Vol.2009-NL-192 No.12
2009/7/23

サポートクラスによる Passive-Aggressive アルゴリズムの多クラス化

松島 慎[†1] 清水伸幸[†1] 吉田和弘[†1]
二宮 崇[†1] 中川裕志[†1]

逐次学習は、訓練データを受け取る毎に簡単なパラメータ更新を行うだけでよいため、計算時間やメモリの効率が良い点、理論的な性能の保障など、多くの面から近年注目されている。本研究は、逐次学習の中でも有力な枠組の一つである Passive-Aggressive アルゴリズム (PA) を多クラス分類に拡張した Support Class Passive-Aggressive アルゴリズム (SPA) を提案する。PA は、損失関数を与えれば、一貫した定式化で多クラス分類のパラメータ更新式を導出することができるが、既存の多クラス PA で用いられている更新式はパラメータ更新後も受け取った訓練データを誤分類する場合がある。我々が提案する SPA は、データを受け取る毎にサポートクラスと呼ぶクラスの集合を定め、このサポートクラス全体に関してパラメータ更新を行う。SPA による更新後の分類器は受け取った訓練データを必ず正しく分類する。文書分類問題および画像認識問題に対する評価実験を行い、SPA が既存の逐次学習の手法に比べ良い精度を達成することを確認した。

Multi-Class Passive-Aggressive Algorithm with Support Classes

SHIN MATSUSHIMA ,[†1] NOBUYUKI SHIMIZU,[†1]
KAZUHIRO YOSHIDA,[†1] TAKASHI NINOMIYA[†1]
and HIROSHI NAKAGAWA[†1]

Online learning is recently getting attention in the field of NLP due to its simple updates, efficiency in terms of computing time and memory space, and theoretical guarantees for its performance. We propose Support Class Passive-Aggressive Algorithm (SPA), an online learning classifier for multiple classes, derived from the Passive Aggressive Algorithm (PA). While the PA framework allows integrations with different loss functions, so far, it has not been combined with a multiclass loss function that exactly classifies the current training instance into the correct class. Our SPA determines a set of *support classes*, and the parameters are updated for the support classes. SPA always classifies the current training instance into the correct class. Experiments show our method improves the traditional PA algorithms.

1. はじめに

日々増加する Web 文書などに見られるように、与えられた一定の訓練データから学習を行うのではなく、訓練データを逐次的に受け取ることができる状況下での学習が必要となる場合があり、これを逐次学習という。逐次学習は、時間がたつにつれて性質が変わっていくデータに対しても適用可能であるだけでなく、訓練データを受け取る毎に簡単なパラメータ更新を行うだけでよい場合が多いため計算時間やメモリーの効率が良い点、また損失関数や誤分類するデータの総数に対し理論的な上限の保障ができる点など、多くの面から近年注目されている学習方法である。

Passive-Aggressive アルゴリズム[1] は、そのような逐次学習アルゴリズムの中でも有力な枠組みの一つであり、損失関数を与えれば、一貫した定式化で多クラス分類やランキングの問題についてもパラメータ更新式を導出することができる。実際に多クラス分類問題のための Passive-Aggressive アルゴリズムも提案されており、複数の 2 値分類器を組み合わせて用いる手法などと比べて、より合理的な学習手法として認められている。しかしながら、既存の多クラス Passive-Aggressive アルゴリズムで用いられている更新式は、受け取った訓練データにおける正解クラスと、現在の分類器が最大のスコアを与えたクラス、すなわち高々 2 つのクラスに対してしかパラメータ更新を行わない。これは、パーセプトロン[2] に対する既存の多クラス拡張などと同様、2 値分類のための逐次学習アルゴリズムにおける、2 クラスに対するパラメータ更新をそのまま多クラス分類問題に持ち込んだものと考えられ、クラス間の関連性など、多クラス分類問題の構造を無視しているといえる。特に、既存の多クラス Passive-Aggressive アルゴリズムでは、パラメータ更新後も、受け取った訓練データを誤分類する場合がある。これはパラメータ更新が 2 クラスしか扱わないのに対して、誤分類は複数のクラスにわたって起こりうるからで、2 値分類問題では発生しなかった問題である。

本稿で我々が提案する Support Class Passive-Aggressive アルゴリズムでは、データを受け

†1 東京大学
The University of Tokyo

取る毎に、サポートクラスと呼ぶ適当なクラスの集合を定め、このサポートクラス全体に関してパラメータ更新を行う。サポートクラスを適切に定めてやることによって、更新後の分類器は、訓練データを正しく分類できるようになる。このアルゴリズムは、受け取った訓練データを一定のマージンで正しく分類することを強制するような損失関数を設計し、Passive-Aggressive アルゴリズムの枠組みで用いることによって自然に得られる。

本稿では次節でまず逐次学習の問題設定、及び PA アルゴリズムの定式化と、我々がその枠組みの中でどのような改良を行ったかを述べる。続く第 3 節では具体的な問題を再度定式化しこれを解析的に解くための詳細を述べ、4 節では評価実験の結果を示しその評価を行った。

2. Passive-Aggressive　アルゴリズム

2.1 逐次学習と Passive-Aggressive　アルゴリズム

本節ではまず K クラス分類の逐次学習の問題設定について説明する。$\mathbf{x} \in X \subset \mathbb{R}^D, y \in Y = \{1,2,\cdots,K\}$ とする。さらにここでは線形学習器のみを考えて、学習アルゴリズムの取りうる仮説 $h : X \to Y$ は、K 本の重みベクトル $\mathbf{w}_u \in \mathbb{R}^D (u = 1,2,\cdots K)$ でパラメータ付けされているとし、$\mathbf{w} = (\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_K)$ とすると、

$$h_{\mathbf{w}}(\mathbf{x}) = \arg\max_{u=1,\cdots,K} (\mathbf{w}_u \cdot \mathbf{x}) \tag{1}$$

であるとする。すなわち、$\mathbf{x}$ を与えられたとき、仮説 $h_{\mathbf{w}}$ は全てのクラスの中で、$\mathbf{w}_y \cdot \mathbf{x}$ が最大となる y を選ぶ。この時に比べる値 $\mathbf{w}_u \cdot \mathbf{x}$ を u の ($\mathbf{x}$ における) スコアという。

逐次学習では毎回単一のデータ $\mathbf{x}$ を受け取り仮説を更新する。データ $\mathbf{x}$ を過去の仮説 $h_{\mathbf{w}}$ によって Y の元へ写したものが現在の $\mathbf{x}$ に対するラベル付け (予測) である。次に、$\mathbf{x}$ に対応する正解ラベル y を受け取り予測との正否などに従い損失を被る。この損失を表現する関数を $\ell(h_{\mathbf{w}}; \mathbf{x}, y)$ と表現する。ここで $\ell(h_{\mathbf{w}}; \mathbf{x}, y) \geq 0$ である。損失に従いアルゴリズムは仮説を変え、新たな仮説 $h_{\mathbf{w}}$ を得る。この繰り返しで仮説の学習が進む。

本稿では代表的な逐次学習の例である Passive-Aggressive(PA) アルゴリズムを扱う。PA アルゴリズムでは、受け取ったデータを $\mathbf{x}$、正解ラベルを y、更新前の現在の重みベクトルを $\mathbf{w}^{old} = (\mathbf{w}_1^{old}, \mathbf{w}_2^{old}, \cdots, \mathbf{w}_K^{old})$、損失関数を ℓ とすると、新たな仮説の重みベクトル $\mathbf{w}^{new} = (\mathbf{w}_1^{new}, \mathbf{w}_2^{new}, \cdots, \mathbf{w}_K^{new})$ は次式の最適化問題の解として与えられる。

$$\mathbf{w}^{new} = \min_{\mathbf{w}} \frac{1}{2} \sum_{u=1}^{K} \left\| \mathbf{w}_u - \mathbf{w}_u^{old} \right\|^2 \quad s.t. \;\; \ell(h_{\mathbf{w}}(\mathbf{x}, y)) = 0 \tag{2}$$

過去の仮説 $h_{\mathbf{w}^{old}}$ がデータ $\mathbf{x}, y$ を 0 の損失をもって予測できたのならば、最適解は自明に $\mathbf{w}_u^{new} = \mathbf{w}_u^{old}$ となる。一方、$h_{\mathbf{w}^{old}}$ が 0 でない損失を被る場合は、現在のデータに対する損失が 0 になるように重みを更新していく。このように受け取るデータに対して時には **Passive** に重みベクトルの更新を行い、時には **Aggressive** に重みベクトルの更新を行うという意味で、このスキームは Passive-Aggressive アルゴリズムと呼ばれる。

2.2 Passive-Aggressive アルゴリズムにおける多クラス分類

Crammer らによる PA アルゴリズムの K クラス分類[1] は、次の様な損失関数を採用し毎回の反復で最適化問題 (式 (2)) を解く。すなわち、

$$\ell(h_{\mathbf{w}}; (\mathbf{x}, y)) = \left[1 - \left(\mathbf{w}_y \cdot \mathbf{x} - \mathbf{w}_u \cdot \mathbf{x}\right)\right]_+$$

ここで $[\bullet]_+$ は $\max(\bullet, 0)$ で定義される閾値関数である。u は現在の仮説における正解クラスを除いた予測クラス

$$u = \arg\max_{u \neq y} \mathbf{w}_u^{old} \cdot \mathbf{x}$$

である。また、$\left(\mathbf{w}_y \cdot \mathbf{x} - \mathbf{w}_u \cdot \mathbf{x}\right)$ をクラス u への (データ $\mathbf{x}$ における) マージンと呼ぶ。これは正解ラベル y のスコアと u のスコアの差であり、現在のデータが u に比べてどれくらい y に割り当てられやすいかを表している。もし u へのマージンが正ならば、現在の重みベクトルは現在のデータを正しく分類する。加えてこの損失関数においてはより確実に仮説がデータを分類するために、このマージンの値が 1 よりも大きいことを要求する。したがって実際の最適化問題は以下のようになる。

$$\mathbf{w}^{new} = \min_{\mathbf{w}} \frac{1}{2} \sum_{v=1}^{K} \left\| \mathbf{w}_v - \mathbf{w}_v^{old} \right\|^2 \quad s.t. \;\; 1 - \left(\mathbf{w}_y \cdot \mathbf{x} - \mathbf{w}_u \cdot \mathbf{x}\right) \leq 0 \tag{3}$$

この最適化問題は線形制約式を一つもつ凸二次計画問題である。これはラグランジュの未定乗数法を用いて比較的容易に解ける。ラグランジュ乗数 τ を用いて、ラグランジュ関数の (ベクトル $\mathbf{w}_v$ に関する) 停留条件は以下のように書ける。

$$\frac{\partial}{\partial \mathbf{w}_v} \left[\frac{1}{2} \sum_{v=1}^{K} \left\| \mathbf{w}_v - \mathbf{w}_v^{old} \right\|^2 + \tau \left\{ 1 - \left(\mathbf{w}_y \cdot \mathbf{x} - \mathbf{w}_u \cdot \mathbf{x}\right) \right\} \right] = \mathbf{0}$$

これを整理すると直ちに重みベクトル $\mathbf{w}_v$ に関する更新式

$$\begin{aligned} \mathbf{w}_v &= \mathbf{w}_v^{old} && (v \neq u, y) \\ \mathbf{w}_v &= \mathbf{w}_v^{old} + \tau \mathbf{x} && (v = y) \\ \mathbf{w}_v &= \mathbf{w}_v^{old} - \tau \mathbf{x} && (v = u) \end{aligned}$$

を得る。ラグランジュ乗数も簡単な計算で求めることができ、

$$\tau = \left[\frac{1 - \left(\mathbf{w}_y^{old} \cdot \mathbf{x} - \mathbf{w}_u^{old} \cdot \mathbf{x} \right)}{2 \left\| \mathbf{x} \right\|^2} \right]_+$$

となることがわかる。

Crammer らはこれらの基本的な PA algorithm の定式化に加えて、ノイズデータなどに対する極端な重みベクトルの更新を避けるための 2 通りの別の定式化を行っている[1)]。これは SVM の定式化にも見られるソフトマージンの概念とマージンスラック変数の導入を用いて次のような形で行われる。

$$\min_{\mathbf{w}} \frac{1}{2} \sum_{u=1}^{K} \left\| \mathbf{w}_u - \mathbf{w}_u^{old} \right\|^2 + C\xi \quad s.t. \ \ell(h_{\mathbf{w}}; (\mathbf{x}, y)) \leq \xi, 0 \leq \xi \qquad (\mathrm{PA-I})$$

$$\min_{\mathbf{w}} \frac{1}{2} \sum_{u=1}^{K} \left\| \mathbf{w}_u - \mathbf{w}_u^{old} \right\|^2 + C\xi^2 \quad s.t. \ \ell(h_{\mathbf{w}}; (\mathbf{x}, y)) \leq \xi \qquad (\mathrm{PA-II})$$

ここで、C はソフトマージンの効果を調節するための超パラメータである。両者とも、より大きな C に対して最適化問題の解は、基本の PA の最適解に近づく。これらもまたラグランジュの未定乗数法により簡単に

$$\tau = \min \left\{ \frac{C}{2}, \frac{1 - \left(\mathbf{w}_y^{old} \cdot \mathbf{x} - \mathbf{w}_u^{old} \cdot \mathbf{x} \right)}{2 \left\| \mathbf{x} \right\|^2} \right\} \quad (\mathrm{PA-I}) \qquad \tau = \frac{1}{2} \left\{ \frac{1 - \left(\mathbf{w}_y^{old} \cdot \mathbf{x} - \mathbf{w}_u^{old} \cdot \mathbf{x} \right)}{\left\| \mathbf{x} \right\|^2 + \frac{1}{2C}} \right\} \quad (\mathrm{PA-II})$$

がわかる。

2.3 Passive-Aggressive アルゴリズムにおける多クラス分類の改良

前述の方法では、損失関数は反復ごとに固定された一つのクラスに対する制約を課していることを述べた。しかしこの場合、この制約を満たしたとしても、更新後の仮説を確実に正しくデータを分類するとは限らず、制約にかかわらない他のクラスに誤分類される可能性がある。

そこで我々は次の損失関数を定義する。

$$\ell(h_{\mathbf{w}}; (\mathbf{x}, y)) = \max_{u \neq y} \left[1 - \left(\mathbf{w}_y \cdot \mathbf{x} - \mathbf{w}_u \cdot \mathbf{x} \right) \right]_+$$

この損失関数はすべてのクラスに依存しており、正解クラスを除くすべてのクラスへのマージンが、1 よりも大きい場合のみ損失関数の最小値 0 をとるので、PA アルゴリズムの枠組みにおいてこれを採用すれば、新たな重みベクトルは現在のデータを必ず正しく分類できる。この損失関数を用いて最適化問題 (式 (2)) を書き下すと以下のようになる。

$$\min_{\mathbf{w}} \frac{1}{2} \sum_{v=1}^{K} \left\| \mathbf{w}_v - \mathbf{w}_v^{old} \right\|^2 \quad s.t. \ \forall u \neq y, \left(\mathbf{w}_y \cdot \mathbf{x} - \mathbf{w}_u \cdot \mathbf{x} \right) \geq 1$$

これは当初の問題とおなじく凸二次計画問題であるが、制約が $K-1$ 本に増えている。ラグランジュ乗数 τ_u を用いて、ラグランジュ関数とその ($\mathbf{w}_v$ に関する) 停留条件は以下のように書ける。

$$\frac{\partial}{\partial \mathbf{w}_v} \left[\sum_{u=1}^{K} \left\| \mathbf{w}_u - \mathbf{w}_u^{old} \right\|^2 + \sum_{u \neq y} \tau_u \left\{ 1 - \left(\mathbf{w}_y \cdot \mathbf{x} - \mathbf{w}_u \cdot \mathbf{x} \right) \right\} \right] = \mathbf{0}$$

これを計算すると

$$\begin{aligned} \mathbf{w}_v &= \mathbf{w}_v^{old} + \sum_{u \neq y} \tau_u \mathbf{x} && (v = y) \\ \mathbf{w}_v &= \mathbf{w}_v^{old} - \tau_v \mathbf{x} && (v \neq y) \end{aligned} \qquad (4)$$

となり、形式的に更新式を導く。式を見るとわかるように、更新時には潜在的に全てのクラスの重みに対して更新を行う。これは、より厳しい損失関数が課す複数の制約に由来しており、既存の PA アルゴリズムには存在しなかったものである。次節で導くアルゴリズムでわかるように、毎回の反復でどのクラスの重みに対して更新を行うべきかをある判断基準を用いて決定することができる。

3. Support Class Passive Aggressive アルゴリズム

前節において、既存の PA アルゴリズムと SPA アルゴリズムへの拡張の動機と概観を述べた。本節では、実際に前節で導いた最適化問題から厳密に凸計画問題を解くための手法を用いて更新を導出する方針、及び導出されたアルゴリズムの詳細を述べる。SPA アルゴリズムでは毎回の更新で次の最適化問題の解を求める。

$$\mathbf{w}^{new} = \min_{\mathbf{w}} \frac{1}{2} \sum_{v=1}^{K} \left\| \mathbf{w}_v - \mathbf{w}_v^{old} \right\|^2 \quad s.t. \ \forall u \neq y. \left(\mathbf{w}_y \cdot \mathbf{x} - \mathbf{w}_u \cdot \mathbf{x} \right) \geq 1$$

これは制約式が $K-1$ 本ある凸二次計画問題であるが、制約式は全て線形であるためスレーターの条件をみたす。スレーターの条件を満たせば、最適解を見つける為には KKT 条件を課せば十分である。すなわち、次の KKT 条件を満たす KKT ベクトル $\mathbf{w}, \tau_u$ が存在すれば、$\mathbf{w}$ はこの問題の最適解である。

$$\frac{\partial}{\partial \mathbf{w}_v} \left[\frac{1}{2} \sum_{u=1}^{K} \left\| \mathbf{w}_u - \mathbf{w}_u^{old} \right\|^2 + \sum_{u \in Y \setminus y} \tau_u \left\{ 1 - \left(\mathbf{w}_y \cdot \mathbf{x} - \mathbf{w}_u \cdot \mathbf{x} \right) \right\} \right] = \mathbf{0} \qquad (\forall v) \qquad (5)$$

$$\begin{aligned} \tau_u \left\{ 1 - \left(\mathbf{w}_y \cdot \mathbf{x} - \mathbf{w}_u \cdot \mathbf{x} \right) \right\} &= 0 && (\forall u \neq y) \\ \tau_u &\geq 0 && (\forall u \neq y) \\ 1 - \left(\mathbf{w}_y \cdot \mathbf{x} - \mathbf{w}_u \cdot \mathbf{x} \right) &\leq 0 && (\forall u \neq y) \end{aligned} \qquad (6)$$

ここで、τ_u は更新式の点からいえばステップサイズに対応する。前節でも議論したように式 (5) は更新式 (式 (4)) を導き、ステップサイズ τ_u の具体的な値が求まれば最適解である $\mathbf{w}$ を求めることができる。

ここでステップサイズ τ_u について以上の KKT 条件を考察すると、(式 (6)) より、任意のクラスラベル u に対し $\tau_u = 0$ もしくは $1 - (\mathbf{w}_y \cdot \mathbf{x} - \mathbf{w}_u \cdot \mathbf{x}) = 0$ が成り立つ。ここで $\tau_u = 0$ なるクラスラベルに関しては、更新式には寄与しない。他方 $\tau_u \neq 0$ なるクラスラベルは更新式に直接寄与し、さらに以下のマージン条件が成り立つ。

$$\mathbf{w}_y \cdot \mathbf{x} - \mathbf{w}_u \cdot \mathbf{x} = 1 \tag{7}$$

この等式は本節の末尾で説明するアルゴリズムの中心的性質を示している。すなわち、ステップサイズが正である (更新すべき) クラスラベルからは、更新後のデータ $\mathbf{x}$ におけるマージンがちょうど 1 になっているという条件を得る。これは言い換えれば、ステップサイズはすべて損失関数が 0 になるために最小限必要な値を定めていることになる。

この条件が成り立ち、正のステップサイズで更新に寄与するクラスを**サポートクラス**と呼ぶ。この言葉を用いてまとめると、更新式を完全に導出するためには、(a) サポートクラスの決定、(b) サポートクラスにおけるステップサイズの決定、ができればよいことになる。これらは、上の KKT 条件のみから解析的に定めることができる。

(a) サポートクラスの決定　全てのクラス u に対して

$$Q_u = \frac{1 - \left(\mathbf{w}_y^{old} \cdot \mathbf{x} - \mathbf{w}_u^{old} \cdot \mathbf{x}\right)}{\|\mathbf{x}\|^2}$$

を計算することによって、逐次的にそれらのクラスがサポートクラスであるか判定することができる。すなわち、この値 Q_u について以下の補題が成り立つ。

補題 (サポートクラス条件). Q_u を降順に並び変え、順に Q_j とする。k がサポートクラスである必要十分条件は、

$$\sum_{j=1}^{k-1} \frac{Q_j}{k} < Q_k \tag{8}$$

となる。ただしここで $\sum_{j=1}^{0} \bullet = 0$ とする。

この補題を用いてサポートクラスの決定が可能である。すなわち、Q_u を値の降順に並び変え Q_1 から順に補題の式 (式 (8)) が満たすかどうかを判定し、最後にこれを満たす 1 から J までの添え字に対応するクラスがサポートクラスである。

(b) サポートクラスにおけるステップサイズの決定　(a) より求まるサポートクラスの集合を S としたとき、$v \in S$ に対し

$$\tau_v = Q_v - \sum_{u \in S} \frac{Q_u}{|S| + 1} \tag{9}$$

となる。

表 1　SPA　アルゴリズム

procedure *SPA*
  foreach $(\mathbf{x}, y)$ **do**
//Search support class
    Compute $Q_u := \frac{1-(\mathbf{w}_y \cdot \mathbf{x} - \mathbf{w}_u \cdot \mathbf{x})}{\|\mathbf{x}\|^2}$;
    Sort $Q_u \to Q_j$ in descending order.
    $k := 1; J := 0;$
    while $\sum_{j=1}^{k-1} \frac{Q_j}{k} < Q_k$ **do**
      $k := k+1; J := J+1;$
    end while
//Set Stepsize
    $\tau_j := Q_j - \sum_{j=1}^{J} \frac{Q_u}{J+1} \quad (\forall j \leq J)$
//Update
    $\mathbf{w}_v := \mathbf{w}_v + \sum_{u \neq y} \tau_u \mathbf{x} \quad (v = y);$
    $\mathbf{w}_v := \mathbf{w}_v - \tau_v \mathbf{x} \quad (v \neq y);$
  end foreach

ステップサイズの導出及び補題の証明は付録を参照されたい。最終的な更新のためのアルゴリズムを (表 1) に示す。

同様の議論を用いて、提案した損失関数を用いた PA-I および PA-II の枠組みにおける更新式も導出することができる。これらのアルゴリズムを SPA-I および SPA-II と呼ぶことにする。これらの更新も、複数のクラスにわたり、サポートクラスは同様の判定基準を持つ。最適化問題、ステップサイズ、サポートクラス条件は表 2 に示す。

4. 実 験 結 果

SPA, SPA-I 及び SPA-II の性能を Machine Learning Group Datasets[⋆1]の 20 Newsgroups corpus と LIBSVM data[⋆2]の USPS データセット及び Reuters-21578[⋆3] を用いて評価した。20

⋆1 http://mlg.ucd.ie/datasets
⋆2 http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/
⋆3 http://www.daviddlewis.com/resources/testcollections/reuters21578/

表 2 SPA-I および SPA-II アルゴリズムの概要

	SPA-I	SPA-II
Optimization	$\mathbf{w}^{new} = \min_{\mathbf{w}} \frac{1}{2}\sum_{v=1}^{K} \left\|\mathbf{w}_v - \mathbf{w}_v^{old}\right\|^2 + C\xi$ $s.t.\ \ \forall u \neq y. (\mathbf{w}_y \cdot \mathbf{x} - \mathbf{w}_u \cdot \mathbf{x}) \geq 1 - \xi, \xi \geq 0$	$\mathbf{w}^{new} = \min_{\mathbf{w}} \frac{1}{2}\sum_{v=1}^{K} \left\|\mathbf{w}_v - \mathbf{w}_v^{old}\right\|^2 + C\xi^2$ $s.t.\ \ \forall u \neq y. (\mathbf{w}_y \cdot \mathbf{x} - \mathbf{w}_u \cdot \mathbf{x}) \geq 1 - \xi$
Stepsize	$\tau_v = Q_v - \sum_{u \in S} \frac{Q_u}{\lvert S\rvert} + \frac{C}{\lvert S\rvert}$	$\tau_v = Q_v - \left(\frac{\lVert\mathbf{x}\rVert^2 + \frac{1}{2C}}{(\lvert S\rvert+1)\lVert\mathbf{x}\rVert^2 + \frac{1}{2C}\lvert S\rvert}\right) \sum_{u \in S} Q_u$
Support class	$\sum_{j=1}^{k-1} \frac{Q_j}{k-1} \geq Q_k - \frac{C}{k}$	$\left(\frac{\lVert\mathbf{x}\rVert^2 + \frac{1}{2C}}{k\lVert\mathbf{x}\rVert^2 + \frac{k-1}{2C}}\right) \sum_{j=1}^{k-1} Q_j \geq Q_k$

表 3 各データセットの属性

	# of classes	# of data	# of features	classes
20Newsgroups	8	2,586	9,971	autos, baseball, graphics, mac, med, motor, politics, space
USPS	10	7,291	256	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Reuters	20	7,800	34488	acq, alum, cocoa, coffee, copper, cpi, crude, earn, gnp, gold, grain, interest, jobs, money-fx, money-supply, reserves, rubber, ship, sugar, trade

表 4 各データセットの誤分類率 (%)

	online							batch	
	PA	PA-I	PA-II	SPA	SPA-I	SPA-II	Perc	Maxent	SVM
20Newsgroups	11.06	10.67	10.17	7.73	7.42	**7.19**	10.87	**7.12**	7.47
USPS	7.12	7.12	6.02	6.30	6.46	**4.46**	6.86	4.98	**4.94**
Reuters	4.31	4.22	4.18	**3.18**	3.30	3.37	4.33	3.50	**3.13**

Newsgroups corpus は約 20,000 の文書からなり、20 の異なるグループに分けられている。この文書集合の中に用意されているセクションの中で “ol-8-1” と呼ばれているものを用いた。これは 8 つのジャンルを示すグループに分けられており素性は BoW で与えられている。すなわち、全ての単語に対応する次元があり、文書においてはそれらの単語の出現回数をその次元の値とする。したがってこれらは非常に高い次元をもつ疎なベクトルとなる。USPS（米国郵便公社）データセットは手書き文字認識[3] のためのデータの一つでありそれぞれの 16 × 16 のピクセルに対して対応するピクセルの濃度を値に持つベクトルをデータとしたものである。Reuters-21578 は上記の Web サイトから入手したデータを整形し、20 クラス分類のデータを作成した。これらの詳細の属性は表 3 に示している。

PA,PA-I,PA-II, 及び SPA,SPA-I,SPA-II に対しそれぞれの評価実験の結果を表 4 に示す。評価は 10 等分における交差検証 (クロスバリデーション) を行い、そのうちの一つを無作為に選び超パラメータ（C）の調節を行った。誤分類率はそれらの結果の平均である。また逐次学習において代表的なアルゴリズムであるパーセプトロン[2](表中”Perc”) に対しても同様の評価を行った。どのデータセットにおいても既存の各手法に比べて良い性能を示している。

また、これらの逐次学習アルゴリズムに加えて、2 つのバッチ学習アルゴリズムの精度も検証し比較した。バッチ学習器は一つのデータセット全体に対して最適化を行う学習器である。ひとつは 最大エントロピーモデル[4] (表中”Maxent”：対数線形モデル, 多クラスロジスティック回帰としても知られている) であり、もうひとつは多クラス SVM[5,6] (表中”SVM”：サポートベクターマシン) である。なおカーネルには線形カーネルを用いた。

バッチ学習は性能の良さのために様々のタスクに使用されているが、逐次学習器はその時間やメモリーの効率と、アルゴリズムの簡潔さから広く使われている。一般的には同じデータにバッチ学習器が適用できるのならば逐次学習よりも良い性能を示す。バッチ学習器は長い CPU 時間と大量のメモリーを使用する。表 4 では Maxent 及び SVM の精度を示している。超パラメータは逐次学習器と同様にして定めた。20 Newsgroups ではバッチ学習の方が良い精度を得ているが、USPS データセットでは SPA-II アルゴリズムがバッチ学習器よりよい精度を達成した。

5. 結　　論

我々は逐次的に現在のデータを正しく分類することを保障する新しい多クラス分類における PA アルゴリズムを提案した。この方法は更新の際に見逃すと学習が十分にできないクラスを複数選びとる必要があり、結果としてサポートクラスの概念を導く。全ての正解スコアよりも高いスコアを持つクラスが更新に寄与するとは限らないので、サポートクラスの決定は自明な問題ではない。しかしながら、我々は簡単な判断基準によってそれらがサポートクラスであるか否かを判定できることを証明した。結果として得られた更新を用いて、他の逐次学習アルゴリズムに比べ、性能の面でより優良であり、バッチ学習器にも匹敵することを実際の実験で示した。

参 考 文 献

1) Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585, 2006.

2) F.Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.
3) J.J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, 1994.
4) Robert Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *COLING-02: proceedings of the 6th conference on Natural language learning*, pages 1–7, Morristown, NJ, USA, 2002. Association for Computational Linguistics.
5) N.Cristianini and J.Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.
6) K.Crammer and Y.Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Jornal of Machine Learning Research*, 2:265–292, 2002.

付　　録

ここではアルゴリズムの導出の詳細を述べる。

A.1 ステップサイズの導出

サポートクラス集合を S とおく。式 (4) と式 (7) を用いて、

$$\left(\mathbf{w}_y^{old}+\sum_{u\in S}\tau_u\mathbf{x}\right)\cdot\mathbf{x}-\left(\mathbf{w}_u^{old}-\tau_u\mathbf{x}\right)\cdot\mathbf{x}=1$$

がわかる。

$$\tau_u\|\mathbf{x}\|^2+\sum_{u\in S}\tau_u\|\mathbf{x}\|^2=1-\left(\mathbf{w}_y^{old}\cdot\mathbf{x}-\mathbf{w}_u^{old}\cdot\mathbf{x}\right)\tag{10}$$

全ての $v\in S$ に対して上式の和をとり整理すると

$$\sum_{u\in S}\tau_u=\frac{\sum_{u\in S}Q_u}{|S|+1}\tag{11}$$

となる。ここで、

$$Q_u=\frac{1-\left(\mathbf{w}_y^{old}\cdot\mathbf{x}-\mathbf{w}_u^{old}\cdot\mathbf{x}\right)}{\|\mathbf{x}\|^2}$$

である。よって再び式 (10) を用いて、

$$\tau_v=Q_v-\sum_{u\in S}\frac{Q_u}{|S|+1}\tag{12}$$

を得る。ここで Q_u は $\mathbf{w}^{old},\mathbf{x},y$ が与えられれば計算可能であるので、サポートクラスが与えられればステップサイズは容易に計算可能であることがわかる。

A.2 サポートクラスの決定 (補題の証明)

以下では次の補題を証明する。

補題 (サポートクラス条件). Q_u の降順に並び変え、順に Q_j とする。Q_j がサポートクラスである必要十分条件は、

$$\sum_{j=1}^{k-1}\frac{Q_j}{k}<Q_k\tag{8}$$

となる。ただしここで $\sum_{j=1}^{0}\bullet=0$ とする。

最初に、サポートクラスは Q_u がある値よりも大きいクラス集合となることを示す。サポートクラスでないクラス v に対し、式 (6) が成立する。すなわち

$$\left(\mathbf{w}_y^{old}+\sum_{u\in S}\tau_u\mathbf{x}\right)\cdot\mathbf{x}-\left(\mathbf{w}_u^{old}-\tau_u\mathbf{x}\right)\cdot\mathbf{x}\geqslant 1\tag{13}$$

である。τ_v は 0 であるのでこれを変形し Q_v に対する以下の不等式を得る。

$$\sum_{u\in S}\tau_u=\frac{\sum_{u\in S}Q_u}{|S|+1}\geq Q_v\quad\forall v\notin S\tag{14}$$

ここで、式 (11) を用いた。一方サポートクラスの v は、$\tau_v>0$, 式 (12) より

$$\sum_{u\in S}\frac{Q_u}{|S|+1}<Q_v\ \forall v\in S\tag{15}$$

を得る。したがってこの二つの不等式から、サポートクラスは Q_u がある値よりも大きいクラス集合となることがわかる。

ここで証明の本題に入る。

(十分性) Q_k が 式 (8) を満たすとする。

$$\sum_{j=1}^{k-1}\frac{Q_j}{(k-1)+1}<Q_k\Leftrightarrow\sum_{j=1}^{k-1}Q_j<kQ_k\Rightarrow\sum_{j=1}^{J}Q_j<(J+1)Q_k\Leftrightarrow\sum_{j=1}^{J}\frac{Q_j}{J+1}<Q_k$$

(三番目の不等式は Q_j の単調性に依る。) したがって k に対応するクラスはサポートクラスである。(式 (15)).

(必要性) Q_k が 式 (8) を満たさないとするとする。

$$\sum_{j=1}^{k-1}\frac{Q_j}{k}\geq Q_k\Leftrightarrow\sum_{j=1}^{k-1}Q_j\geq kQ_k\Leftrightarrow\sum_{j=1}^{k}Q_j\geq(k+1)Q_k\Rightarrow\sum_{j=1}^{k}\frac{Q_j}{k+1}\geq Q_k\geq Q_{k+1}$$

したがって k よりも大きい任意の番号 j で式 (15) を満たさない。したがって $J<k$ であるので、k に対応するクラスはサポートクラスではない。よって題意は示された。(証明終わり)