Trans., PAMI, Vol.22, No.1, pp.4-37 (2000).
7) 柳井晴夫, 高根芳雄: 新版多変量解析法, 朝倉書店 (1977).
8) たとえば, 佐藤　新, 末永高志, 坂野　鋭: クラスタ判別法の医療データ解析への応用, 人工知能学会KBS研究会資料, SIG-FAI/KBS-J-39 (11/14) (2001).

(平成14年4月1日受付)