# 分割統治による Support Vector Machine の学習の高速化

山森 一人[1)] 岩﨑 隆[2)] 吉原 郁夫[3)] 相川 勝[4)]

# High-Speed Learning of Support Vector Machine by Divide and Conquer Method

Kunihito YAMAMORI Takashi IWASAKI Ikuo YOSHIHARA Masaru AIKAWA

Abstract

Support Vector Machine (SVM) is a kind of pattern classification method. Because learning of SVM is based on quadratic optimization, SVM can easily handle large scale problems with many training examples. But learning of SVM takes a long time for such large problems. This paper proposes a new method to achieve fast learning for SVM by dividing training examples. In addition, faster learning will be achieved by parallel implementation and using of alternative kernels for dividing process and conquer process. Simulation result show that own method achieves faster learning with keeping the same accuracy than that of conventional SVMs.

Key Words:
Support Vector Machine, Divide and Conquer, Pattern Classification, Kernel, Parallel

## 1 はじめに

自然言語処理やバイオインフォマティクスなどの分野では，ある特徴に基づいて膨大なサンプル集合を分類する必要があることが多い．こうしたパターン分類のため，自然言語処理分野では SVM(Support Vector Machine) が多く利用されている [1)]．

SVM の利点として，過学習を起こしにくくテストサンプルに対する優れた汎用能力を持つこと，二次計画問題を解くことにより学習を行うので最適解が一意に定まり，局所解の問題が存在しない利点がある．しかし，大規模な問題になると学習にかかる計算量が爆発的に増えることが欠点として挙げられている．SVM における学習を高速化するため，現在までに多くの手法が提案されている [2)]．しかしこれらの高速化手段を適用したとしても，実用規模の問題ではサンプルの数やサンプルとなるベクトルの次元数がさらに大きくなることを考えると，計算時間の一層の短縮は重要な課題である．

本研究では SVM における学習の高速化を目的とし，学習で使用するサンプルに対し分割統治を適用することで学習時間を削減する手法を提案する．SVM の学習は二次計画問題に帰着するため，サンプルの組み合わせ数を減らすことで計算量を減らすことができる．また，分割したサンプルから並列に SV(Support Vector) を抽出することで学習時間をさらに短縮する．

## 2 Support Vector Machine

### 2.1 線形分離可能な場合における SVM の学習

$n$ 個の学習サンプル集合 $\boldsymbol{x}_i (i = 1, \cdots, n)$ を 2 つのクラス $C_1$ と $C_2$ に分類する場合，識別境界を定める識別関数 $f(x)$ は (1) 式で表される．

$$f(x) = sign(g(x)) = sign(\boldsymbol{w}^t \boldsymbol{x} + b). \quad (1)$$

ここで $\boldsymbol{w}, b$ は識別関数を決定するパラメータである．サンプル $\boldsymbol{x}_i$ を (1) 式に代入することにより+1，-1 の出力が得られ，$\boldsymbol{x}_i$ の属する正解クラスラベルを $y_i (i =$

[1)]情報システム工学科准教授
[2)]情報システム工学科学生，現在日本電気 (株)
[3)]情報システム工学科教授
[4)]工学部教育研究支援技術センター

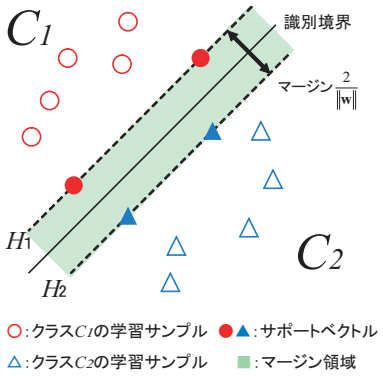


図. 1 線形分離可能な場合の識別境界とサポートベクトル

$1, \cdots, n)$ とすると，

$$\forall_i, g(x) = \boldsymbol{w}^t \boldsymbol{x}_i + b \begin{cases} \geq 1, & \boldsymbol{x}_i \in C_1, \\ \leq -1, & \boldsymbol{x}_i \in C_2, \end{cases} \tag{2}$$

$$y_i = \begin{cases} 1, & \boldsymbol{x}_i \in C_1, \\ -1, & \boldsymbol{x}_i \in C_2, \end{cases} \tag{3}$$

と定義できる．(2) 式，(3) 式より，(4) 式が得られる．

$$\forall_i, y_i \cdot (\boldsymbol{w}^t \boldsymbol{x}_i + b) - 1 \geq 0. \tag{4}$$

(4) 式が満たされているということは，図 1 で示すように $K$ 次元平面 $H_1 : (\boldsymbol{w}^t \boldsymbol{x} + b) = 1$ と $H_2 : (\boldsymbol{w}^t \boldsymbol{x} + b) = -1$ の間にはサンプルが存在しないことを示し，全てのサンプルが完全に分離されていることを意味する．このとき，$K$ 次元平面 $H_1$，$H_2$ 間の距離はマージンと呼ばれ，その距離は $\frac{2}{\|\boldsymbol{w}\|}$ となる．マージンを最大化するためには $\| \boldsymbol{w} \|$ を最小化すれば良く，問題を扱いやすくするために $\frac{1}{2} \| \boldsymbol{w} \|^2$ を最小化することとし，これを目的関数とする．SVM はこの最小化問題をラグランジュ未定乗数法によって双対問題に帰着させて解く．ラグランジュ未定乗数 $\alpha (\geq 0)$ を用いて，(4) 式の制約条件の下で目的関数を (5) 式に書き換える．

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} \| \boldsymbol{w} \|^2 - \sum_{i=1}^{n} \alpha_i [y_i (\boldsymbol{w}^t \boldsymbol{x} + b) - 1]. \tag{5}$$

(5) 式をパラメータ $\boldsymbol{w}, b$ に関して偏微分し 0 とおくと，(6) 式，(7) 式が得られる．

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\alpha})}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i = \boldsymbol{0}, \tag{6}$$

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\alpha})}{\partial b} = \sum_{i=1}^{n} \alpha_i y_i = 0. \tag{7}$$

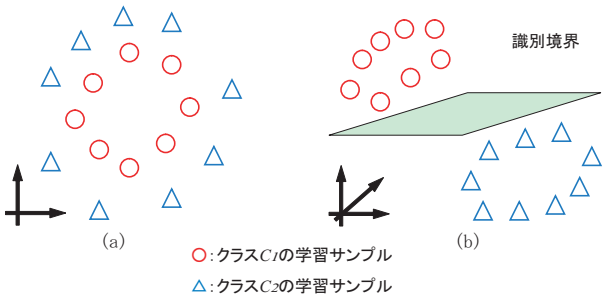


図. 2 カーネルトリック

(6) 式と (7) 式を (5) 式に代入すると，

$$L(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^t \boldsymbol{x}_j, \tag{8}$$

$$s.t. \quad \forall_i, \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0,$$

となり，$L(\boldsymbol{\alpha})$ を最大化する双対問題が得られる．(8) 式を最大化する $\alpha_i$ を $\alpha_i^*$ とおく．$\alpha_i^*$ が 0 ではないとき，つまり $\alpha_i^* > 0$ となるサンプル $\boldsymbol{x}_i$ は SV(Support Vector) と呼ばれ，2 つの $K$ 次元平面 $H_1$，$H_2$ のどちらかの上に存在している．一般的に SV の数は元の学習サンプル数 $n$ に比べ少なくなる．つまり，学習サンプルの中から SV を見つけ出すことによってパラメータは $\boldsymbol{w}, b$ を決定することができる．パラメータ $\boldsymbol{w}$ の最適値 $\boldsymbol{w}^*$ は双対問題の最適解 $\alpha_i^*$，および (6) 式と (7) 式から，

$$\boldsymbol{w}^* = \sum_{i=1}^{n} \alpha_i^* y_i \boldsymbol{x}_i, \tag{9}$$

として得られる．また，パラメータ $b$ の最適値 $b^*$ は，非線形計画法において相補性条件と呼ばれる $\forall_i, \alpha_i^* [y_i \cdot (\boldsymbol{w}^{*t} \boldsymbol{x}_i + b^*) - 1] = 0$ が成立するので，任意の $\boldsymbol{x}_s (s \in SV)$ より，

$$b^* = y_s - \boldsymbol{w}^{*t} \boldsymbol{x}_s, \tag{10}$$

として得られる．最終的に線形分離可能な SVM の識別関数は (11) 式で表現される．

$$\begin{aligned} f(x) &= sign(\boldsymbol{w}^{*t} \boldsymbol{x} + b^*), \\ &= sign\left( \sum_{i \in SV} \alpha_i^* y_i \boldsymbol{x}_i^t \boldsymbol{x} + b^* \right). \end{aligned} \tag{11}$$

つまり，SVM はマージンの最大化という基準から，$\alpha_i^* = 0$ となる識別境界付近の SV が自動的に選ばれて識別関数が構成される．

## 2.2 カーネルトリック

図 2(a) のようなサンプル集合は線形分離できない．そこで，学習サンプル集合を非線形変換し，図 2(b) の

ように新たな空間に写像して線形の識別を行うカーネルトリックと呼ばれる方法が用いられる．写像後のサンプルを $\phi(\boldsymbol{x})$ で表すと，(11) 式は，

$$\begin{aligned} f(x) &= sign(\boldsymbol{w}^t\phi(\boldsymbol{x})+b), \\ &= sign\left(\sum_{i=1}^{n}\alpha_i y_i \phi(\boldsymbol{x}_i)^t\phi(\boldsymbol{x})+b^*\right), \end{aligned} \quad (12)$$

のように書き直せる．写像後のサンプル $\phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j)$ の内積が学習サンプル $\boldsymbol{x}_i$，$\boldsymbol{x}_j$ のみから計算できるならば，$K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ によって最適な非線形写像を構成できる．このような $K$ をカーネルと呼ぶ．これにより高次元に写像された空間での特徴計算を避け，カーネルの計算のみで識別関数が構成できるカーネルトリックを用いることで目的関数である (8) 式は，

$$L(\boldsymbol{\alpha}) = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j), \quad (13)$$

に置き換えることができ，また識別関数の (12) 式は，

$$f(x) = sign\left(\sum_{i=1}^{n}\alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x})+b^*\right), \quad (14)$$

となる．よく使用されるカーネルとしては，(15) 式で表される $d$ 次の多項式カーネルがある．

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = (1+\boldsymbol{x}_i^t\cdot\boldsymbol{x}_j)^d. \quad (15)$$

最適な識別関数を構成できるカーネルは与えられる学習サンプルに依存するので，事前に実験を通じて決定しておく必要がある．

## 3 分割統治による SVM の高速化

### 3.1 従来手法における高速化

SVM の学習は (8) 式に示したように二次計画問題を解くことに帰着し，単純な SVM の学習の計算量は $O(n^2)$ である．学習サンプル数が多い実用規模の問題では，サンプルの組み合わせ数が爆発的に増え，膨大な計算時間が必要となる．そこで現在までに chunking[3] や Decomposition method[2] などの高速化の手段が用いられている．

### 3.2 学習サンプル集合の分割統治

SVM の学習時間は用いるサンプル数の組み合わせ数に依存するため，数千から数万の学習サンプル数からなる実用規模の問題に対しては従来法では十分ではない．本研究では，Decomposition method に学習サンプル集合の分割統治を組み合わせる手法を提案する．サンプル集合を分割することでサンプルの組み合わせ数を減らし，計算時間を短縮する．また，分割したサンプルから並列に SV を抽出することで計算時間をさらに短縮する．

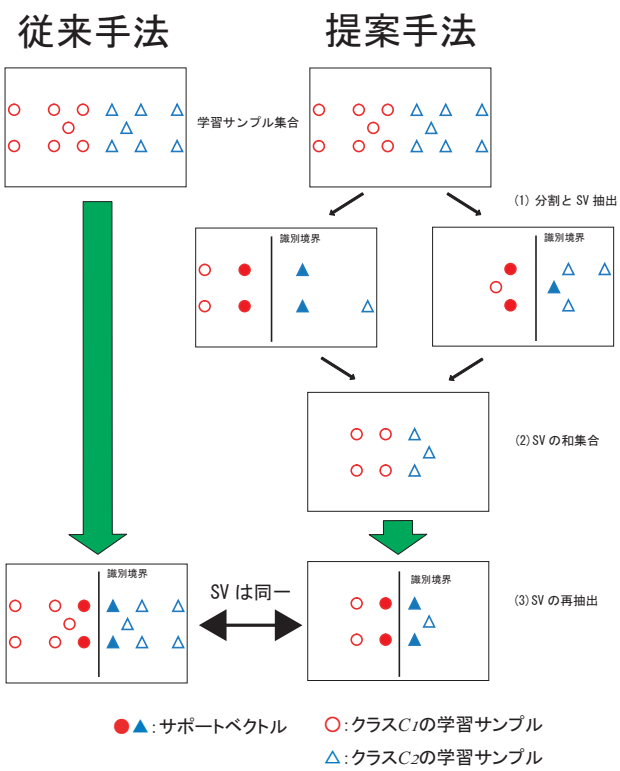


図. 3 従来手法と提案手法における SVM の学習手順

### 3.3 提案手法での SVM の学習手順

従来手法と比較した提案手法の SVM の学習手順を図 3 に示す．従来手法では，図 3 の学習サンプル集合から直接 SV を抽出する．提案手法での SV 抽出の手順を以下に示す．図中の番号は下記の番号と対応する．

(1) 学習サンプル集合を分割し，各サブ集合で SV を抽出する．

(2) 各サブ集合で抽出された SV を合わせ，新たな集合を作成する．

(3) 作成した集合から SV を再抽出する．

線形分離不可能な場合，分割の仕方によって各サブ集合で抽出される SV が変化し，再抽出される SV も異なる．しかし実問題では，十分にサンプル数が多いうえ，明確な識別境界を定めることができない場合が多く，分割の影響を受けにくいと考えられる．次章では代表的な実問題を用いて提案手法の速度と精度について議論する．

表. 1 実験データ詳細

| データ名 | 次元数 | サンプル個数 | 使用カーネル |
|---|---|---|---|
| a6a 学習サンプル | 123 | 11,220 | 三次多項式カーネル |
| a6a テストサンプル | 123 | 21,341 | 三次多項式カーネル |
| a9a 学習サンプル | 123 | 32,561 | 二次多項式カーネル |
| a9a テストサンプル | 123 | 13,281 | 二次多項式カーネル |
| rcv1.binary 学習サンプル | 47,236 | 20,242 | 三次多項式カーネル |
| rcv1.binary テストサンプル | 47,236 | 677,399 | 三次多項式カーネル |

表. 2 実験に使用した計算機の仕様

| ハードウェア | |
|---|---|
| CPU | Intel Xeon 2.8GHz × 2 |
| 2 次キャッシュ | 2MB |
| メインメモリ | 2GB |
| **ソフトウェア** | |
| OS | SUSE Linux 10.1 |
| コンパイラ | Gnu C Compiler 4.1.0 |
| SVM ツール | $SVM^{light}$ 6.01 |

## 4 実験と考察

### 4.1 サンプル集合と実験条件

実験に用いる SVM の実装系として $SVM^{light}$[2] を利用する．実験に用いたサンプル集合は，アメリカ国勢調査データの一部である a6a と a9a[4]，ロイターの記事をカテゴリに分類したベンチマーク用テキスト分類のデータ rcv1.binary[5] を利用する．実験のサンプル集合の主な特徴を表 1 にまとめた．カーネルは事前に $SVM^{light}$ で従来手法による実験を行い，良い精度が得られたカーネルを使用する．実験に用いた計算機環境は表 2 の通りである．

### 4.2 評価方法

本研究で提案する手法を評価するため，提案手法と $SVM^{light}$ の各々の実行時間を計測して比較検討を行う．提案手法は SVM の学習時間を短縮する手法であることから分類時間を含めず，サブ集合からの SV 抽出を 1CPU で逐次的に行った場合の学習時間のみを測定する．さらに，複数 CPU で実行した場合の理想的な並列学習時間を計算により求める．そのため複数 CPU で実行した場合は，SV の再抽出時間に学習サンプルの読み込み時間を含み，通信時間を含まない．

提案手法では従来手法と識別境界の位置が異なる．そのため，分類の精度に対して 2 種類の評価方法により精度の比較検討も同時に行う．まず，学習サンプル自身をどの程度正確に分類できるかを再代入法により評価する．次に，二分割交差検定を用い，学習サンプルには含まれないテストサンプルを分類させてその分類精度を調べる．それぞれの評価方法で，テストサンプルに対し正しく分類されたサンプルの割合を正解率

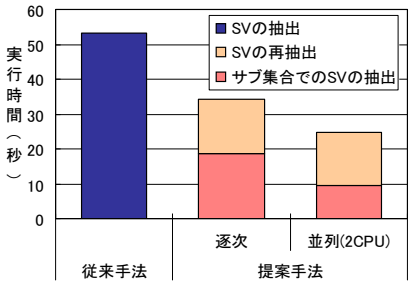


図. 4 a6a での学習時間の比較

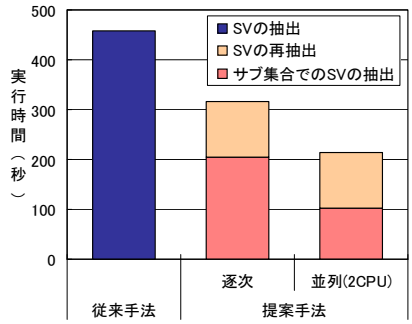


図. 5 a9a での学習時間の比較

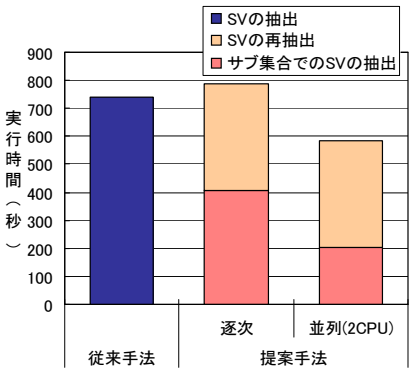


図. 6 rcv1.binary での学習時間の比較

として評価する．サブ集合内には，それぞれのクラスに属する学習サンプルが含まれていなければならないという条件を考慮した上で，サンプル集合をランダムに 2 分割したサブ集合を 10 セット用意し，学習時間，分類精度ともに平均して評価する．

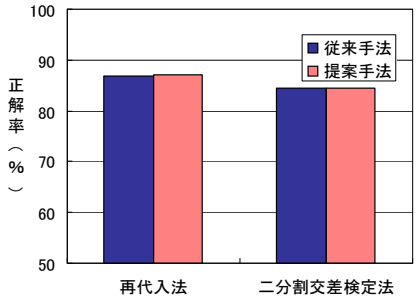


図. 7　a6a での各精度評価法の正解率の比較

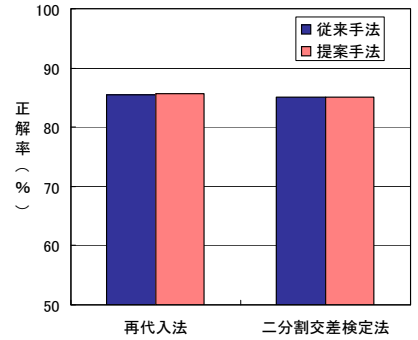


図. 8　a9a での各精度評価法の正解率の比較

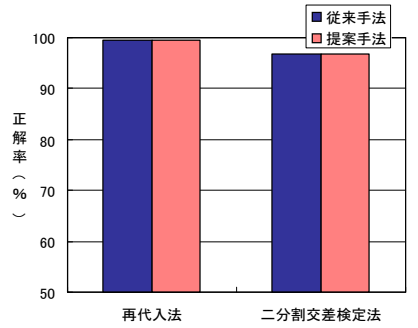


図. 9　rcv1.binary での各精度評価法の正解率の比較

### 4.3　実験結果と考察

サンプル集合 a6a，a9a，rcv1.binary での学習時間と各精度評価法での正解率の比較を図 4〜図 6，図 7〜図 9 にそれぞれ示す．

図 4，5 の結果から，アメリカ国勢調査データの一部である a6a，a9a の提案手法での学習時間は，それぞれ同じような傾向が見られ，逐次に行う場合，従来手法に比べ 1.5 倍の高速化が得られていることが分かる．理想的な 2CPU での並列学習では従来手法に比べ 2 倍の高速化が得られることが期待できる．

図 6 の結果から，テキスト分類のデータ rcv1.binary の提案手法での学習時間は，a6a や a9a とは異なる傾向が見られる．逐次に行う場合，提案手法では従来手法に比べ 1.06 倍遅くなっていることが分かる．しかし，理想的な 2CPU での並列学習では従来手法の 1.3 倍の高速化が得られることが期待できる．

SVM の計算量はサンプルの組み合わせ数が半分になることで，各サブ集合での学習時間は $\frac{1}{4}$ となる．すなわち逐次処理の場合，各サブ集合での学習時間を合わせた時間は，従来手法での学習時間の $\frac{1}{2}$ となる．実際に，実験結果では $\frac{1}{2}$ 以下であり，計算量削減の効果は見られる．しかし SV を再抽出する時間が必要であり，この時間が高速化の妨げとなっている．

また，図 7〜図 9 の結果から，提案手法の分類精度は各精度評価法において，全てのデータで従来手法と同程度を保つことができている．

以上より，実問題では分割統治を用いることによって精度を保ちつつ，SVM の学習の高速化が可能である．

### 4.4　テキスト分類データ rcv1.binary における高速化

rcv1.binary について提案手法の逐次処理でも高速化を実現するため，自然言語処理のテキスト分類はほぼ線形分離可能[6]という性質を利用し，以下の 2 種類の実験を行った．多項式カーネルの次元数をある程度変えても分類精度が変わりにくいことを考慮し，サブ集合での SV の抽出，SV の再抽出それぞれで使用するカーネルの組み合わせを，二次多項式カーネルまたは三次多項式カーネルに変更して実験を行った．また，分割統治により高速化が可能であることがわかったので，分割数を 4 分割に増やした場合の処理速度も調べた．それぞれの精度の評価方法は 4.3 節と同じく再代入法と二分割交差検定法の 2 種類で評価する．

カーネルの組み合わせを変更した場合の逐次処理における学習時間と各精度評価法での正解率の比較を図 10，図 11 にそれぞれ示す．図 10，図 11 の項目名 $k$ 次・$l$ 次はサブ集合からの SV 抽出に $k$ 次，SV の再抽出に $l$ 次の多項式カーネルを用いたことを示す．図 10 の結果から，逐次処理における学習時間は，サブ集合での SV の抽出に二次多項式カーネル，SV の再抽出に二次多項式カーネルを使用した場合が 1.4 倍の高速化となり最も高速である．一方，図 11 の結果から，各精度評価法においての精度は，SV の再抽出に三次多項式カーネルを使用したときに従来手法と同程度の精度を保っている．SV の再抽出に二次多項式カーネルを使用したときには，従来手法と 0.2% という僅かな差だが，精度の低下が見られた．

以上よりサブ集合での SV の抽出に二次多項式カーネル，SV の再抽出に三次多項式カーネルを使用した場合に最も精度を保ちつつ，逐次処理でも 1.3 倍の高

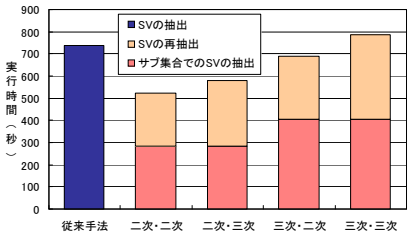


図. 10 rcv1.binary のカーネルの組み合わせを変更した方法での学習時間の比較

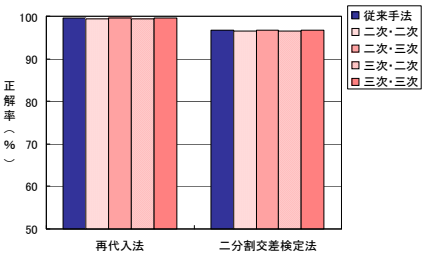


図. 11 rcv1.binary のカーネルの組み合わせを変更した方法での各精度評価法の正解率の比較

速化が得られていることが分かる．

学習サンプル集合を 4 分割した場合での学習時間と各精度評価法での正解率の比較を図 12，図 13 にそれぞれ示す．図 12 の結果から，逐次に行う場合の学習時間は，従来手法より 1.2 倍の高速化が得られていることが分かる．理想的な 4CPU での並列学習では 1.4 倍の高速化が得られることが期待できる．また，図 13 の結果から，各精度評価法においての精度は，従来手法と同程度の精度を保つことができている．

以上よりカーネルの組み合わせを変更することや，分割する数を増やすことによって，分割統治で SVM の学習の高速化が可能であることがわかった．

## 5 おわりに

本研究では SVM での学習時間の短縮を目的とし，学習サンプルを分割することによって小規模な問題に帰着させる分割統治により SVM での学習を高速化する手法を提案した．統計調査やテキスト分類のデータに提案手法を適用したところ，高速化が可能であり，サブ集合を並列化することでさらに高速化が可能であることが分かった．また，カーネルの組み合わせを変更することや分割数を増やすことでも高速化が可能であることが分かった．分類精度について再代入法での正解率と二分割交差検定法での正解率で評価し，どちらの評価方法でも提案手法は従来手法と同程度の精度を示した．以上から，実問題では分割統治による SVM の学習の高速化が可能であることが分かった．

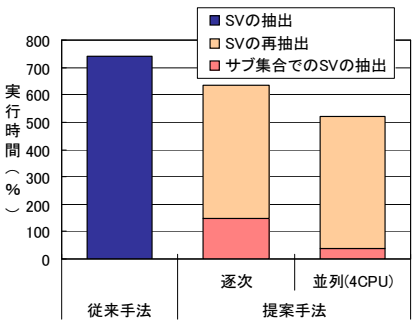


図. 12 rcv1.binary の学習サンプル集合を 4 分割した方法での学習時間の比較

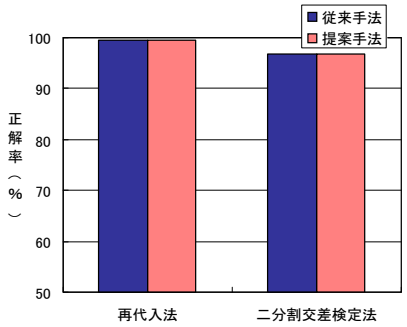


図. 13 rcv1.binary の学習サンプル集合を 4 分割した方法での各精度評価法の正解率の比較

今後の課題として，サンプル数の少ないデータに対して分割しても精度に影響が少ない分割手法の提案や，提案手法のマルチスレッドによる並列化が挙げられる．

## 参考文献

[1] 佐々木裕, 磯崎秀樹, 鈴木潤, 国領弘治, 平尾努, 加賀秀人, 前田英作, “SVM を用いた学習型質問応答システム SAIQA-II”, 情報処理学会論文誌, Vol. 45, No. 2, pp. 635–646 (2004).

[2] T. Joachims, “Making large-Scale SVM Learning Practical”, Advances in Kernel Methods - Support Vector Learning, pp. 169–184, MIT Press (1999).

[3] V. N. Vapnik, “The Nature of Statistical Learning Theory”, Springer (1999).

[4] J. C. Platt, “Fast Training of Support Vector Machines using Sequential Minimal Optimization”, Advances in Kernel Methods - Support Vector Learning, pp. 185–208, MIT Press (1999).

[5] D. D. Lewis, Y. Yang, T. G. Rose and F. Li, “A New Benchmark Collection for Text Categorization Research”, Journal of Machine Learning Research, Vol. 5, pp. 361–397 (2004).

[6] S. Chakrabarti, “Mining the Web: Discovering Knowledge from Hypertext Data”, Morgan-Kauffman (2002).