

多クラスサポートベクターマシンにおける 各SVMモジュールの独立特徴選択

胡 欣[†] マウリシオ・クグレ[†] 黒柳 奨[†] 岩田 彰[†]

[†] 名古屋工業大学大学院 工学研究科情報工学専攻 〒466-8555 名古屋市御器所町

E-mail: †huxin@mars.elcom.nitech.ac.jp, ††mauricio@kugler.com, †††{bw,iwata}@nitech.ac.jp

あらまし 近年, サポートベクターマシン (Support Vector Machine, SVM) と呼ばれるパターン識別手法が優れた識別率性能により注目を浴びている. しかし, SVM を応用する場合, 多クラス問題においてその認識スピードが問題となる. これに対して特徴選択 (FSS) により特徴次元数とサポートベクトルの数を削減することで, SVM の認識時間を短縮することができ, リアルタイムの処理が可能になる. また多クラス SVM に対しては全体の入力空間で特徴選択を行うより, 各 SVM モジュールにおいて独立に特徴選択を行うことが有効であろう. そこで本論文では, 多クラス SVM における各 SVM モジュールの独立特徴選択の新しい手法を提案し, 実データを用いて実験を行った. その結果, 識別率を落とすことなく, 特徴次元数及びサポートベクトルの削減により識別時間を元の時間より大幅に削減する効果を得た.

キーワード 特徴選択, サポートベクターマシン, 多クラス SVM, サポートベクトル, 多クラス識別問題, SBS

Splitting the feature subset selection of multiclass support vector machines

Xin HU[†], Mauricio KUGLER[†], Susumu KUROYANAGI[†], and Akira IWATA[†]

[†] The authors are with the Department of Computer Science & Engineering, Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Nagoya, 466-8555, Japan.

E-mail: †huxin@mars.elcom.nitech.ac.jp, ††mauricio@kugler.com, †††{bw,iwata}@nitech.ac.jp

Abstract One drawback on the use of Support Vector Machines (SVM) in real applications is its slow classification speed, proportional to the product of number of features and number of support vectors. Feature Subset Selection (FSS) is one way for reducing the dimensionality, normally reducing the number of support vectors, and consequently the recognition time. However, for the multiclass SVM, applying FSS in the whole input space does not achieve an optimal feature subset for each independent binary classifier. This work proposes a new structure, in which the FSS is performed independently for each SVM. The experiments with real world data showed a much higher average dimensionality reduction, decreasing the recognition time by several orders with a comparable performance with the full features set.

Key words feature subset selection, support vector machine, multiclass SVM, support vectors, multiclass classification problems, SBS

1. はじめに

パターン認識で扱うデータはその特徴次元数が過多である場合, 計算時間が長く, Overfitting による精度の悪化を招く場合がある. このような場合には特徴選択を行うことで, 不要な特徴を取り除き, 分類器の精度の低下を抑え, 計算時間の抑制をすることが可能である.

パターン認識手法の一つとしてニューラルネットワークを認識エンジンとして用いる方法が多く提案されているが, 近年, 非線形分類問題の解決法としてサポートベクターマシン (Support Vector Machine, SVM) が注目を浴びている. SVM は, 線形しきい素子を用い, 2 クラスのパターン識別器を構成する方法である. 学習サンプルからマージン最大化という基準で線形しきい素子のパラメータを学習する. また, 非線形な問

題に対しては特徴ベクトルをカーネル関数により高次元空間へ写像し、その空間で線形分離を行うことにより識別を行う。

SVMは現在知られている多くの手法の中でも最も認識性能の優れた学習モデルの一つであり、文字認識など多くの実験で高い識別率が報告されている。また近年いくつもの実用的な応用事例が提案されており、これらはバイオインフォマティクス（バイオ情報学）、計算言語学、コンピュータビジョンなどの多岐の分野にわたる [1]。しかし、多クラス問題にSVMを応用する場合、多数のSVMモジュールを使用するため、識別スピードが遅いという問題がある。そこで本論文では、多クラスSVMに対して各SVMモジュールの独立特徴選択を行う新しい手法を提案する。

本稿の構成は次のようになっている。まず、2.節では、サポートベクターマシンを述べる。次いで3.節では特徴選択の概要について述べる。4.節では、提案手法を説明する。5.節では、提案手法で実データによる認識実験の結果と考察を示す。

2. サポートベクターマシン (SVM)

2.1 ニクラスの識別を行う SVM

SVMは分離超平面からサンプルまでの距離 (margin) が最大化するように分離超平面を構築しクラス分類を行う。この分離超平面はサポートベクトルと呼ばれるクラス境界近傍に存在するいくつかのサンプルにより表現される。非線形の問題に対しては、カーネルトリックによって入力空間をより高次の特徴空間に写像し、そこで線形分離を行うことで適用が可能である。SVMは統計的学習理論に基づく新しい2クラスのパターン認識手法であり、ニューラルネットワークなどの従来法と比較して汎化能力が高い点と最適解が求まる点に特徴があり、そのため学習に用いていないデータに対しても高い識別率を示す。SVMがこのような特徴を示すのは、その学習に認識誤りと汎化性能の両面から最適化が行われ、これが二次の凸計画問題として定式化されているためである。

認識サンプル (未知なサンプル) \mathbf{x} を識別する場合 SVMの最終的な分類関数 (識別関数) は

$$f(\mathbf{x}) = \text{sgn} \left(\sum_{i \in SV} y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b \right) \quad (1)$$

となる。

ここで、 \mathbf{x}_i は最適な識別超平面上の i 個目のサポートベクトルであり、 y_i は i 個目のサポートベクトルの教師ラベルを意味している。 $K(\mathbf{x}_i, \mathbf{x})$ はカーネル、 α_i はラグランジュ乗数で、 b は閾値である。関数 $\text{sgn}(u)$ は、 $u > 0$ のとき 1 を取り、 $u \leq 0$ のとき -1 をとる符号関数である。

カーネル $K(\mathbf{x}_i, \mathbf{x})$ から最適な非線形写像を構成できる。本研究では次に示す Gaussian カーネルを用いた。

$$K(\mathbf{x}_i, \mathbf{x}) = \exp \left(-\frac{\|\mathbf{x}_i - \mathbf{x}\|^2}{2\sigma^2} \right) \quad (2)$$

ここで、サンプルの特徴次元数を n とすると

$$\|\mathbf{x}_i - \mathbf{x}\|^2 = \sum_{n=1}^N (x_{i_n} - x_n)^2 \quad (3)$$

となる。

式 (3) により、Gaussian カーネルを計算する場合には乗算、減算、加算の回数がそれぞれ特徴次元数の数 N になり、計算量は特徴次元数と比例することが明らかである。そして、式 (1) 識別関数により、SVMでパターン識別する場合、識別時間はサポートベクトルの数と特徴次元数の積に比例する。 N を特徴次元数、 SV をサポートベクトルの数にすると計算量は $O(N \cdot SV)$ である。

研究の目的は多クラス SVMにおける各 SVMモジュールの独立特徴選択することにより、サポートベクトルのサンプル数と特徴次元数を減らし、より多く識別時間の削減を図る。

2.2 SVMの複数クラス化

多クラス問題に対するSVMの適用法について述べる。SVMは二クラス識別器であるため、多クラス (Q クラス) 問題に直接適用することはできない。二クラス識別器を多クラス問題に適用する典型的な方法としては、あるクラスとその他のクラス間で識別を行う識別器を学習して用いる One Versus the Rest (1-v-r) や、すべてのクラス対ごとに識別器を構成し、それらを組み合わせる One Versus One (1-v-1) がある。 [2] [3]

1-v-r の場合は Q クラスデータを学習する場合、すべてのデータを用いた $SV M_i, (i = 1, 2, \dots, Q)$ を構築する必要がある。その結果を統合するというものである (最大の $f(\mathbf{x})$ を出力したクラスに識別する)。この方法はクラス数とサンプル数が多いときは計算量が非常に多い。

1-v-1 の場合は Q クラスデータを学習するためには、学習に2クラス分のデータを用いた $SV M_i, (i = 1, 2, \dots, Q(Q-1)/2)$ 個のSVMが必要となる。ゆえに、1-v-1は構築しなければならないSVMの個数は1-v-rより多くなるが、それぞれのSVMを構築する際に用いるデータ数は1-v-rより少ないため、1-v-rよりも高速な学習が可能であることが知られている。

本研究では、1-v-1の手法は1-v-rの手法と比較した場合に高速な学習が可能であること、またそれぞれのSVMは二クラス分類問題を担当するので、問題がシンプルであり、次元数をより多く削除できると期待できるため1-v-1の構成法を用いた。

3. 特徴選択

一般的にパターン認識に用いるデータベースは、事例とそれを表す特徴からなる。特徴の過多、計算時間が長くなり、計算コストが高いという問題点がある。パターン認識においては、良い性能をもつ識別子 (特徴空間を分割する規則) を作るには非常に多くの学習パターンを必要とする。一般に、特徴次元数が増えると必要な学習パターンの数は指数的に増加する [4]。学習パターンを得るには高いコストがかかるため、得られる学習パターンの個数は限られる。逆に、識別に有効な情報を得るには、特徴数が多くなる傾向がある。この結果、特徴数の増加につれ、識別子の性能が逆に低下する次元の呪いに陥る。特徴選択 (FSS - Feature Subset Selection) とは、与えられた特徴集合から識別器械にとってよりよい特徴集合の候補を与えるものである。元の \mathbf{D} 個の特徴を持つ特徴集合 \mathbf{S} から識別に有効な

Initialize:Subset of remaining features $\mathbf{s} = [1, 2, \dots, n]$ Train the classifier with the remaining features \mathbf{s} **repeat**(a) **for** each of the remaining features **do** compute $RR^{(i)}$ /* $RR^{(i)}$ is the recognition rate without i^{th} feature*/(b) Find the worst feature x $x = \arg \max (RR^{(i)})$ (c) Remove the feature that maximizes RR $\mathbf{s} = [1, \dots, x - 1, x + 1, \dots, n]$ until \mathbf{s} is empty.

図1 識別率を評価基準とする SBS アルゴリズム

情報をもつ d 個の特徴からなる部分集合 \mathbf{s} を探す処理である (ただし, $D > d$ で d はできるだけ小さい数). この評価関数の出力値 $J(\cdot)$ が高いほど良い特徴集合であると言える. $J(\mathbf{s})$ の最大問題である. 評価基準関数の一つとして識別率 $1 - P_e$ (P_e : 誤り率) を用いることができる.

これまでに様々な特徴選択の手法が提案されている. Doak [5] は特徴選択のプロセスの視点から見ると最善的 (Exhaustive), 乱数的 (Randomized) と連続的 (Sequential) の三つのカテゴリに分けられた. 連続的な特徴選択, たとえばよく知られている SFS (Sequential Forward Selection) と SBS (Sequential Backward Selection) は比較的安価で, 応用アプリケーションにおいてはよく使われている. SFS は特徴数が 0 の状態から特徴次元数を増やし, 停止基準に満たすまでよりよい評価関数を最大化していく方法である. SBS は全部の特徴から特徴数を減らしていく方法である.

これ以外に Pubdil ら [6] により提案された継続的 Floating アルゴリズム SFBS (Sequential Floating Forward Search) と SBFS (Sequential Floating Backward Search) がある. この方法は SBS と SFS といった一方向だけの探索アルゴリズムより高いパフォーマンス得られるが [7], 計算量が増加するため, 特に特徴次元数が 100 を超えるような大規模な問題に対して計算時間が膨大になる.

そこで, 本研究は実現の簡単さと計算時間を配慮し, SBS のアルゴリズムを使用することにした. SBS アルゴリズムは全部の特徴から一個ずつ特徴を取り除いていくことで, 最も良い特徴の組を選ぶ. 図 1 は SBS をベースで, 評価基準を識別率とし特徴選択を行うアルゴリズムである.

4. 提案手法

パターン認識に要する計算コストを削減する方法としては図 2 に示すように, 全体の入力空間で特徴選択を行い, その結果を識別器に入力する方法が多く用いられている. 識別器として SVM を用いる場合においても, 特徴選択の後, 多クラス SVM を用い認識させる方法が提案されている [8] [9] [10]. この手法は全体の入力空間で特徴選択を行うため, ここではグローバル

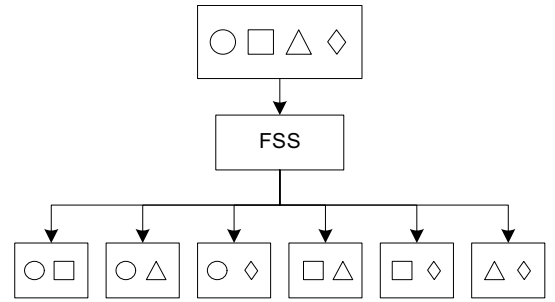


図2 グローバル特徴選択

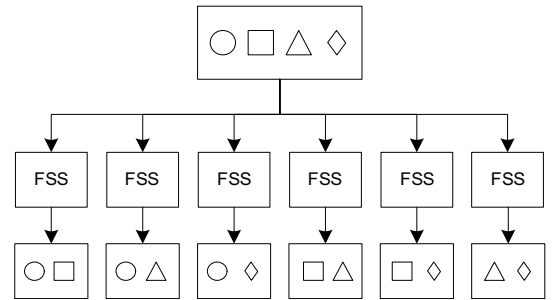


図3 独立特徴選択

特徴選択 (Global feature selection) と呼ぶことにする. この方法は識別器として階層型ニューラルネットワークや Nearest Neighbor 法を用いる場合などにも汎用的に適用可能な方法である. しかし多クラス SVM に関しては, 識別器が実際には多くの二クラス分類器から構成されているため, これら二クラス分類器それぞれに対して特徴選択を行うことにより各部分空間に最適化した効率のよい特徴選択が可能になることが考えられる.

そこで本論文では, 多クラス SVM の 1-v-1 のアルゴリズムを用い, 各クラス対ごとに識別器を構成し, 各 SVM モジュールにおいて特徴選択を行う方法を提案する. これを独立特徴選択 (Split feature selection) と呼ぶことにする. 図 3 に提案手法を示す. 空間上の $\circ, \square, \triangle, \diamond$ 4 種類のクラスのデータを分類することを考える. まず, 多クラス SVM は 1-v-1 アルゴリズムでクラス対ごとに識別機を構成する. それぞれ $[\circ, \square], [\circ, \triangle], [\circ, \diamond], [\square, \triangle], [\square, \diamond], [\triangle, \diamond]$ の六つの識別器を構成する. 構成された各 SVM で識別率を評価基準とし, SBS のアルゴリズムで特徴選択を行う. 各 SVM モジュールにおける特徴選択のアルゴリズムを図 4 に示す.

5. 認識実験

実験に用いたデータベースは, UCI Machine Learning Repository [11] (Satimage, Segment, Pendigits, Vehicle) から入手した機械学習用データベースをアイテム集合のトランザクションとして取り扱えるように変換したものである. 表 1 に各データベースの仕様を示す.

実験データを作成するために, 分類対象となるデータを 2 分割または 3 分割する. 分割された各データは, モデルの評価を行う目的で利用される. データサンプルが十分な場合には学習用データ (training data), 評価用データ (control data) とテス

表 1 元のデータベース

Database	Training	Test	Classes	Features
Satimage	4435	2000	6	36
Segment	210	2100	7	18
Pendigits	7494	3498	10	16
Vehicle	846	-	4	18

表 2 実験に用いた分割したデータベース

Database	Training	Control	Test	Classes	Features
Satimage	3105	1330	2000	6	36
Segment	770	770	770	7	18
Pendigits	7494	1998	1500	10	16
Vehicle	306	270	270	4	18

ト用データ (test data) の三つに分割することがもっとも適切である [10]. 評価用データは推定されたモデルの妥当性を確認するために使用する.

本研究で用いたデータベースは元のデータベースを各クラスサンプルのバランスを取るように、それぞれ学習、評価、テストという三つの新しいデータに分割した. Satimage データベースの学習データベースを新しく学習データと評価データに分割した. Pendigits データベースのテストデータを新しく評価データとテストデータに分割した. Segment データベースは元の学習データとテストデータを結合し、平均に三つのデータを分割した. Vehicle データベースは学習データしかなかったため、元の学習データを三つのデータを分割した. 元のデータベースを分割して得た新しいデータベースは表 2 に示す.

本実験においては、まず特徴選択を行わない多クラス SVM で、評価データの識別率を基準として識別率が一番高くなるパラメータを求める. 次に、得られたパラメータで各データベースの特徴選択をグローバル特徴選択と独立特徴選択で行う. そして、特徴選択を行ったデータベースにおける評価データを基準とし、再びパラメータを調節する. 最後に、全部の特徴を用いたデータベースと、グローバル特徴選択及び独立特徴選択を行ったデータベースで、適切なパラメータ (評価データで求

(a) for each SVM⁽ⁱ⁾

Initialize:

Subset of remaining features $s_i = [1, 2, \dots, n]$

Train the i^{th} classifier with the remaining features s_i

repeat

(b) for each of the remaining features

do compute $RR^{(j)}$

/* $RR^{(j)}$ is the recognition rate without j^{th} feature*/

(c) Find the worst feature x of the i^{th} SVM

$x = \arg \max (RR^{(j)})$

(d) Remove the feature that maximizes $RR^{(j)}$

$s_i = [1, \dots, x - 1, x + 1, \dots, n]$

until s_i is empty.

図 4 各 SVM モジュールの特徴選択

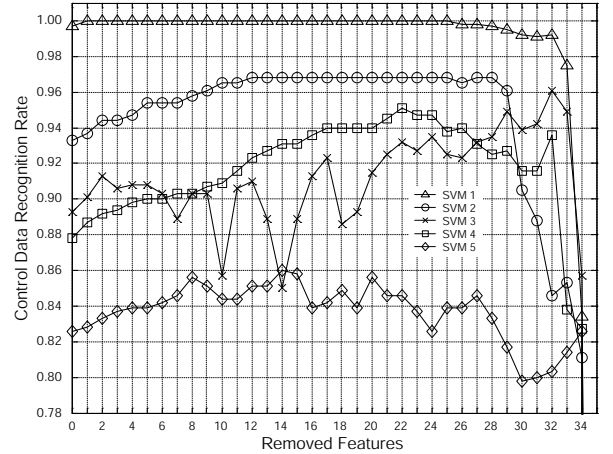


図 5 独立特徴選択の実例 (Satimage)

められたパラメータ) でテストデータにおける識別率と識別時間を求めた.

多クラスサポートベクターマシンにおける各二クラスの独立特徴選択の実例を挙げる. この実例は Satimage データを用いたものである. Satimage データベースはクラス数は 6 で、特徴次元数は 36 である. 1-v-1 アルゴリズムで 15 個 SVM 識別器が構成する. その中の 5 つの SVM の特徴選択を図 5 に示す. 図の横軸は削除した特徴の数で、縦軸は評価データの識別率である. 5 つの SVM はそれぞれ SVM1, SVM2, ..., SVM5 で表示する. 各 SVM モジュールの最高識別率においては SVM5 は 86% で、SVM1 は 100% である. そして、最高識別率を得られる特徴次元数は SVM3 が 4 個の特徴次元数に対し、SVM5 は 22 個特徴次元数が必要になる. このことから、多クラス問題において、各 SVM モジュールによって評価データの最高識別率は異なり、最高識別率が得られるときの特徴次元数も違うことが分かる. そして、SVM1 と SVM2 では識別率が高く、特徴次元数を削減しても識別率の変化は滑らかである. SVM3, SVM4 では用いたデータの識別が困難であるため、識別率が低く一つの特徴の取り除くだけで識別率が大きく変化している. 以上のことから、各 SVM モジュールの特徴選択はまったく違うことが明らかである. このため、多クラス SVM における各 SVM モジュールの独立特徴選択が必要である.

図 6 と図 7 は実験結果を示す. 図 6 と図 7 の横軸はいずれもデータベースの名前であり、縦軸はそれぞれ評価データの識別率とテストデータの識別率である. 図 6 に示すようにグローバル特徴選択と独立特徴選択を用いた場合では全部の特徴次元数を用いた場合と比較すると、グローバル特徴選択の評価データの識別率は全部の特徴次元数を用いた場合より 0.1%~1.95% 高くなった. また独立特徴選択の評価データの識別率は全部の特徴次元数を用いた場合より 1%~7.52% 高くなった. よって評価データに対しては独立特徴選択を行った場合、全部の特徴を用いた場合とグローバル特徴選択を行った場合より高い識別率を得られることが明らかになった.

図 7 では、グローバル特徴選択と独立特徴選択は全部の特徴を用いた場合よりそれぞれ -1.15%~1.85%, -1.9%~4.68% 高

い、グローバル特徴選択は Satimage と Segment データベースにおいては、特徴選択を行わない場合よりテストデータの識別率はやや低く、Pendigits と Vehicle データベースにおいて、識別率が高い。独立特徴選択は Satimage と Pendigits データベースにおいて識別率が低くなる傾向が見られたが、Segment と Vehicle データベースにおいてはより高い識別率が得られた。すなわち、グローバル特徴選択と独立特徴選択はテストデータの識別率に大きな影響を与えないと言える。

本研究の目的は独立特徴選択により、識別時間を縮小することである。認識時間は 2 節に前述したように SVM の計算量はサポートベクトルの数と特徴次元数の積と比例する。そこで各データベースで特徴選択を行わない場合の識別時間 ($O(N \cdot SV)$) を 1 とし、グローバル特徴選択と独立特徴選択を行う場合の識別時間を求める。テストデータの識別時間を図 8 に示す。

四つのデータベース Satimage, Segment, Pendigits, Vehicle でグローバル特徴選択は全部の特徴を用いた場合の認識時間のそれぞれ 14%, 44%, 60%, 93% に対し、独立特徴選択の認識時間はそれぞれ 6%, 18%, 9%, 67% である。Satimage, Segment データベースにおいて、独立特徴選択はグローバル特徴選択より半分以上の認識時間を削減できた。Pendigits データベースにおいては独立特徴選択の手法を用いた場合、全部の特徴を用いたときと比較するとわずか 1/10 の識別時間になり、グローバル特徴選択より 85% の識別時間を減らすことができた。このことから、独立特徴選択はグローバル特徴選択より多くの特徴を削減でき、識別時間を縮小できることが明らかになった。

また、Satimage と Vehicle データベースにおいて、グローバル特徴選択と独立特徴選択を用いた場合、Vehicle データベースより Satimage データベースのほうが、多くの特徴を削減したことが明らかである。これは Satimage データベースは Vehicle データベースより多くの特徴次元数を持っているためと考えられる。さらに、Pendigits データベースにおいては、ほかのデータベースと比較すると、独立特徴選択はグローバル特徴選択より多くの特徴を削減したことがわかる。これは Pendigits データベースはほかのデータベースより多くのクラスがあるため、全体がより多くの SVM モジュールに分割され、独立特徴選択を行うことで多くの特徴を削減できたと考察できる。

以上の結果より、独立特徴選択は特徴選択を行わない場合とグローバル特徴選択を行う場合のいずれよりも多くの特徴次元数とサポートベクトルを減らし、テストデータの識別率を保ちながら識別時間を大幅に削減することが確認できた。

6. ま と め

本論文では、多クラスサポートベクターマシンにおける各二クラスの独立特徴選択という新しい手法を提案し、SBS の特徴選択をベースに識別率を評価値として四つのデータベースを用い実験を行った。独立特徴選択は特徴選択を行わない場合とグローバル特徴選択と比較するとより高い評価データの識別率を得られ、かつテストデータの識別率はほぼ変化しない。独立特徴選択を用いることで、評価データの識別率は全部の特徴を用いた場合より 1%~7.52% 高くなった。そしてテストデータの識

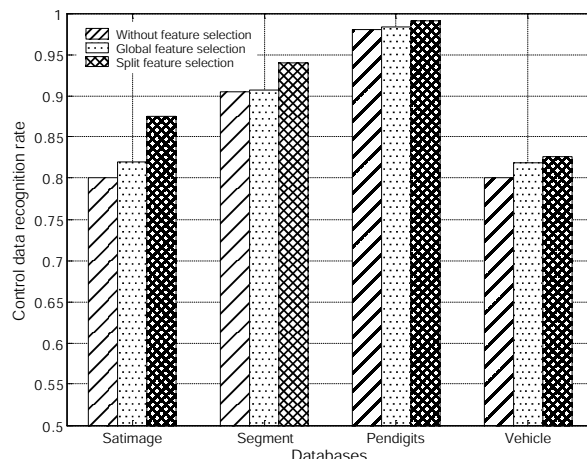


図 6 評価データ識別率

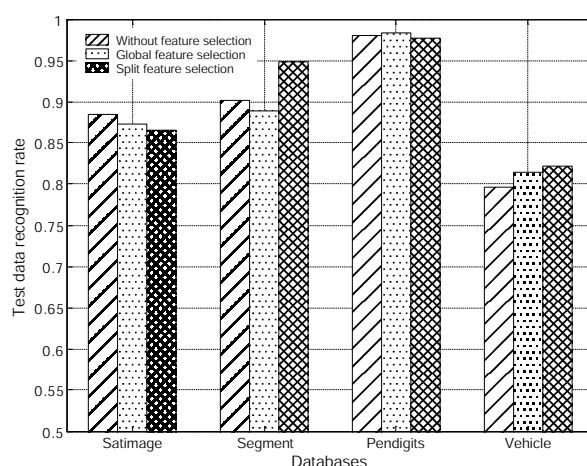


図 7 テストデータ識別率

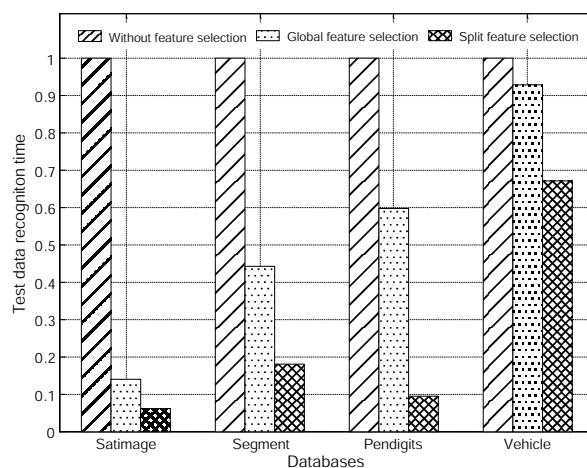


図 8 テストデータの識別時間

別時間においては特徴選択を行わない場合より 6%~67% に削減でき、グローバル特徴選択を行う場合と比較しても 28%~85% に削減され、独立特徴選択の効果が明らかとなった。また実験結果より、より多くのクラスと特徴を有するデータベースに対して、独立特徴選択が有効であることが考えられる。このため本手法は手書き漢字などの多クラス、多次元のデータベースへの応用も有効であろう。

提案手法はシンプルな手法である識別率を評価基準としたが、他の評価基準なども考えられる。また本研究は特徴選択はSVMを用いたラッパーモデル [12] であり、直接的にパフォーマンスが最大となる特徴セットを求めることができたが、学習時間が長いという問題点がある。今後は、本手法を前処理としての特徴選択であるフィルタモデルとの組み合わせに拡張することで、さらに学習に要する計算コストを削減する予定である。

謝 辞

第一著者は私費外国人留学生学習奨励費、日本学生支援機構の援助を受けている。また第二著者は日本政府(文部科学省)奨学金と堀情報科学振興財団の援助を受けている。

文 献

- [1] N. Cristianini, J. Shawe-Taylor, An introduction to support vector machines and other kernel-based learning methods, Cambridge University Press, Cambridge, 2000.
- [2] 丸山 健一, 丸山 稔, 宮尾 秀俊, 中野 康明, “SVMを用いた高累積正解率を持つ多クラスパターン識別,” 情報処理学会論文誌, vol.44, pp.2270-2273, 2003.
- [3] C.-W. Hsu, C.-J. Lin, “A comparison of methods for multi-class support vector machines,” IEEE Transactions on Neural Networks, vol.13, no.2, pp.415-425, 2002.
- [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, オーム社出版局, 東京, 1998.
- [5] J. Doak, “An evaluation of feature selection methods and their application to computer security,” University of California, Department of Computer Science, Techninal Report CSE-92-18, 1992.
- [6] P. Pudil, J. Novovicova, J. Kittler, “Floating search methods in feature selection,” Pattern Recognition Letters, vol.15, no.11, pp.279-283, 1994.
- [7] A. Jain, D. Zongker, “Feature selection: evaluation, application and small sample performance,” IEEE Transactions Pattern Analysis and Machine Intelligence, vol.19, no.2, pp.153-158, 1997.
- [8] J. Brank, M. Grobelnik, N. Milic-Frayling, D. Mladeni, “Feature selection using linear support vector machines,” Techninal Report MSR-TR-2002-63, Microsoft Research, 2002.
- [9] M. Kudo and P. Somol and P. Pudil and M. Shimbo and J. Sklansky, “Comparison of classifier-specific feature selection algorithms,” Pattern Recognition, vol.33, 25-41, 2000.
- [10] J. Reunanen, “A Pitfall in Determining the Optimal Feature Subset Size,” Proceedings of the 4th International Workshop on Pattern Recognition in Information Systems (PRIS2004), Porto, pp.176-185, 2004.
- [11] D. J. Newman, S. Hettich, C. L. Blake and C.J. Merz, “UCI Repository of machine learning databases,” University of California, Irvine, Department of Information and Computer Sciences, 1998, <http://www.ics.uci.edu/~mllearn/MLRepository.html>
- [12] R. Kohavi, G. John, “Wrapper for feature subset selection,” Artificial Intelligence, vol.97, no.1-2, pp.273-324, 1997.