数理解析研究所講究録
第1649巻 2009年 137-144

最大エントロピー原理に基づくオンライン学習

大田 貴文[1)], 畑埜 晃平[2)], 竹田 正幸[2)]
九州大学大学院システム情報科学府情報理学専攻[1)]
九州大学大学院システム情報科学研究院情報理学部門[2)]
{Takafumi.Ohta, hatano, takeda}@i.kyushu-u.ac.jp

概要

我々は ∞-ノルムマージン最大化超平面を求めるオンライン学習アルゴリズムを提案する．本アルゴリズムは，∞-ノルムマージン γ^* で線形分離可能な n 次元データに対して，$O(\frac{\ln n}{\varepsilon^3})$ の更新回数で，$\gamma^* - \varepsilon$ 以上の超平面を出力する．この更新回数の理論的評価は従来手法に劣るものの，本アルゴリズムは実行速度において従来手法をはるかに上回ることを実験によって示す．

1 序論

近年，疎な線形予測器の学習が注目を集めている．大まかに言えば，疎な線形予測器とは，(巨大な特徴空間上の) 少数の“重要”な特徴のみによって出力が決まるような線形予測器を指す．疎な線形予測器は多くの応用を持つ．例えば，線形回帰における LASSO [14]，信号処理における compressed sensing [2] などが挙げられる．疎な線形予測器の利点の1つは，特徴選択に用いる事が出来る点である．多くの応用においては，予測性能の良さだけでなく，どの特徴が予測性能に寄与するかを知ることは非常に重要である．そこで，疎な特徴をもつ予測性能の高い線形予測器を学習することにより重要な特徴の選択に役立つ．

特に，学習データの爆発的な増加に伴い，高速なオンライン学習アルゴリズムが求められている．その背景には，情報爆発に伴い，学習に使用されるデータの量が爆発的に増加してきたため，SVM に代表される既存のオフライン学習という手法では計算時間が非常にかかるようになってきたことがあげられる．オンライン学習アルゴリズムはメモリを消費せず，そのためストリームデータ上の学習にも適している．また，オンラインアルゴリズムを“1パス”(学習データを列とみなしオンライン学習アルゴリズムを1通り走らせる) ないし数パス走らせることにより，オフライン型の学習アルゴリズムに匹敵する予測性能が得られるという結果も報告されている [1]．

2値分類問題における疎な線形分類器を学習する手法として ∞-ノルムマージンを最大化超平面を学習方法が知られている．∞-ノルムマージン最大化超平面とは，データ (正例・負例) と自身との“距離”を最大化するような超平面をいう，ただし，ここでの距離は ∞-ノルムによって定義される．オフライン学習アルゴリズムの中で，∞-ノルムマージンを最大化するアルゴリズムは線形計画法や Boosting などがあげられるが，オンライン学習アルゴリズムにおいて，∞-ノルムマージンを最大化する効率のよいアルゴリズムは今のところ存在しない．

関連研究にまず Winnow アルゴリズム [8] が挙げられる．データを ∞-ノルムマージン γ^* で線形分離するような超平面が存在するならば，Winnow アルゴリズムはデータを線型分離する超平面を $O(\frac{\ln n}{\gamma^{*2}})$ 回の更新回数で学習できる (n はデータの次元のサイズ)．後に，Winnow アルゴリズムは制約付きのエントロピー最大化問題を解くことで導出できることが明らかになった．[1][16, 9]．最大エントロピー原理に基づく他の分類手法として，Jaakkola らのアルゴリズム [7], regulazied Winnow [16]，そして ROME [9] などがある．しかし，Winnow を含め，これらのアルゴリズムは ∞-ノルムマージンを最大化するという性質を持たない．

∞-ノルムマージンを最大化する代替的なアプローチの1つは p-ノルムを用いることである．Winnow アルゴリズムや Perceptron アルゴリズム [12] の拡張として p-norm Perceptron アルゴリズム [5, 4] が挙げられる．このアルゴリズムは，データが p-ノルムマージン γ で分離できる場合に $O(1/\gamma^2)$ 回の更新回数で線形分離超平面を学習する．特に，$p = O(\ln n)$ の時，p-norm Perceptron アルゴリズムは Winnow アルゴリズムと同様に振る舞うことが知られている [4]．さらに，その拡張版である ALMA [3] や PUMMA [6] は，近似的に最大 p-ノルムマージン超平面を学習する．更新回数は $O(\frac{p}{\varepsilon^2\gamma^2})$ であり，得られるマージンは $(1-\varepsilon)\gamma$ 以上である．特に，$p = O(\ln n)$ のとき，∞-ノルムマージンも近似的に最大化できる．厳密には，$p = c\ln n$ とおくと，∞-ノルムマージン $(1-\varepsilon')\gamma^*/e^{\frac{1}{c}}$ 以上の超平面を $O(\frac{c\ln n}{\varepsilon'^2\gamma^{*2}})$ 回の更新回数で学習する．よって，$c = 1/\ln\frac{1-\varepsilon'}{1-\varepsilon}$ とすることにより，$(1-\varepsilon')\gamma^*$ 以上の ∞-ノルムマージンを達成できる．しかし，結果として更新回数は $O(\frac{\ln n}{\varepsilon^3\gamma^{*2}})$ となり，$O(\frac{1}{\varepsilon})$ 倍余計に時間がかかってしまう．

本研究において，我々は，∞-ノルムマージン最

[1] より厳密には，元々の Winnow アルゴリズムは非正規化相対エントロピーと呼ばれるエントロピーと関連した量を最大化する．

大化超平面を近似的に求めるオンライン学習アルゴリズム MEMMA(Maximum Entropy Maximum Margin Algorithm) を提案する．MEMMA は $O(\frac{\ln n}{\varepsilon^3})$ の更新回数で $\gamma^* - \varepsilon$ 以上の ∞-ノルムマージンを持つ超平面を出力する．更新回数の理論的上限は PUMMA に劣るものの，本実験においては，MEMMA は PUMMA よりもはるかに高速に動作する事を示す．また，本実験の結果は MEMMA の更新回数が $O(\frac{\ln n}{\varepsilon^2})$ であることを示唆しており，更新回数の理論的上限は改善の余地がある．

我々の手法は線形計画問題における摂動 (perturbation) のアプローチ [11] に基づく．元々の ∞-ノルムマージン最大超平面を求めるバッチ学習問題は線形計画問題で定式化できる．一方，我々は，マージンだけではなく線形分類器のエントロピーも最大化するような問題を設定している．これは，元の線形計画問題の目的関数にエントロピー項を加えることで実現でき，結果として，我々の扱う問題は凸計画問題となる．実際，線形計画問題の目的関数に十分小さい凸項を加える事により，得られる凸計画問題は一意な最適解を持ち，さらに，その解は元の線形計画問題を最適化することが知られている [11]．したがって，我々の凸計画問題は元の線形計画問題を近似すると期待できる．

さらに，我々はその凸計画問題を 2 個の線形制約を持つ緩和された凸計画問題の列に帰着させる．ここで，各緩和された凸計画問題はニュートン法など勾配を用いた最適化手法によって高速に解くことが可能である．結果として，我々は，元の線形計画問題をオンライン的に解くことができる．

同様のアプローチを用いたバッチ式の学習アルゴリズムとして，Mangasarian のアルゴリズム [10] や Warmuth らによる Entropy Regularized LP-Boost [15] が挙げられる．特に，我々の手法は後者のアプローチに啓発されたものである．

一方，本論文では，元の線形計画問題の目的関数にエントロピー項を追加しないでオンライン的に解く場合，$\Omega(\frac{n}{\gamma^*})$ 回の更新を要することも示す．よって，エントロピー項の追加により，本手法は更新回数を次元数 n に対して指数的に改善していることがいえる．

2 準備

$\mathcal{X} \subset \mathbb{R}^n$ を事例空間と呼ぶ．また，事例空間 $\mathcal{X}$ の要素 $\boldsymbol{x}$ を事例またはインスタンスと呼び，$||\boldsymbol{x}||_\infty \leq 1$ である．$||\cdot||_\infty$ については後述する．各事例 $\boldsymbol{x}$ はラベル $y \in \{-1, +1\}$ をもつ．事例とラベルの組 $(\boldsymbol{x}, y)$ を例と呼び，ラベルが $+1$ の例を正例，-1 の例を負例と呼ぶ．

n 次元の確率分布の集合を $\mathcal{P}^n = \{\boldsymbol{p} \in [0,1]^n \mid \sum_{i=1}^n p_i = 1\}$ とする．各 $\boldsymbol{p} \in \mathcal{P}^n$ に対して，エントロピー $H(\boldsymbol{p})$ は $H(\boldsymbol{p}) = -\sum_{i=1}^n p_i \ln p_i$ と定義される．

また，b をバイアスと呼び，重みベクトル $\boldsymbol{p}$ の原点からの離れ具合を表す．重みベクトルとバイアスの組 $(\boldsymbol{p}, b)$ を超平面と呼ぶ．各事例 $\boldsymbol{x}$ のラベル y は $y = \mathrm{sign}(\boldsymbol{p}^* \cdot \boldsymbol{x} + b^*)$ で与えられる．この $\boldsymbol{p}^*$ を真の重みベクトル，b^* を真のバイアス，$(\boldsymbol{p}^*, b^*)$ を真の超平面と呼ぶ．

p-ノルムについて説明する．ノルムとは，距離の定義のひとつであり $\boldsymbol{x} \in \mathbb{R}^n$ の p-ノルムは $||\boldsymbol{x}||_p$ と表される．その値は以下の式で定義される．

$$||\boldsymbol{x}||_p = \left(\sum_{i=1}^n |x_i|^p\right)^{1/p}$$

ただし，

$$||\boldsymbol{x}||_\infty = \lim_{p\to\infty}\left(\sum_{i=1}^n |x_i|^p\right)^{\frac{1}{p}} = \max_{i=1\ldots n} |x_i|$$

今回我々が扱うノルムは，この ∞-ノルムである．

次に，マージンについて説明する．例集合 $S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_T, y_T)\}$ に対して，p-ノルムマージンを $\gamma_p = \min_{i=1\ldots T} y_i(\boldsymbol{p}\cdot\boldsymbol{x}_i + b)/||\boldsymbol{p}||_q$ (ただし，$1/p + 1/q = 1$) と定義する．この定義により，例集合 S に対する ∞-ノルムマージン γ_∞ (または単にマージン γ) は $\gamma_\infty = \min_{i=1\ldots T} y_i(\boldsymbol{p}\cdot\boldsymbol{x}_i + b)$ である．また，真の超平面 $(\boldsymbol{p}^*, b^*)$ が与えられた時のマージン γ^* を真のマージンと呼ぶ．マージンの値は，様々なオンライン学習アルゴリズムにおいて，その更新回数を決める重要な要素である．

次にオンライン学習の基本的な流れについて説明する．基本的な流れは以下のとおりである．

t 回目の試行において，事例 $\boldsymbol{x}_t$ を受け取る．オンライン学習アルゴリズムは，予測のための超平面 $(\boldsymbol{p}_t, b_t)$ を用いて，ラベルの予測 $\hat{y}_t = \mathrm{sign}(\boldsymbol{p}_t \cdot \boldsymbol{x}_t + b_t)$ を計算する．そして真のラベル y_t を受け取り，もし間違っていたら，すなわち $\hat{y}_t \neq y_t$ なら $(\boldsymbol{p}_{t+1}, b_{t+1}) = \mathrm{UPDATE}(\boldsymbol{w}_t, b_t)$ として，超平面 $(\boldsymbol{p}, b))$ を更新する．予測が正しければ，$(\boldsymbol{x}_{t+1}, b_{t+1}) = (\boldsymbol{w}_t, b_t)$ とし，更新を行わない．この更新関数 UPDATE が各アルゴリズムによって異なり，オンライン学習アルゴリズムの性能は，更新が行われなくなるまでの更新回数によって評価される．

最後に，我々のオンライン学習アルゴリズムの目標を説明する．入力として，パラメータ ε $(0 < \varepsilon < 1)$，および，十分長い例の列 $S = ((\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_T, y_T))$ が与えられるとする．ただし，S に対して ∞ ノルムマージン γ^* をもつ超平面 $(\boldsymbol{p}^*, b^*)$ が存在すると仮定する．このとき，なるべく小さい更新回数で ∞ ノルムマージンが $\gamma^* - \varepsilon$ 以上の超平面を出力することがオンライン学習アルゴリズムの目標である．

3 アルゴリズム

本章では，今回我々が提案する ∞-ノルムマージン最大化オンライン学習アルゴリズム MEMMA について紹介する．この MEMMA は ∞-ノルムマージンを最大化するオフライン学習アルゴリズムの考えを改良し，オンライン学習に適応させたものであるので，まずは ∞-ノルムマージンを最大化するオフライン学習アルゴリズムについて紹介する．

3.1 ∞-ノルムマージン最大化オフライン学習

まず，オフライン学習において ∞-ノルムマージンを最大化するもっとも単純な学習アルゴリズムを考える．そのアルゴリズムは，以下のようにして重み

ベクトル $\boldsymbol{p}$, バイアス b , マージン γ を求める.

$$(\tilde{\boldsymbol{p}}, \tilde{b}, \tilde{\gamma}) = \arg \max_{\boldsymbol{p}\in\mathcal{P}, b, \gamma\in\mathbb{R}} \gamma \qquad (1)$$

subject to :

$y_i(\boldsymbol{p}\cdot\boldsymbol{x}_i + b) \geq \gamma$

$(1 \leq i \leq T)$

これによって求まる超平面 $(\boldsymbol{p}, b)$ は例集合 S の全ての要素 $(\boldsymbol{x}, y)$ に対して, $y(\boldsymbol{p}\cdot\boldsymbol{x}+b) \geq \gamma$ を満たし, かつその γ は最大となるので, 得られる超平面 $(\boldsymbol{p}, b)$ が例集合 S に対してもつマージン $\min_{(\boldsymbol{x},y)\in S} y(\boldsymbol{p}\cdot\boldsymbol{x}+b)$ は最大となる.

この考えをオンライン学習に適応させたアルゴリズムを次節で説明する.

3.2 ∞-ノルムマージン最大化オンライン学習

前節で紹介した考え方を単純にオンライン学習に適応したアルゴリズムを図 1 に示す.

begin

1. (初期化) $\boldsymbol{p}_1 = (\frac{1}{n}, \ldots, \frac{1}{n}) \in \mathcal{P}^N$. $\gamma_1 = 1, b_1 = 0$ とする.
2. For $t = 1$ to T,
   (a) 事例 $\boldsymbol{x}_t$ を受け取る.
   (b) 予測 $\hat{y}_t = \mathrm{sign}(\boldsymbol{p}_t\cdot\boldsymbol{x}_t + b_t)$ を計算する.
   (c) ラベル y_t を受け取る.
   (d) もし, $y_t(\boldsymbol{p}_t\cdot\boldsymbol{x}_t + b_t) < \gamma_t - \varepsilon$ なら以下のように更新 :

$$(\boldsymbol{p}_{t+1}, b_{t+1}, \gamma_{t+1}) = \arg \max_{\boldsymbol{p}\in\mathcal{P}, b, \gamma\in\mathbb{R}} \gamma$$

subject to :

$y_i(\boldsymbol{p}\cdot\boldsymbol{x}_i + b) \geq \gamma \quad (1 \leq i \leq t)$

end.

図 1: ∞-ノルムマージン最大化オンライン学習アルゴリズム.

このアルゴリズムは, 線形計画問題として定式化でき, 内点法等を用いて多項式時間で解くことができる. また, その性能に関しては以下の定理が成り立つ.

定理 1. このアルゴリズムが γ^* のマージンを得るために必要な更新回数の下界は以下の式で与えられる.

$$\Omega\left(\frac{n}{\gamma^*}\right)$$

証明. 簡単のため, γ^* が $1/\gamma^*$ が整数になるような値であるとする. このとき, 学習データ $S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_t, y_t)\}$ を考える. 各例 $(\boldsymbol{x}_j, y_j)$ は, $j = kn + l (k = 0, ..., l < n)$ の場合以下の性質を満たす.

$$y_j x_{j,i} = \begin{cases} 1 - k\gamma^* - (i-1)\delta, & i < l \\ 1 - (k+1)\gamma^* - (i-1)\delta, & i \geq l. \end{cases}$$

事例	次元		
t	1	2	3
$y_1\boldsymbol{x}_1$	1	$1-\delta$	$1-2\delta$
$y_2\boldsymbol{x}_2$	$1-\gamma^*$	$1-\delta$	$1-2\delta$
$y_3\boldsymbol{x}_3$	$1-\gamma^*$	$1-\gamma^*-\delta$	$1-2\delta$
$y_4\boldsymbol{x}_4$	$1-\gamma^*$	$1-\gamma^*-\delta$	$1-\gamma^*-2\delta$
$y_5\boldsymbol{x}_5$	$1-\gamma^*$	$1-\gamma^*-\delta$	$1-\gamma^*-2\delta$
$y_6\boldsymbol{x}_6$	$1-2\gamma^*$	$1-\gamma^*-\delta$	$1-\gamma^*-2\delta$
$y_7\boldsymbol{x}_7$	$1-2\gamma^*$	$1-2\gamma^*-\delta$	$1-\gamma^*-2\delta$
$y_8\boldsymbol{x}_8$	$1-2\gamma^*$	$1-2\gamma^*-\delta$	$1-2\gamma^*-2\delta$
...	...	...	...
$y_t\boldsymbol{x}_t$	γ^*	$\gamma^*-\delta$	$\gamma^*-2\delta$

表 1: $n = 3$ の場合のデータ例.

ここで, $\delta = \frac{\gamma^*}{n}$ である. $n = 3$ の場合を例として表 1 に示す. 各例のラベルは任意に決定することができる. 最初の例をのぞいて, アルゴリズムが更新を行うようなラベルを与える. すると, 各試行 j において, 次の性質をもつ.

1. 各試行 j において, アルゴリズムは更新を行う. その結果, 重みベクトル $\boldsymbol{p}_{j+1}$ は l 番目の要素が 1 となる単位ベクトルとなる.
2. S_j に対する $\boldsymbol{p}_{j+1}$ のマージンは $1-k\gamma^*-(l-1)\delta$ となる. 特に $y_j\boldsymbol{p}_{j+1}\cdot\boldsymbol{x}_j = 1-k\gamma^*-(l-1)\delta$ である.

よって, 試行 $t = n(\frac{1}{\gamma^*} - 1) + 1$, 得られるマージンは γ^* となる.

□

更新回数の式を見てわかるとおり, このアルゴリズムは次元数が増えれば増えるほど, 更新回数も同じく増加してしまう. また, 制約の数も次元例の数の増加とともに増え, それに伴い計算時間も増加してしまうという問題がある. そのため, このアルゴリズムは良いアルゴリズムとは言えない. そこで, 3.1 の更新式を改良させたオフライン学習を考える.

3.3 エントロピー項を考慮した ∞-ノルムマージン最大化オフライン学習

3.1 の更新式の目的関数にエントロピー項を追加した更新式を考える. その更新方法は以下のようになる.

$$(\hat{\boldsymbol{p}}, \hat{b}, \hat{\gamma}) = \arg \max_{\boldsymbol{p}\in\mathcal{P}, b, \gamma\in\mathbb{R}} \gamma \quad + \eta H(\boldsymbol{p}) \qquad (2)$$

subject to :

$y_i(\boldsymbol{p}\cdot\boldsymbol{x}_i + b) \geq \gamma$

$(1 \leq i \leq T)$

ここで, η はマージンとエントロピーのトレードオフ変数である.

エントロピー項を加えることで, 以下の定理が成り立つ.

定理 2 (Cf. Mangasarian, Meyer[11]). 十分小さい $\bar{\eta} > 0$ に対して, $\eta < \bar{\eta}$ を満たす場合, (2) 式の最適解 $(\hat{\boldsymbol{p}}, \hat{b}, \hat{\gamma})$ は (1) 式の最適解の 1 つである. また, $(\hat{\boldsymbol{p}}, \hat{b}, \hat{\gamma})$ は (1) 式の最適解の中でエントロピー $H(\boldsymbol{p})$ を最大にするものである.

この定理の証明は [11] にてほぼ同様の証明がなされているので，ここでは省略する．この定理により，エントロピー項を加えることで，元の線形計画問題を狭義の凸計画問題に置き換えることができ，それにより，最適解が一意に定まる．また，その解は元の線形計画問題を最適化することが言える．

この更新方法をオンライン学習に適応させたものが次節で紹介する MEMMA アルゴリズムである．もちろん，単純にオンライン学習の形に変えただけでは，制約の数は事例の数に等しくなってしまう，バイアス項が直接計算できないなどの問題が存在するため，その問題を解決するよう更新方法に若干の工夫を行った．

MEMMA(ε)
begin

1. (初期化)　正例，負例　$(\boldsymbol{x}_1^{pos}, +1)$, $(\boldsymbol{x}_1^{neg}, -1)$ を 1 つずつ受け取る．
2. For $t = 1$ to T,
   (a) 事例 $\boldsymbol{x}_t$ を受け取る．
   (b) 以下のように式を更新：

$$(p_t, b_t, \gamma_t) = \arg \max_{\boldsymbol{p} \in \mathcal{P}, b, \gamma \in \mathbb{R}} \gamma + \eta H(\boldsymbol{p})$$

subject to :

$$(\boldsymbol{p} \cdot \boldsymbol{x}_t^{pos} + b) \geq \gamma,$$

$$(\boldsymbol{p} \cdot \boldsymbol{x}_t^{neg} + b) \leq -\gamma, \text{ and}$$

$$\gamma - \eta \boldsymbol{p} \cdot \ln(\boldsymbol{p}_{t-1}) \leq \gamma_{t-1} + \eta H(\boldsymbol{p}_{t-1}),$$

ただし $\eta = \frac{\varepsilon}{C \ln n} (C > 2)$.

   (c) 予測 $\hat{y}_t = \mathrm{sign}(\boldsymbol{p}_t \cdot \boldsymbol{x}_t + b_t)$ を計算．
   (d) ラベル y_t を受け取る．
   (e) もし，$y_t(\boldsymbol{p}_t \cdot \boldsymbol{x}_t + b_t) < \gamma_t - \varepsilon_t$ ならば以下のように更新（ただし，$\varepsilon_t = \varepsilon - \eta H(p_t)$）：

$$(\boldsymbol{x}_{t+1}^{pos}, \boldsymbol{x}_{t+1}^{neg}) = \begin{cases} (\boldsymbol{x}_t, \boldsymbol{x}_t^{neg}), & (y_t = +1) \\ (\boldsymbol{x}_t^{pos}, \boldsymbol{x}_t), & (y_t = -1). \end{cases}$$

そうでなければ以下のように更新：

$$(\boldsymbol{x}_{t+1}^{pos}, \boldsymbol{x}_{t+1}^{neg}) = (\boldsymbol{x}_t^{pos}, \boldsymbol{x}_t^{neg}).$$

end.

図 2: MEMMA アルゴリズム．

3.4 Maximum Entropy Maximum Margin Algorithm

本節では，我々の提案する MEMMA アルゴリズムを紹介する．我々の提案するアルゴリズムを図 2 に示す．アルゴリズムの大まかな流れは，オンライン学習の各試行 t において現在のマージン γ_t，重みベクトル $\boldsymbol{p}_t$，バイアス b_t を用いてラベルの予測を行った際に，予測が間違えてしまう場合に以下のように γ，$\boldsymbol{p}$，b を更新する．

$$(p_t, b_t, \gamma_t) = \arg \max_{\boldsymbol{p} \in \mathcal{P}, b, \gamma \in \mathbb{R}} \gamma + \eta H(\boldsymbol{p}) \qquad (3)$$

subject to :

$$(\boldsymbol{p} \cdot \boldsymbol{x}_t^{pos} + b) \geq \gamma,$$

$$(\boldsymbol{p} \cdot \boldsymbol{x}_t^{neg} + b) \leq -\gamma,$$

$$-\gamma + \eta \boldsymbol{p} \cdot \ln(\boldsymbol{p}_{t-1}) \geq -\gamma_{t-1} - \eta H(\boldsymbol{p}_{t-1}).$$

ここで，η はマージンとエントロピーのトレードオフ変数，$\boldsymbol{x}_t^{pos}, \boldsymbol{x}_t^{neg}$ はそれぞれもっとも最近間違えた正例，負例である．

この更新式の解 $\gamma, \boldsymbol{p}, b$ は解析的に求めることはできないが，ニュートン法とラグランジュの未定乗数法を用いることで数値的に求めることができる．具体的には解は以下の様に与えられる．

$$p_{t,i} = \frac{p_{t-1,i}^{\beta} e^{\frac{\alpha}{\eta}(x_{t,i}^{pos} - x_{t,i}^{neg})}}{\sum_i^n p_{t-1,i}^{\beta} e^{\frac{\alpha}{\eta}(x_{t,i}^{pos} - x_{t,i}^{neg})}}$$

ただし，α，β はラグランジュ未定乗数であり，次の最適化問題の解である．

$$\max_{\alpha, \beta} \Theta(\alpha, \beta)$$

subject to:

$$2\alpha + \beta = 1$$

$$\alpha \geq 0, \beta \geq 0.$$

ただし，

$$\Theta(\alpha, \beta) = -\beta(\gamma_{t-1} + \eta H(\boldsymbol{p}_{t-1})) - \eta \ln \sum_i p_{t-1,i}^{\beta} e^{\frac{\alpha}{\eta}(x_{t,i}^{pos} - x_{t,i}^{neg})}.$$

3.5 収束の証明

先ず，MEMMA アルゴリズムによって得られる超平面，マージンと，真の超平面，真のマージンとの関係を述べる．

補題 1. 真のマージンを γ^*，真の重みベクトルを p^* としたとき，真の更新式と，ラウンド t 時 $(t = 1...T)$ の更新式との関係は以下を満たす．

$$\gamma^* - \eta \boldsymbol{p}^* \cdot \ln \boldsymbol{p}_t \leq \gamma_t + \eta H(\boldsymbol{p}_t). \qquad (4)$$

証明. t の値で場合分けして考える．

(1) $t = 1$ のとき

γ，p の初期値は $p_{1,i} = \frac{1}{n}, \gamma_1 = 1$ なので，更新式は

$$(\text{左辺}) = \gamma^* - \eta \sum_i^n p_i^* \ln p_{1,i} = \gamma^* - \eta \ln \frac{1}{n},$$

$$(\text{右辺}) = \gamma_1 - \eta \sum_i^n p_{1,i} \ln p_{1,i} = 1 - \eta \ln \frac{1}{n}.$$

$0 \leq \gamma^* \leq 1$ なので，

$$\gamma^* - \eta \ln \frac{1}{n} \leq 1 - \eta \ln \frac{1}{n}.$$

よって，$t = 1$ のとき (4) が成立する．

(2) $t \geq 2$のとき

$t = k$のとき(4)が成立すると仮定すると，$t = k+1$のとき示したい式は

$$\begin{aligned} & \gamma^* - \eta \textstyle\sum_i^n p_i^* \ln p_{k+1,i} \\ \leq \quad & \gamma_{k+1} - \eta \textstyle\sum_i^n p_{k+1,i} \ln p_{k+1,i}. \end{aligned}$$

この式の左辺と右辺を別々に考える．

$$p_{k+1,i} = \frac{p_{k,i}^{\beta} e^{\frac{\alpha}{\eta}(x_i^{pos} - x_i^{neg})}}{\sum_i^n p_{k,i}^{\beta} e^{\frac{\alpha}{\eta}(x_i^{pos} - x_i^{neg})}}$$

より，

$$\begin{aligned} (\text{左辺}) =& \gamma^* - \eta \sum_i^n p_i^* \ln p_{k+1,i} \\ =& \gamma^* - \eta \sum_i^n p_i^* (\beta \ln p_{k,i} + \frac{\alpha}{\eta}(x_i^{pos} - x_i^{neg}) \\ & - \ln \sum_i^n p_{k,i}^{\beta} e^{\frac{\alpha}{\eta}(x_i^{pos} - x_i^{neg})}) \\ =& \gamma^* - \eta\beta \sum_i^n p_i^* \ln p_{k,i} \\ & - \alpha \sum_i^n p_i^* (x_i^{pos} - x_i^{neg}) \\ & + \eta \sum_i^n p_i^* \ln \sum_i^n p_{k,i}^{\beta} e^{\frac{\alpha}{\eta}(x_i^{pos} - x_i^{neg})} \end{aligned}$$

ここで，$\sum_i^n p_i^* = 1$，また，

$$2\gamma^* \leq \boldsymbol{p}^* \cdot (\boldsymbol{x}^{pos} - \boldsymbol{x}^{neg})$$

なので，

$$\begin{aligned} (\text{左辺}) =& (1 - 2\alpha)\gamma^* - \eta\beta \sum_i^n p_i^* \ln p_{k,i} \\ & + \eta \ln \sum_i^n p_{k,i}^{\beta} e^{\frac{\alpha}{\eta}(x_i^{pos} - x_i^{neg})}. \end{aligned}$$

ここで $2\alpha + \beta = 1$ より，また，簡単のために $\eta \ln \sum_i^n p_{k,i}^{\beta} e^{\frac{\alpha}{\eta}(x_i^{pos} - x_i^{neg})} = C$ とすると

$$(\text{左辺}) = \beta\{\gamma^* - \eta \sum_i^n p_i^* \ln p_{k,i}\} + C. \quad (5)$$

右辺も同様の計算を行うと，

$$(\text{右辺}) = \beta\{\gamma_{k+1} - \eta \sum_i^n p_{k+1,i} \ln p_{k,i}\} + C. \quad (6)$$

上で求めた(5)と(6)の関係をβの値で場合分けして考える．

$\beta = 0$のとき

$$(\text{左辺}) = (\text{右辺}) = C.$$

$\beta \neq 0$のとき

KKT条件より

$$\gamma_{k+1} - \eta \sum_{i=1}^n p_{k+1,i} \ln p_{k,i} = \gamma_k - \eta \sum_{i=1}^n p_{k,i} \ln p_{k,i}$$

が成り立つので，

$$\begin{aligned} (\text{右辺}) &= \beta\{\gamma_{k+1} - \eta \sum_i^n p_{k+1,i} \ln p_{k,i}\} + C \\ &= \beta\{\gamma_k - \eta \sum_{i=1}^n p_{k,i} \ln p_{k,i}\} + C. \end{aligned}$$

$t = k$のとき成立するという仮定より

$$\gamma^* - \eta \sum_i^n p_i^* \ln p_{k,i} \leq \gamma_k - \eta \sum_i^n p_{k,i} \ln p_{k,i}$$

が成り立つことと，$\beta \geq 0$より

$$(\text{左辺}) \leq (\text{右辺})$$

が言える．よって，$\beta = 0, \beta \neq 0$の場合ともに(左辺) $\leq$ (右辺)が成立するので，$t = k$のときに(4)が成り立つと仮定すれば$t = k+1$のときも(4)が成立することが言える．

帰納法により，すべての$t = 1...T$において(4)式

$$\begin{aligned} \gamma^* - \eta \sum_i^n p_i^* \ln p_t \quad \leq \quad \gamma_t - \eta \sum_i^n p_t \ln p_t \end{aligned}$$

が成立することが言える． □

3.6 更新回数の上限

次に，このアルゴリズムが保証する更新回数の理論的上限値について論じる．

補題 2. (*Pinsker*の不等式 [13])
任意の重みベクトル$\boldsymbol{p}, \boldsymbol{q} \in \mathcal{P}$に対して，以下の性質が成り立つ．

$$\Delta(\boldsymbol{p}, \boldsymbol{q}) \geq \frac{1}{2}\|\boldsymbol{p} - \boldsymbol{q}\|_1^2$$

補題 3. (*Hölder*の不等式)
$1 \leq p < \infty, \frac{1}{p} + \frac{1}{q} = 1$とするとき，任意のベクトル$\boldsymbol{x}, \boldsymbol{y}$に対して以下の式が成り立つ．

$$\boldsymbol{x} \cdot \boldsymbol{y} \leq \|\boldsymbol{x}\|_p \|\boldsymbol{y}\|_q$$

補題1より，真の超平面$(\boldsymbol{p}^*, b^*)$とそのマージンγ^*は更新式(3)の解である．ゆえに以下の命題が成り立つ．

命題 1. すべての$t \geq 0$に対して，以下の式が成立する

$$\gamma^* + \eta H(\boldsymbol{p}^*) \leq \gamma_t + \eta H(\boldsymbol{p}_t)$$

この命題1から以下の補題が導き出される

補題 4. ラウンド t 時における，更新前と更新後の更新式 $\gamma+\eta H(\boldsymbol{p})$ の変化量は以下の式であらわすことができる．$\eta=\frac{\varepsilon}{c\ln n},(c>2)$ のとき，

$$\gamma_t+\eta H(\boldsymbol{p}_t)-\gamma_{t+1}-\eta H(\boldsymbol{p}_{t+1})\geq\frac{(c-2)^2\eta\varepsilon^2}{8c^2}$$

証明. γ_t と γ_{t+1} の値の関係で場合分けして考える．

(i) $\gamma_t\geq\gamma_{t+1}+X$ のとき

$\gamma_t+\eta H(\boldsymbol{p}_t)-\gamma_{t+1}-\eta H(\boldsymbol{p}_{t+1})$ を Δ_1 とすると，$H(\boldsymbol{p}_t)\geq 0, H(\boldsymbol{p}_{t+1})\leq\eta\ln n$ より

$$\begin{aligned}\Delta_1&=\gamma_t+\eta H(\boldsymbol{p}_t)-\gamma_{t+1}-\eta H(\boldsymbol{p}_{t+1})\\&\geq X+\eta H(\boldsymbol{p}_t)-\eta H(\boldsymbol{p}_{t+1})\\&\geq X-\eta\ln n.\end{aligned}$$

(ii) $\gamma_t\leq\gamma_{t+1}+X$ のとき

$\gamma_t+\eta H(\boldsymbol{p}_t)-\gamma_{t+1}-\eta H(\boldsymbol{p}_{t+1})$ を Δ_2 とすると

更新時の条件

$$-\gamma_{t+1}+\eta\sum p_{t+1,i}\ln p_{t,i}\geq-\gamma_t+\eta\sum p_{t,i}\ln p_{t,i}$$

より，

$$\begin{aligned}\Delta_2&\geq-\gamma_{t+1}+\eta\sum p_{t+1,i}\ln p_{t+1,i}\\&\quad+\gamma_{t+1}-\eta\sum p_{t+1,i}\ln p_{t,i}\\&=\eta\sum p_{t+1,i}(\ln p_{t+1,i}-\ln p_{t,i})\\&=\eta\sum p_{t+1,i}\ln\frac{p_{t+1,i}}{p_{t,i}}\\&=\eta\Delta(\boldsymbol{p}_{t+1},\boldsymbol{p}_t).\end{aligned}$$

ここで，Pinsker の不等式 (補題 2) より，$\Delta(\boldsymbol{p}_{t+1},\boldsymbol{p}_t)\geq\frac{1}{2}\|\boldsymbol{p}_{t+1}-\boldsymbol{p}_t\|_1^2$ が言える．

さらに，Hölder の不等式 (補題 3) より，

$$\begin{aligned}&\frac{1}{2}\|\boldsymbol{p}_{t+1}-\boldsymbol{p}_t\|_1^2\|y_t\boldsymbol{x}_t\|_\infty^2\\\geq&\frac{1}{2}|(\boldsymbol{p}_{t+1}-\boldsymbol{p}_t)\cdot y_t\boldsymbol{x}_t|^2\\=&\frac{1}{2}|\gamma_{t+1}-y_t\boldsymbol{p}_t\cdot\boldsymbol{x}_t|^2.\end{aligned}$$

また，$\gamma_{t+1}-y_t\boldsymbol{p}_t\cdot\boldsymbol{x}_t$ は

$$\begin{aligned}\gamma_{t+1}-y_t\boldsymbol{p}_t\cdot\boldsymbol{x}_t&\geq\gamma_{t+1}-\gamma_t+\varepsilon_t\\&\geq\varepsilon_t-X,\\&\geq\varepsilon-X-\eta\ln n.\end{aligned}$$

よって，

$$\Delta_2\geq\frac{1}{2}\eta|\varepsilon-X-\eta\ln n|^2$$

となる．

以上の式により，X,η,ε に求められる条件は $\eta\ln n\leq X\leq\varepsilon-\eta\ln n$ なので，$\eta=\frac{\varepsilon}{c\ln n}(c>2)$ である．$\eta=\frac{\varepsilon}{c\ln n}(c>2), X=\frac{\varepsilon}{2}$ とすると

$$\begin{aligned}\Delta_1&\geq\frac{\varepsilon}{2}-\frac{\varepsilon}{c}\\&=\frac{(c-2)\varepsilon}{2c}\\\Delta_2&\geq\frac{1}{2}\eta|\eta-\frac{\varepsilon}{c}-\frac{\varepsilon}{2}|^2\\&=\frac{(c-2)^2\eta\varepsilon^2}{8c^2}\end{aligned}$$

$\Delta_1\geq\Delta_2$ なので，更新式の差分の最小値は Δ_2 となる．□

定理 3. アルゴリズムの更新回数は高々

$$O\left(\frac{\ln n}{\varepsilon^3}\right)$$

回である．また，上記の回数の更新の後，少なくとも

$$\gamma^*-\varepsilon$$

のマージンをもつ超平面を出力する．

証明. 補題 4 より，一度の更新で更新式の値は少なくとも $\frac{(c-2)^2\eta\varepsilon^2}{8c^2}$ だけ変化することが分かっている．ここで，更新式の最小値，最大値を考えると，最小値 0，最大値 $1+\eta\ln(n)$ なので，更新回数の上限 M は

$$\begin{aligned}M&=\frac{1+\eta\ln(n)}{\frac{(c-2)^2\eta\varepsilon^2}{8c^2}}\\&=\frac{8c^3\ln n}{(c-2)^2\varepsilon^3}+\frac{8c^2\ln n}{(c-2)^2\varepsilon^2}.\end{aligned}$$

よって，更新回数の上限は $O\left(\frac{\ln n}{\varepsilon^3}\right)$ である．また，更新の終了条件を考えると，すべての $t(1\leq t\leq T)$ に対して

$$y_t(\boldsymbol{p}_t\cdot\boldsymbol{x}_t+b_t)\geq\gamma_t-\varepsilon_t,$$

ただし $\varepsilon_t=\varepsilon-\eta H(\boldsymbol{p}_t)$．命題 1 より，$\gamma^*+\eta H(\boldsymbol{p}^*)\leq\gamma_t+\eta H(\boldsymbol{p}_t)$ が言える．この式を変形させると

$$\begin{aligned}\gamma_t&\geq\gamma^*+\eta H(\boldsymbol{p}^*)-\eta H(\boldsymbol{p}_t)\\&\geq\gamma^*-\eta H(\boldsymbol{p}_t).\end{aligned}$$

よって，

$$y_t(\boldsymbol{p}_t\cdot\boldsymbol{x}_t+b_t)\geq\gamma^*-\varepsilon.$$

□

4 実験

4.1 他の学習アルゴリズムとの比較

MEMMA の更新回数や，得られるマージンがどのようなものか調べるために，他のオンライン学習アルゴリズムとの比較実験を行った．比較に用いたアルゴリズムは PUMMA アルゴリズムである．PUMMA は p-ノルムマージンを最大化するように学習するアルゴリズムであり，∞-ノルムマージンを直接扱うことはできないが，$p=c\ln n$(ただし $c=1/\ln\frac{1-\delta'}{1-\delta},\delta'=\delta/2$) に設定することで近似的に ∞ ノルムマージンを最大にする学習を行うことができる (ただし，$\delta\gamma^*=\varepsilon$, 保証されるマージンは $(1-\delta)\gamma^*$)．

次に，実験に用いた学習例や実験手順について説明する．事例の次元のサイズは $n=100$ とし，例の個数は 1000 である．例のラベル付けは r-of-k 関数を用いた．r-of-k 関数とは，ある特定の k 個の変数のうち r 個以上が $+1$ なら $+1$ を，そうでないなら -1 を返す論理関数である．$\boldsymbol{x}\in\{+1,-1\}^n$ だとすると，r-of-k 関数 $h_{r,k}$ は

$$h_{f,k}=x_{i_1}+x_{i_2}+\cdots+x_{i_k}+k-2r+1$$

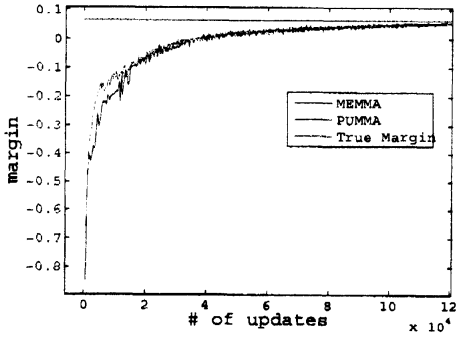

図 3: 人工データに対する更新回数と得られるマージンの値．x 軸が更新回数，y 軸が得られたマージンの値で，グラフの上部にある直線が目標のマージン (真のマージンの 90%である)．

	MEMMA	PUMMA
更新回数	115822	119428
マージン	0.057148	0.057004
実行時間 (秒)	181	936

表 2: MEMMA と PUMMA を同じ条件で作られた 10 個のデータセットに対して実行した際の更新回数，得られたマージン，実行時間の平均値．

となる．ただし $i_j \in \{i \mid i \text{は} n \text{以下の自然数}\}$．このとき，この r-of-k 関数の任意の例集合 $S = \{\boldsymbol{x} \mid \boldsymbol{x} \in \{+1,-1\}^n\}$ に対する真のマージンは $\gamma^* = 1/k$ 以上となる．なお，各例はラベルが正となる確率が 1/2 となるようランダムに生成する．

実験の手順は，MEMMA，PUMMA ともに，すべての学習例に対して，真のマージンの 90% を得られることを終了条件とし，終了までの更新回数，実行時間，実際に得られたマージンを計測する．より具体的には，例集合を列とみなし，その列を入力として各アルゴリズムを一通り実行する．この操作を与えられた入力列に対して更新がゼロになるまで繰り返す．なお，実験に用いた計算機は 3.8GHz Intel Xeon Processor 及び 8 GB Memory を搭載した Linux マシンで，プログラムは MATLAB にて作成した．また，この実験はそれぞれデータの生成から 10 回ずつ行い，その結果を記録した．この実験結果を図 3，表 2 に示す．

この図 3 は更新回数とマージンの値の関係を示す図なのだが，MEMMA と PUMMA はほぼ同じ更新回数でほぼ同じマージンを得ていることがわかる．更新回数や，得られるマージンの詳しい値を示したのが表 2 である．この表を見ると，更新回数，得られるマージンに関してわずかに MEMMA が勝っていることがわかる．また，実行時間に関しては MEMMA の方が圧倒的に良い性能であることもわかる．

4.2 MEMMA の更新回数の実験的評価

次に，MEMMA アルゴリズムの更新回数の実験的な評価をするための実験を行った．定理 3 より更新回数の上限は ε に依存することが分かっているので，ε の値を $\varepsilon = \gamma^*/i$ $(i = 1...30)$ と変化させて，その時の MEMMA の更新回数の変化を調べた．実験データなど，ε の値以外は上の実験 4.1 と同じである．その実験結果が図 4 である．この図は ε の対数を x 軸に，更新回数の対数を y 軸にとったグラフである．図の下部の線が実行結果を表す．上部の線は $\ln(\text{更新回数}) = -3\ln(\varepsilon) + D$($D$ は定数) の直線，中央の線は $\ln(\text{更新回数}) = -2\ln(\varepsilon) + C$($C$ は定数) の直線である．

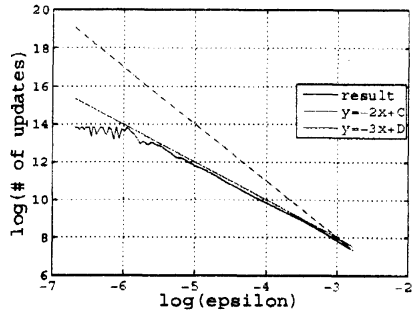

図 4: ε を変化させた際の更新回数の変化．横軸が $\ln\varepsilon$，縦軸が ln(更新回数) である．

図を見てわかるとおり，実行結果は中央の直線，つまり更新回数と ε の関係が $\ln(\text{更新回数}) = -2\ln(\varepsilon) + C$($C$ は定数) のときと非常に似ている．この式の対数を外すと，更新回数 $= \frac{K}{\varepsilon^2}$ (K は定数) となる．つまり，実験的には MEMMA の更新回数は $O\left(\frac{\ln n}{\varepsilon^2}\right)$ 回であるといえる．

5 結論

本論文では，∞-ノルムマージンを最大化するよう更新ができるオンライン学習アルゴリズム MEMMA を提案した．そして，MEMMA の更新回数の理論的上限が $O\left(\frac{\ln n}{\varepsilon^2}\right)$ 回であり，その時に $\gamma^* - \varepsilon$ のマージンを保証することを理論的に示した．さらに，人工データを用いた評価実験において，近似的に ∞-ノルムマージンを最大化するよう更新ができるオンライン学習アルゴリズム PUMMA と比べて，僅かに MEMMA アルゴリズムの方が高性能，実行時間に関してははるかに高性能であることを示した．

また，実験結果から MEMMA アルゴリズムの更新回数の上限は $O\left(\frac{\ln n}{\varepsilon^2}\right)$ 回であると予測されることを確認した．つまり，MEMMA アルゴリズムの更新回数の理論的上限はさらに早くなる可能性を残している．

謝辞

有益な議論をして下さった瀧本英二先生に感謝します．

参考文献

[1] A. Bordes, S. Ertekin, J. Weston, and Léon Bottou. Fast kernel classifiers with online and

active learning. *Journal of Machine Learning Research*, 6:1579–1619, 2005.

[2] D. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[3] C. Gentile. A new approximate maximal margin classification algorithm. *Journal of Machine Learning Research*, 2:213–242, 2001.

[4] C. Gentile. The robustness of the p-norm algorithms. *Machine Learning*, 53(3):265–299, 2003.

[5] A. J. Grove, N. Littlestone, and D. Schuurmans. General convergence results for linear discriminant updates. *Machine Learning*, 43(3):173–210, 2001.

[6] K. Ishibashi, K. Hatano, and M. Takeda. Online Learning of Maximum p-Norm Margin Classifiers with Bias. In *Proceedings of the 21st Annual Conference of Learning Theory*, pages 69–80, 2008.

[7] T. Jaakkola, M. Meila, and T. Jebara. Maximum Entropy Discrimination. In *Advances in Neural Information Processing Systems 12*, 1999.

[8] N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, 1988.

[9] P. M. Long and X. Wu. Mistake bounds for maximum entropy discrimination. In *Advances in Neural Information Processing Systems 17*, pages 833–840, 2004.

[10] O. Mangasarian. Exact 1-norm support vector machines via unconstrained convex differentiable minimization. *Journal of Machine Learning Research*, 7:1517–1530, 2006.

[11] O. Mangasarian and R. Meyer. Nonlinear pertubation of linear programs. *SIAM Journal on Control and Optimization*, 17(6):745–752, 1979.

[12] M. L. Minsky and S. A. Papert. *Perceptrons*. MIT Press, 1969.

[13] M.S. Pinsker. *Information and Information Stability of Random Variables and Processes*. Holden-Day, 1964.

[14] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, B*, 58(1):267–288, 1996.

[15] M. Warmuth, K. Glocer, and S. V. N. Vishwanathan. Entropy regularized lpboost. In *Proceedings of the 19th International Conference on Algorithmic Learning Theory*, pages 256–271, 2008.

[16] T. Zhang. Regularized winnow methods. In *Advances in Neural Information Proccessing Systems 13*, pages 703–309, 2000.