

# 大規模 SVM 学習のための2段階逐次ワーキングセット選択手法

## Two-stage Incremental Working Set Selection for Support Vector Training

ズン デュック グエン<sup>◇</sup> 松本 一則<sup>◇</sup>  
橋本 和夫<sup>◇</sup> 滝嶋 康弘<sup>◇</sup>  
寺邊 正大<sup>◇</sup>

Dung Duc NGUYEN Kazunori MATSUMOTO  
Kazuo HASHIMOTO Yasuhiro TAKISHIMA  
Masahiro TERABE

本稿では大規模な訓練データに対してサポートベクトルマシン(SVM)の高速学習に適した手法「iSVM」を提案する。通常のdecomposition手法同様、iSVMも最小のワーキングセット(WS)から始めて、訓練サンプルを追加しながらWSを逐次更新していく。iSVMでは学習過程の前半と後半で異なっている。前半では、ランダムに選んだ一定数の訓練サンプルの中から最も有望なサンプルを1個選び出し、WSに追加していく。後半では、既に得たSVMの近似解を用いて、残りの訓練サンプル全部に対して追加対象として適切かどうかの判定を行っていく。訓練サンプル数に比べサポートベクトルの数が少ない問題で特にiSVMの学習効率が良いことが分かった。KDD-CUP 1999の500万件のサンプルで検証した結果、libSVMの分類精度が92.5%で処理時間が25,200秒なのに対し、提案したiSVMは94.1%の精度で処理時間が2,919秒と8倍以上高速であった。

We propose iSVM - an incremental algorithm that achieves high speed in training support vector machines (SVMs) on large datasets. In the common decomposition framework, iSVM starts with a minimum working set (WS), and then iteratively selects one training example to update the WS in each optimization loop. iSVM employs a two-stage strategy in processing the training data. In the first stage, the most prominent vector among randomly sampled data is added to the WS. The second stage uses temporal solutions to scan through the whole training data once again to find the remaining support vectors (SVs). We show that iSVM is especially efficient for training SVMs on applications where data size is much larger than number of SVs. On the KDD-CUP 1999 dataset with nearly five millions training examples, iSVM takes 2,919 sec to train an SVM with 94% testing accuracy, compared to 25,200 sec with LibSVM.

<sup>◇</sup> 正会員 KDDI 研究所 [dd-nguyen@kddilabs.jp](mailto:dd-nguyen@kddilabs.jp)  
<sup>◇</sup> 非会員 KDDI 研究所 [{matsu,takishima}@kdbl.is.ocha.ac.jp](mailto:{matsu,takishima}@kdbl.is.ocha.ac.jp)  
<sup>◇</sup> 非会員 東北大学大学院情報科学研究科 [{kh,terabe}@aiet.ecei.tohoku.ac.jp](mailto:{kh,terabe}@aiet.ecei.tohoku.ac.jp)

## 1. はじめに

近年、サポートベクトルマシン(SVM)は種々の学習アプリケーションに適用されている[1, 2]. しかし、SVMの学習を行なうためには2次計画問題(quadratic programming (QP) problem)を解く必要があり、 $I$  を訓練用サンプルの数とすると、最悪の場合、計算時間は $O(I^2)$ 、メモリは $O(I^2)$ 必要となる[2].

SVMのスケラビリティ問題に対し、最も良く使用される手法はdecompositionであり、LibSVM [4],  $SVM^{light}$  [9], CoreSVM [5, 6], HeroSVM [14], SimpleSVM [12]などの実装に使用されている。decompositionの基本的な考え方は、訓練データの集合をアクティブ・ワーキングセット(WS)と非アクティブWSの2つに分け、各訓練用サンプルに対する重み係数の更新をアクティブWSに限定し、非アクティブWSの重み係数は一時的に固定したままにする。decompositionの極端な例が、Sequential Minimal Optimization (SMO) と呼ばれる手法で、選び出した2個の訓練用サンプルだけでアクティブWSを構成し、2個の重み係数だけを最適解に近づくように更新する。訓練サンプルの選出と重み係数の更新処理を繰り返しながら最適解に近づいていくSMOでは、収束までの繰返し回数を少なくするため、全訓練用サンプルをスキャンして適切なサンプルペアを見つけていく。このためサンプル数が多い場合、SMOでも学習に多大な時間を要するという問題がある。

そこで本稿では、iSVM と呼ぶ、大規模なSVM学習に適した、2段階の逐次ワーキングセット選択手法を提案する。本手法では、まず各クラスから1個ずつ選んだサンプルからなる最小のWSを構成し、SVM学習の初期解を得る。その後、WSに訓練サンプルを1個ずつ追加しながらSVMの解を更新していく。WSに追加するサンプルの選び方が、学習過程の前半と後半で異なっている。前半ではランダムサンプリングで得た一定個数のサンプル集合の中から最も効果がありそうな1個をWS追加用に選び出す。後半では追加対象として適切かどうかのチェックを残った訓練サンプルの全てに対して行う。

また本稿では、iSVMの学習時間は訓練用サンプルの数に比例、もしくは、サポートベクトル数の3乗に比例することを示す。さらに、大規模なベンチマークデータを使って、訓練サンプルの数に比べサポートベクトルの数が少ない判別問題で特にiSVMが高速に動作することを示す。

以下、2章ではSVM学習過程のQP問題への定式化とQP問題を解くためのdecomposition手法について簡単に述べる。3章ではiSVMを詳しく手法を説明し、その処理量について考察する。本研究に関連する研究を4章で紹介し、他手法との比較実験の結果を5章に示す。

著者らによる先行的な検討結果[18]と異なり、本報告では、訓練サンプルの数に比べサポートベクトルの数が少ない判別問題で特にiSVMが高速に動作する理由についての考察と検証を行っている。

## 2. SVM 学習および decomposition のアルゴリズム

### 2.1 SVM の学習

SVMの学習[1, 2]では、ラベル  $y_i \in \{-1, +1\}$  が付与された  $I$  個の訓練サンプル  $x_i \in R^d$  が与えられる。この時、SVMの学習は以下の最適化問題を解くことに同じである。

$$\min_{\alpha} L(\alpha) = \frac{1}{2} \sum_{i,j=1}^l y_i y_j \alpha_i \alpha_j K(x_i, x_j) - \sum_{i=1}^l \alpha_i \quad (1)$$

$$\text{subject to } \sum_{i=1}^l y_i \alpha_i = 0, \\ 0 \leq \alpha_i \leq C, i=1, \dots, l$$

ただし、 $K(x_i, x_j)$  はカーネル関数、 $C$  は訓練データのノイズサンプルに対するペナルティの最大値であり、 $\{\alpha_i \mid i=1, \dots, l\}$  は以下のように判別関数に使用される。

$$y = \text{sgn} \left( f(x) = \sum_{\alpha_i \neq 0} y_i \alpha_i K(x_i, x) + b \right) \quad (2)$$

最適化問題(1)は $l$ 個の変数 $\alpha_i$ と $l^2$ 個のパラメータ $K_{ij} = K(x_i, x_j)$ ,  $i, j=1, \dots, l$ を含んでいる。この $l^2$ 個のパラメータ $K_{ij}$ はサンプル数が10万個を超えると、通常の計算機のメモリには入りきらない。

## 2.2 Sequential Minimal Optimization (SMO) と decomposition

文献[7, 8, 9]にあるよう、decompositionの基本的な考え方は、サイズ $l$ のQP問題をより小さな複数のQP問題に分解し、その部分問題ごとに最適化を繰り返すことにある。この繰り返し毎にdecompositionでは、 $l$ 個の訓練サンプルが2つのカテゴリに分けられる。アクティブ集合 $W$ に属するサンプルに対応する係数 $\alpha_i$ は更新されるが、非アクティブ側の係数は更新対象とならない。目的関数 $L$ を最小化するようにアクティブ側の係数を何らかの方法によって更新する。そして、再び適切な $W$ を求めなおす。終了条件が満たされるまで、この繰り返し処理が行われる。

Sequential Minimal Optimization (SMO)はdecompositionの極端な例であり、2個の訓練サンプルだけでアクティブ集合 $W$ を構成する。部分問題を解くためには $2l$ 個のパラメータ $K_{ii} \dots K_{ij}, K_{jj} \dots K_{ji}$ が一時的に必要なことになるだけなので使用メモリの点で有利である。しかも解析的な手法で $\alpha_i$ の更新後の値が決定できるため、部分問題の最適化が高速に行える。

$x_i$ と $x_j$ が選ばれた訓練サンプルの時、(1)の目的関数 $L$ を最適にするための新たな $\alpha_i$ と $\alpha_j$ の値は、制約 $(0 \leq \alpha_i \leq C, i=1, \dots, l)$ を無視すれば次式のようなになる。

$$\alpha_i^{new} = \alpha_i^{old} + \frac{(E_j - E_i)}{\kappa} \quad (3)$$

$$\alpha_j^{new} = y_j (const - y_i \alpha_i^{new})$$

ただし、 $const = y_i \alpha_i^{old} + y_j \alpha_j^{old}$ ,  $\kappa = K_{ii} + K_{jj} - 2K_{ij}$ ,  $E_k = f(x_k) - y_k$ である。この更新手順によって、目的関数 $L$ の値は以下の値の分だけ改善される。

$$\Delta L_{ij} = - \frac{(E_i - E_j)^2}{2\kappa} \quad (4)$$

この改善量を目安にして、適切な $(x_i, x_j)$ を選ぶ種々のヒューリスティクスがSVMの実装として提案されている[10, 4]。表1にSMOの実装の枠組みを示す。

## 3 2段式逐次ワーキングセット選択手法 (iSVM) のアルゴリズム

表1の処理で最も処理量が大きいのはステップ5であり、全訓練サンプルに対して、 $E_k$ の値を更新しなければいけない。サポートベクトル(対応する係数 $\alpha_i$ の値が0より大きい訓練サン

プル $x_i$ )だけが最終的に式(2)に示した判別関数に関するの

表1 SMOのアルゴリズム  
Table1 Algorithm of SMO

- 
- Input:* Training data  $T = \{x_k \mid k=1, \dots, l\}$
0. Initialize a feasible solution
  1. Set iteration  $t = 0$
  2. **While** *StoppingCondition* is not satisfied
  3.     Select a pair of vectors  $(x_i, x_j)$
  4.     Update  $(\alpha_i, \alpha_j)$  analytically
  5.     Update violation states  $E_k, k=1, \dots, l$
  6.     Set  $t = t + 1$
  7. **Endwhile**

*Output:* Coefficients  $\alpha = \{\alpha_k \mid k=1, \dots, l\}$

---

表2 iSVMのアルゴリズム  
Table2 Algorithm of proposed iSVM

- 
- Input:* Training data  $T = \{x_k \mid k=1, \dots, l\}$
0. Select the first working set  $W_0$ . Find the first temporal solution  $S_0$ . Set  $T = T - W_0$
  1. Set iteration  $t = 0$
  2. **While** *StoppingCondition* is not satisfied
  3.     Select one vector  $x_t$  in  $T$
  4.     Update  $W_{t+1} = W_t \cup \{x_t\}, T = T - \{x_t\}$
  5.     Find new solution  $S_{t+1}$  on  $W_{t+1}$
  6.     Set  $t = t + 1$
  7. **Endwhile**

*Output:* Coefficients  $\alpha = \{\alpha_k \mid k=1, \dots, l / W_t\}$

---

に、SMOのステップ5は全ての訓練サンプルを対象にするので計算効率はよくない。このため、非サポートベクトルを訓練集合から取り除く shrinking と呼ばれる手法[9]も提案されている。しかしながら、訓練サンプルがサポートベクトルであるかどうかは、計算を始める段階では不明である。そこで、サポートベクトルを逐次発見していく手法を提案する。逐次発見手法の基本的な考え方は、

- (1) 計算途中で一旦求めたSVMを利用して適切なサポートベクトル候補を見つけること
- (2) 訓練集合全体でなく、選ばれた訓練サンプルに限定して部分問題を最適化すること

である。以下、表2に示した、iSVMと呼ぶ提案手法のアルゴリズムについて述べる。

### 3.1 初期化

iSVMでも、最初のワーキングセット $W_0$ を構築するために2

個の訓練サンプルが選ばれる。2クラスの判別問題の場合、単に各クラスから1個ずつ訓練サンプルを適当に選び出している。この初期選択のやり方は既存の類似研究に比べると単純である。文献[11]の著者らは学習用パラメータ $p$ が与えられた下で、 $p$ 個の訓練サンプルをランダムに選び出している。また、文献[4]では、式(4)などを使って、最も距離の近い2個の訓練サンプルが選び出される。また、coreSVM [5, 6] やSimpleSVM [12]では異なった実装が行なわれている。しかし、我々の予備実験によると、iSVMでは初期選択の違いが結果にあまり影響されないことがわかったので、初期選択は簡単化した。

### 3.2 2 段式のワーキングセット更新

SVM の学習においては、どの訓練サンプルが SV になるのか事前に分かっているならば、非SV を訓練サンプルから取り除くことで、精度を落とすことなく学習を効率化できる。このため、余分な訓練サンプルを取り除くための戦略が重要になってくる。我々は、WS更新に用いる訓練サンプルの選択手法を学習処理の前半(3.2.1節)と後半(3.2.2節)で切替える戦略を採用した。

#### 3.2.1 再サンプリングによる選択

iSVMのステップ3では、小規模な訓練サンプルの集合から最も有望そうなサンプルを選び出す。選択されるのは、前回の繰返し処理で得たSVMの解に照らし合わせて、最も推定精度が良くない訓練サンプルである。具体的には、 $t$  回目の繰返し処理で以下の  $x_t$  が選ばれる。

$$x_t = \arg \max_{x_k} f_t(x_k) - y_k \quad (5)$$

ただし、

$$f_t(x) = \sum_{x_i \in W_t} y_i \alpha_i K(x_i, x) + b_t \quad (6)$$

は、 $t$  回目の一時的な解  $S_t$  である。

選択基準の式 (5) は SMO[10]等の実装に使用されたものと同じである。しかし、iSVM がサンプリングによって構築した一定数個の訓練サンプル集合の中から最も有望な更新用のサンプルを選び出すのに対し、従来の SMO 実装では全訓練サンプルの中から選び出している。この手法の違いによる処理量の違いについては3.4節で議論する。

このサンプリングは繰返し処理のたびに行われる。サンプリングを毎回行う理由は2つある。1つは、サンプリングで構築したサンプル集合の中で最も有望な訓練サンプルであっても必ずしも全訓練サンプルの中では最適ではないため、サンプリングを繰り返してリスクを減らすこと。2つ目は、アクティブ WS が小さい段階の SVM の一時解はあまり精度が良くないためである。学習の早い段階で訓練サンプルを除外していくと、本当の SV を取り除いてしまう可能性がある。

一時解  $S_t$  がある程度良くなると、この段階の訓練サンプル選択手法をやめて、次の段階の選択手法に切替えて良い。具体的には、著者らは以下のいずれかの条件が成り立つと第二段の選択手法に切替えている。

- 1) ランダムに選択したN個の訓練サンプルで構築した訓練集合のどのサンプルでも最適化関数の値を向上できない
- 2)  $W_t$  中のサンプルの数が事前設定した数に達した(本稿の実験では1,000)
- 3)  $W_t$  の全データが既に1回以上アクティブ WS に入るかど

うかのチェックが行われた

最初の条件は、kernel matrix approximation と呼ばれる手法 [13] や CoreSVM [5, 6] 等で使用されているものと同じである。これは、59 個のサンプルがあれば、アクティブ WS に追加すべき上位 5%のサンプルの内の 1 個は含まれるといったヒューリスティックス[13] に基づいた条件である。2 番目と 3 番目の条件は、一時解が安定していて十分に最適解に近いと判断するためのものである。

#### 3.2.2 全数検査による選択

第一段階終了時にSVMの準最適解が求まっているという仮定に基づき、第二段階ではまだ残っている訓練サンプルを全て一つずつ検査していく。訓練サンプル  $x_t$  を検査して、同サンプルが一時解  $S_t$  で目的関数をあまり向上できないならば訓練対象から取り除き、向上できるならば  $W_t$  に追加してSVMの一時解を更新する。検査対象のサンプルが無くなったときに第二段階は終了する

### 3.3 アクティブ WS における最適化

従来のSMO同様、iSVMは、アクティブ・ワーキングセット  $W_t$  のサンプルだけで目標関数  $L$  を最小化するので、効率的な学習が可能である。さらに、 $W_{t-1}$  から求めた一時解  $S_{t-1}$  を使って  $W_t$  に対する最適化を行うので、最後に追加されたサンプルによる影響だけを考慮して最適化を行うことができ、表2のステップ5の収束が早いといった特徴をiSVMは持っている。

### 3.4 計算量に対する考察

ここでは iSVM における計算量の問題を考察する。

$t$  回目の繰返しにおいて、ステップ 3 では、1 個の訓練サンプルを検査するのに  $O(|W_t|)$  の処理時間がかかる。選択手法の第一段階では、条件 3) で終了する最悪の場合、全訓練サンプルが一度は検査される。そして第二段階では、残っていた訓練サンプルが全て 1 回は検査される。よって、ステップ 3 では  $O(|W_t|I)$  の時間がかかる。

一般的には、サイズ  $|W_t|$  のQP問題を解くためには  $O(|W_t|^3)$  の時間がかかる。しかし、提案の iSVM の場合、 $t-1$  回目の繰返しで得た一時解  $S_{t-1}$  が利用できるため、ステップ 5 の 1 回分の実行は容易に  $O(|W_t|^2)$  で可能である。実際は更に効率よい更新手法[12, 17]を用いれば、1回あたり  $O(|W_t|)$  で処理できる。結局、学習の全工程において最適解の計算には  $O(|W_t|^3)$  の時間がかかる。

iSVM の時間複雑性は  $O(|W_t|^3 + |W_t|I)$  となる。選択手法が第二段階に至ると shrinking 手法が使えるので、最終的なアクティブ・ワーキングセット  $W_t$  のおおよその大きさは、最終的なサポートベクトルの数に等しくなる。よって、訓練サンプルの数に線形、もしくはサポートベクトル数の3乗に比例となる。

表1のSMOの時間複雑性はSMOの繰返し回数を  $\tau$  とすると  $O(\tau I)$  である。しかし、実際、 $\tau$  の値は  $I$  に依存しているため、SMOの処理時間は  $I$  の線形より通常大きくなる。一方、最終的なサポートベクトルの数が訓練サンプルの数  $I$  に比べて十分に小さければ、iSVMの処理時間は訓練サンプルの数に線形になる。このため、サポートベクトルの数が訓練サンプルの数に比べて少ない場合、通常のSMOの実装より iSVMの方が有利なる。

## 4. 関連研究

既に述べたように iSVM は decomposition 手法の一種である。ここでは従来手法と iSVM の違いについてまとめる。

3.1 節で述べたように iSVM は従来手法に比べて単純な方法でアクティブ WS の初期値を求めている。我々の予備実験によると、iSVM では初期選択の違いが結果にほとんど影響されないことがわかっており、簡単な処理で WS の初期値が求まる利点がある。

WS の更新に関しては、SimpleSVM [12] や CoreSVM [5, 6] などのいくつかの実装で、iSVM 同様、繰り返しの度に毎回 1 個ずつの訓練サンプルを追加すると同じ逐次更新のアルゴリズムが利用されている。追加するサンプルを選択するときの基準に違いがあって、目的関数への寄与度合い [12], [5, 6], [4] や確率的な推定 [11] で工夫がなされている。一方、iSVM は他と違って、学習過程の前半と後半で異なる選択手法を切替える戦略をとっている。iSVM の場合、前半では繰り返しの度に、ランダムサンプリングをやり直してサンプル集合  $T$  を構成しなおすが、文献 [5, 6], [12] では訓練サンプルの再検査を行うかどうかが明確になっていない。再検査を行わないと、早い段階で本来サポートベクトルとなるべきサンプルが精度の良くない一時解のせいで取り除かれる危険性がある。

また iSVM は、他手法に比べて、より精妙な終了条件を持っている。3.2.1 節に示した終了条件の 1) だけを実装する CoreSVM は終了判定が甘く設定してあり、次章の実験結果が示すように学習結果が安定していないといった問題が生じる。

## 5. 評価実験

学習と評価には、表 3 の異なる 4 つの領域のデータセットを使用した。同表の "Web" は UCI Machine Learning Repository にある web ページ分類のデータであり、"IJCNN" は IJCNN2001 で使われたテキスト復号のデータ、"USPS zero-one" は USPS の手書き文字判別データの「0」と「1」を識別するデータ、"KDD-CUP 1999" は KDD1999 の会議で使用されたネットワーク侵入検知のデータである。実験に使用したのは、3GHz の CPU と 2GB のメモリを持った Windows PC である。

### 5.1 iSVM の高速性の検証

ここでは、SMO の優れた実装である libSVM [4], KDDCUP 1999 のデータセットで注目すべき性能を最近出した CoreSVM と iSVM とで処理性能を比較する。測定したのは、学習時間、サポートベクトルの数、分類精度である。学習用のパラメータは、分類精度がよくなる様に以下のように設定した。ガウスカーネル  $K(x, y) = \exp(-\gamma \|x-y\|^2)$  に対し、Web の場合  $\gamma = 0.1$ ,  $C = 10$ , IJCNN の場合  $\gamma = 1$ ,  $C = 10$ , USPS zero-one の場合  $\gamma = 0.003$ ,  $C = 10$ , KDD-CUP の場合  $\gamma = 0.1$ ,  $C = 1,000,000$  とした。

iSVM と CoreSVM は確率的な処理を含むので、学習結果は本来毎回同じではない。そこで、学習と評価を 10 回繰返した。なお、CoreSVM の実装 (バイナリ) は毎回同じ学習結果を出すように設定してあるので、訓練サンプルの並び順を換えて異なる実行結果を出すようにした。また、iSVM は第二段のサンプル選択で shrinking 手法を適用する事でさらに高速化することは可能であるが、今回の実験では 2 段式のワーキングセット選択手法に主眼があるため shrinking は実装していない。

表 4 に測定結果を示す。3.4 節で考察したように、SV 数が訓

練サンプルの数に比べ少ない場合は、iSVM の学習時間は線形なので SMO より有利である。実際、SV の数が訓練サンプルの数に比べ少ないデータセット USPS と KDD-CUP で、iSVM を libSVM と比較すると、shrinking を実装していないに関わらず iSVM は同

表 3 実験に使用したデータセット

Table 3 Datasets used in experiments

データセット	次元数	訓練サンプルの数	評価用サンプルの数
Web	300	49,749	14,951
IJCNN	22	49,990	91,701
USPS zero-one	676	266,079	75,383
KDD-CUP 1999	127	4,989,431	311,029

等以上の学習精度を達しながら高速な学習を行なっている。このことから提案の 2 段式の逐次ワーキングセット手法が有効であったと言える。CoreSVM と比較すると、学習時間は Web ならびに IJCNN で iSVM が勝っているが、KDD-CUP では CoreSVM が 2 秒といった極端に短い学習時間を達成している。しかし、KDD-CUP において、CoreSVM の学習精度は安定しておらず、iSVM や libSVM に比べると精度の平均値は低い。CoreSVM は KDD-CUP のデータセットの学習に特化しており、種々の問題を解くには学習の終了条件が適切に設定できていないと言える。

### 5.2 学習終了条件による学習時間の分析

次に、KDD-CUP のデータセットを使って、iSVM の学習終了条件を変更して、学習時間を短縮した場合の学習精度を測定した。第一段階の処理だけで第二段階を行わない場合、第二段階の処理で検査対象になる訓練サンプルの内、10%の訓練サンプルしか検査しない場合、20%の場合、40%の場合、80%の場合を測定した。

5.1 節の結果を含め、測定結果を図 1 に示す。同図の一番左側に示した iSVM の第一段階だけで学習を終了した場合の学習時間は極端に短いですが、学習精度にばらつきがある。また、検査対象の割合を減らすことで学習時間が大幅に短縮できることがわかる。ただし、第二段階の検査対象を減らすことで学習精度のばらつきは大きくなることも同図から確認できる。

第一段階の処理だけで学習を終了した iSVM について、より詳細に解析すると、ワーキングセット更新の平均回数は 32 であり、最大でも 1,900 個 (全訓練サンプルの 0.04% に相当) のサンプルを検査しただけで第一段階の処理が終了していた。学習を第一段の処理だけで第二段の処理を行わない場合の学習時間は 2.8 秒であった。これは CoreSVM の学習時間と殆ど同じである。しかし、学習結果を安定させる観点からは、iSVM で第二段の処理を省略する学習条件は好ましくはない。

大規模データセットからの学習問題として著名な KDD-CUP データセットに対しては、種々の学習方式で挑戦が行われている。表 5 にデータランダムサンプリング、active learning [15]、クラスタベースの手法 [16]、CoreSVM [5, 6]、LibSVM [4] と iSVM の学習時間と評価用データでの判別エラーの数を示す。CoreSVM、LibSVM と iSVM については著者らが行った実験の結果であり、その他の手法については、文献 [5] から引用した。iSVM がエラー量の点で最善であり、学習精度が安定していない CoreSVM を除けば他手法の中で最速のクラスタベースの手法よりも 1.6 倍高速である。本提案の iSVM は大規模データの学習に適していることが KDD-CUP データセットで確認できた。

## 6. おわりに

大規模データに対する学習に適した, iSVM と呼ぶ新たな手法

を提案した. iSVM は decomposition 手法の枠組みに属するが,

表4 学習性能の測定結果 (iSVM と CoreSVM については測定 10 回繰り返している)

Dataset	iSVM				LibSVM			CoreSVM		
	#SV	#SV (%)	Accuracy (%)	Time (s)	#SV	Accuracy (%)	Time (s)	#SV	Accuracy (%)	Time (s)
Web	3,289	6.61	99.26±0.02	101	4,660	99.32	214	8,828	99.36±0.00	1140
IJCNN	2,810	5.62	98.91±0.04	170	3,154	98.99	55	5,326	99.07±0.01	495
zero-one	1,200	0.45	99.54±0.00	344	1,598	99.54	1432	1,639	99.52±0.02	343
KDD-CUP99	735	0.01	94.07±0.90	2919	1,132	92.48	25,200	43	90.66±4.16	2

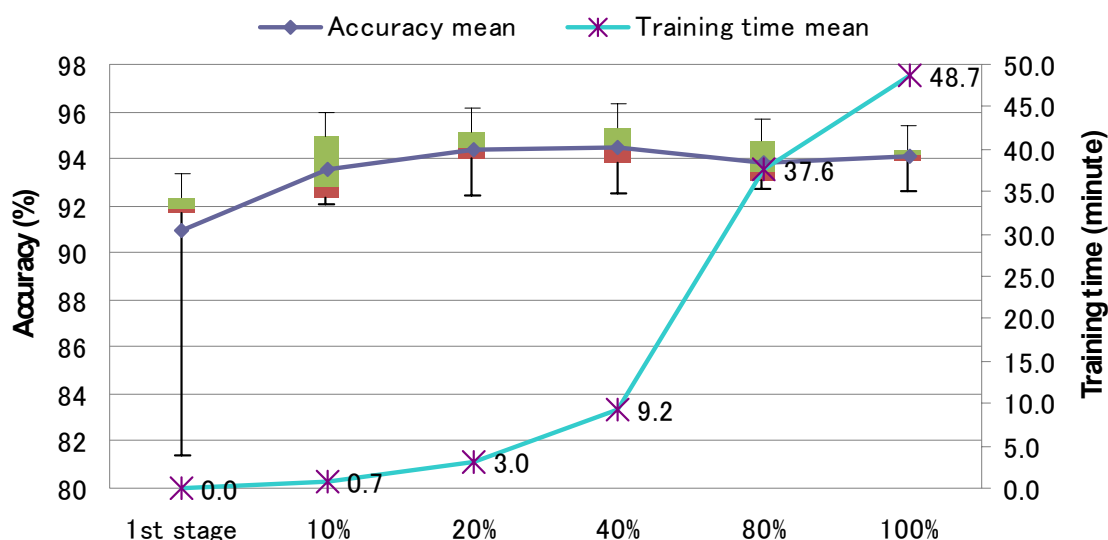


図1 学習終了条件を変更した時の学習時間と平均分類精度の関係

Figure 1 Training time mean and Accuracy mean for variars stopping conditions

表5 KDD-CUP データセットに対する各種手法の学習性能  
Table 5 Performance of iSVM and other methods for KDD-CUP

Method	Training time (s)	Testing errors
Random sampling (5%)	16,351	25,587
Active learning	94,192	21,634
Cluster-based	4,752	20,938
LibSVM	25,200	23,389
CoreSVM	2	19,513
iSVM	2,919	18,444

2 段式のアクティブワーキングセット更新方法を採用していることに特徴がある. 第一段目ではできるだけ少数の訓練サンプルで SVM の準最適解を求めることを目的としている. 第二段目では検査されていない残りの全訓練サンプルを使って, 学習結

果の安定と精度向上を図っている. アルゴリズムの解析と実証実験の結果, 提案の iSVM は訓練サンプルの数に比べサポートベクトルの数が少ない判別問題で特に有効であることが分かった.

訓練用サンプルの数が約 500 万件である KDD-CUP 1999 の ネットワーク侵入検知のデータセットを学習する場合, SM0 を実装した LibSVM[4] では分類精度が 92.5% で処理時間が 25,200 秒かかるのに対し, iSVM の精度は 94.1% で上回っており, しかも計算時間が 2,919 秒と 8 倍以上の高速性を達成している.

### [文献]

- [1] C. Cortes and V. Vapnik, "Support vector networks," Machine Learning, vol. 20, pp. 273-297, 1995
- [2] C. Cristianini and J. Shawe-Taylor, "An Introduction to Support Vector Machines", Cambridge University Press, 2000
- [3] C. J. C. Burges, "A tutorial on support vector machines

- for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121-167, 1998.
- [4] R. Fan, P. Chen, and C. Lin, "Working Set Selection Using Second Order Information for Training Support Vector Machines". *J. Mach. Learn. Res.* 6, 1889-1918, 2005.
- [5] I. W. Tsang, J. T. Kwok, and P.-M. Cheung, "Core vector machines: Fast SVM training on very large datasets," in *J. Mach. Learn. Res.*, vol. 6, pp. 363-392, 2005.
- [6] I. W. Tsang, A. Kocsor, and J. T. Kwok. "Simpler core vector machines with enclosing balls" *Proceedings of the Twenty-Fourth International Conference on Machine Learning (ICML)*, pp.911-918, Corvallis, Oregon, USA, June 2007.
- [7] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: an application to face detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1997.
- [8] E. Osuna, R. Freund, and F. Girosi, "An improved training algorithm for support vector machines," in *Neural Networks for Signal Processing VII - Proceedings of the 1997 IEEE Workshop*, N. M. J. Principe, L. Gile and E. Wilson, Eds., New York, pp. 276-285, 1997.
- [9] T. Joachims, "Making large-scale support vector machine learning practical," in *Advances in Kernel Methods: Support Vector Machines*, A. S. B. Scholkopf, C. Burges, Ed., MIT Press, Cambridge, MA, 1998.
- [10] J. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods - Support Vector Learning*, B. Scholkopf, C. J. C. Burges, and A. J. Smola, Eds., Cambridge, MA: MIT Press, 1999.
- [11] P. Mitra, C.A. Murthy, S. K. Pal, "Data Condensation in Large Databases by Incremental Learning with Support Vector Machines," *icpr*, p. 2708, 15th International Conference on Pattern Recognition (ICPR'00) - Volume 2, 2000.
- [12] S. Vishwanathan, A. Smola, and M. Murty, "SimpleSVM", *Proceedings of the Twentieth International Conference on Machine Learning* (pp. 760-767), 2003.
- [13] A. Smola and B. Scholkopf, "Sparse greedy matrix approximation for machine learning", in *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*, pages 911-918, Stanford, CA, USA, June 2000.
- [14] J. X. Dong, A. Krzyzak, and C. Y. Suen, "Fast SVM Training Algorithm with Decomposition on Very Large Datasets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 4, pp. 603-618, April 2005.
- [15] H. Yu, J. Yang, and J. Han. "Classifying large data sets using SVM with hierarchical clusters". In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 306-315, Washington DC, USA, 2003.
- [16] D. Boley and D. Cao. "Training support vector machine using adaptive clustering". In *Proceedings of the SIAM International Conference on Data Mining*, pages 126-137, Lake Buena Vista, FL, USA, April 2004.
- [17] G. Cauwenberghs and T. Poggio, "Incremental and decremental support vector machine learning", In T. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, Cambridge, MA, 2001. MIT Press.
- [18] D. D. Nguyen, K. Matsumoto, K. Hashimoto, Y. Takishima, and M. Terabe, "Two-stage Incremental Working Set Selection for Fast Support Vector Training on Large Datasets," In *Proceeding of 2008 IEEE International Conference on Research , Innovation and Vision for the Future (RIVF)* (will be appeared in July 2008)

---

### ズン デュック グエン Dung Duc NGUYEN

1993 Hanoi University of Pedagogy 卒。 2003 北陸先端大大学院修士課程了。 2006 同大学院博士課程了。 1993年から2007年までの Institute of Information and Technology (Hanoi) での勤務を経て KDDI 研究所に入社, 知能メディアグループに所属。 機械学習, データマイニング, パターン認識, ベトナム語の文字認識に興味を持つ。 日本データベース学会会員。

### 松本 一則 Kazunori MATSUMOTO

1984 京大・工・情報工学卒。 1986 同大大学院修士課程了。 同年国際電信電話(株)入社, 研究所所属。 現在, KDDI 研究所知能メディアグループにて, マルチメディア検索, コンテンツ配信の研究開発に従事。 電子情報通信学会会員。

### 橋本 和夫 Kazuo HASHIMOTO

1977 東北大・工・電子工学卒。 1979 同大大学院修士課程了。 同年国際電信電話(株)入社, 同研究所で人工知能の応用研究に従事。 2006 東北大学大学院情報科学研究科教授、Web コミュニケーションの研究に従事。 電子情報通信学会、情報処理学会、人工知能学会、AAAI、IEEE 会員。 情報科学博士。

### 滝嶋 康弘 Yasuhiro TAKISHIMA

1986年東大・工・電気卒。 1988 同大大学院電子工学修士課程了。 国際電信電話(株)(現 KDDI(株)) 入社。 現在, (株)KDDI 研究所知能メディアグループリーダー。 この間, 動画像の符号化方式, 動画通信システム, 情報理論の研究・開発に従事。 電子情報通信学会, 映像情報メディア学会, 画像電子学会会員。 工博。

### 寺邊 正大 Masahiro TERABE

1993 京大・工・精密工学卒。 1995 同大大学院修士課程了。 同年 (株) 三菱総合研究所入社。 現在, 主任研究員。 2006年より東北大院 (情科) 助教授 (現在, 准教授) を併任。 工博。 主にデータマイニングの応用研究等に従事。 JSAI, SICE, AAAI, ACM の各会員