情報論的学習理論テクニカルレポート 2009
Technical Report on Information-Based Induction Sciences 2009 (IBIS2009)

行と列の生成による線形計画ブースティング
Linear Programming Boosting by Column and Row Generation

畑埜　晃平*　　瀧本　英二†
Kohei Hatano　　Eiji Takimoto

Abstract: We propose a new boosting algorithm based on a linear programming formulation. Our algorithm can take advantage of the sparsity of the solution of the underlying optimization problem. In preliminary experiments, our algorithm outperforms a state-of-the-art LP solver and LPBoost especially when the solution is given by a small set of relevant hypotheses and support vectors.

Keywords: ブースティング，ソフトマージン，線形計画法

1　はじめに

近年，機械学習や関連分野において疎な分類器が注目を集めている．例えばテキスト分類においては，特徴のサイズが１０万以上になるがそのうち重要な特徴はほんのごく一部となるとなることがある．このような場合，疎な分類器は分類だけでなく特徴選択にも有効である．

疎な分類器を学習する主なアプローチとして，ℓ_1 ソフトマージン最適化問題として定式化する方法が挙げられる．この問題では，マージンを大きくするような線形分類器を重みの ℓ_1 ノルムを正則化することによって求める．マージンを用いた一般化誤差の研究によって，このアプローチは分類において頑健であることが保証されている [12, 5]．また，近年，ℓ_1 ソフトマージン最適化問題は類似性指標を用いた学習にも応用されている [6, 8, 1, 14].

ℓ_1 ソフトマージン最適化問題は線形計画問題であるため，シンプレックス法や内点法など標準的な最適化手法によって解くことが可能である．しかし，特徴や例のサイズが１万以上の場合，多大な計算時間を要することがある．

LPBoost は Demiriz らによって提案されたよく知られたブースティング手法であり，ℓ_1 ソフトマージン最適化問題を解くように設計されている [5]．LPBoost におけるブースティングの繰り返し回数の上限は知られておらず，また，最悪時の繰り返し回数の下界は他のブースティング手法よりも指数的に悪いことがわかっているものの（後述），実際多くの場合十分に高速である（LPBoost のハードマージン最適化問題に関する初期の結果については [7] が知られている）．

定義域 $\mathcal{X}$ 上 の m 個のラベル付き事例 $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)$ $(\boldsymbol{x}_i \in \mathcal{X}, y_i \in \{-1, +1\})$ と n 個の仮説（特徴）$h_1, \ldots, h_n (h_j : \mathcal{X} \to [-1, +1])$ が与えられたとする．直接ソフトマージン最適化問題を解く代わりに，LPBoost は以下の操作を繰り返す：(1) 各繰り返し t において，事例上の現在の分布 $\boldsymbol{d}_t$ に対して“ よい仮説 ”h_t を見つける．(2) 仮説の集合を $\{h_1, \ldots, h_t\}$ に制限した“ 小さな ”ソフトマージン最適化問題を解くことによって次の分布 $\boldsymbol{d}_{t+1}$ を構成する．最終的な仮説は過去に選ばれた仮説の線形結合である，ただし，各仮説の係数は最後に解いた“ 小さな ”ソフトマージン最適化問題のラグランジュ乗数によって与えられる．ここで，$m \times n$ の行列を考える．行列の各成分は $u_{ij} = y_i h(\boldsymbol{x}_i)$ $(i = 1, \ldots, m, j = 1, \ldots, n)$ とする．ここで，各行が事例に，各列が仮説に対応している．この行列は ℓ_1 ソフトマージン最適化問題の制約行列を表しており，行列の視点から見れば，LPBoost は列を生成しながら線形計画問題を繰り返し解いていることになる．実際，LPBoost は Column Generation [11] と呼ばれる最適化手法における古典的なテクニックを用いた線形計画問題の解法と見なすこともできる．

しかしながら，LPBoost は問題の疎性を十分に活かしきれていない．実際，ℓ_1 ソフトマージン最適化問題には２種類の疎性がある．最初の疎性は仮説間に現れる．

*九州大学大学院システム情報科学研究院情報学部門, 〒 819-0395 福岡市西区元岡 744, tel. 092-802-3787, e-mail hatano@i.kyushu-u.ac.jp,
Department of Informatics, Kyushu University, 744 Motooka, Nishi-ku, Fukuoka, Japan
†九州大学大学院システム情報科学研究院情報学部門, 〒 819-0395 福岡市西区元岡 744, tel. 092-802-3782, e-mail eiji@i.kyushu-u.ac.jp
Department of Informatics, Kyushu University, 744 Motooka, Nishi-ku, Fukuoka, Japan

すなわち，最適解においては，“ 重要な ”仮説のみが非負の係数を持つ．そして，もう 1 つの疎性は事例間に現れる．より正確には，いくつかの“ 重要な ”事例（しばしば“ サポートベクター ”とも呼ばれる）だけが，最適解に影響し, それら以外の事例を取り除いても最適解は変化しない．

本稿では，我々は仮説間と事例間の両方の疎性を利用した新しいブースティング手法を提案する．本提案手法 Sparse LPBoost は行と列の両方を生成するアプローチを取る．Sparse LPBoost は“ 重要そうな ”列（仮説）と行（事例）を選択しながら線形計画問題を繰り返し解く．我々は，精度パラメータ $\varepsilon > 0$ が与えられたとき，Sparse LPBoost はソフトマージンが $\gamma^* - \varepsilon$ 以上であるような仮説の線形結合を出力することを示す，ここで，γ^* は最適なソフトマージンである．

本稿では，人工データおよび実データにおける準備的な実験において，Sparse LPBoost が，従来手法である線形計画ソルバや LPBoost よりも ℓ_1 ソフトマージン最適化問題をより高速に解くことを示す．特に，事例や仮説の総数が 1 万以上であるような大規模なデータにおいて，Sparse LPBoost は他の手法に比べて数倍以上の高速化を達成した．

いくつかの関連研究を挙げておく．Warmuth らは Entropy Regularized LPBoost [16] と呼ばれる LPBoost の改良版を提案している．Entropy Regularized LPBoost も ℓ_1 ソフトマージン最適化問題を近似的に解くことが可能であり，繰返し回数 $O(\log(m/\nu)/\varepsilon^2)$ で実行できることが示されている．一方，LPBoost の繰り返し回数の最悪時の下界は $\Omega(m)$ であることも知られている [15]．この手法も LPBoost と同様，仮説間の疎性を利用している．

Mangasarian [10] や Sra [13] の手法は元の線形計画問題を目的関数に 2 次の項を加えることにより，2 次計画問題に変換しそれぞれニュートン法や Bregman 法 [3] を用いて解を求める．彼らの手法は仮説間，事例間のいずれの疎性も利用していない．

Bradley と Mangasarian [2] は，本手法と同様に，元の線形計画問題をより小さな問題に分解して解く手法を提案している．しかしながら，この手法は LPBoost と同様に行（仮説）だけを生成する．

以上，いずれの既存手法も，本手法のように事例間, 仮説間両方の疎性を利用しているわけではない．

2 準備

事例の空間を $\mathcal{X}$ とする．与えられた事例の集合を $S = ((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m))$ とする，ここで，各事例 $\boldsymbol{x}$ は $\mathcal{X}$ に属し，各ラベル y_i は -1 または $+1$ の値をとるとする $(i = 1, \ldots, m)$．仮説の集合を $\mathcal{H}$ とする，ただし，各仮説 $h \in \mathcal{H}$ は事例の空間 $\mathcal{X}$ から $[-1, +1]$ への関数とする．自然数 k に対して，$\mathcal{P}^k$ を k 次元の確率分布の集合，つまり，$\mathcal{P}^k = \{\boldsymbol{p} \in [0,1]^k : \ \sum_{i=1}^k p_i = 1\}$ とおく．仮説の（正規化された）重み付け $\boldsymbol{\alpha} \in \mathcal{P}^n$ に対するラベル付き事例 $(\boldsymbol{x}, y)$ のマージンを $y_i \sum_{j=1}^n \alpha_j h_j(\boldsymbol{x}_i)$ と定義する．事例の集合上の分布 $\boldsymbol{d} \in \mathcal{P}^m$ に対する仮説 $h \in \mathcal{H}$ のエッジを $\sum_{i=1}^m y_i d_i h(\boldsymbol{x}_i)$ と定義し，$\gamma_{\boldsymbol{d}}(h)$ と表記する．

2.1 ℓ_1 ソフトマージン最適化問題

ℓ_1 ソフトマージン最適化問題は以下のように定式化される（例えば，[5, 16] など参照）．

$$\max_{\rho, \boldsymbol{\alpha}, \boldsymbol{\xi}} \rho - \frac{1}{\nu} \sum_{i=1}^m \xi_i \qquad (1)$$

$$\text{sub.to}$$

$$y_i \sum_j \alpha_j h_j(\boldsymbol{x}_i) \geq \rho - \xi_i \ (i = 1, \ldots, m),$$

$$\boldsymbol{\alpha} \in \mathcal{P}^n, \ \boldsymbol{\xi} \geq \mathbf{0},$$

$$\min_{\gamma, \boldsymbol{d}} \gamma \qquad (2)$$

$$\text{sub.to}$$

$$\gamma_{\boldsymbol{d}}(h_j) = \sum_i d_i y_i h_j(\boldsymbol{x}_i) \leq \gamma \ (j = 1, \ldots, n),$$

$$\boldsymbol{d} \leq \frac{1}{\nu}\mathbf{1}, \ \boldsymbol{d} \in \mathcal{P}^m,$$

ただし主問題は (1) ，双対問題は (2) でそれぞれ表される．これらの問題は線形計画問題であり，主問題 (1) の最適解を $(\rho^*, \boldsymbol{\alpha}^*, \boldsymbol{\xi}^*)$，双対問題 (2) の最適解を $(\gamma^*, \boldsymbol{d}^*)$ とすると，双対性により，$\rho^* - \frac{1}{\nu}\sum_{i=1}^m \xi_i^* = \gamma^*$ となることが知られている．また，本稿ではそれぞれの目的関数項をソフトマージンと呼ぶことにする．

注目すべき最適解の性質は疎ということである．KKT 条件により，最適解は次のような性質を満たす．

$$d_i^* \left(y_i \sum_j \alpha_j^* h_j(\boldsymbol{x}_i) - \rho^* + \xi_i^* \right) = 0 \quad (i = 1, \ldots, m).$$

$$d_i^* \geq 0, \ y_i \sum_j \alpha_j^* h_j(\boldsymbol{x}_i) - \rho^* + \xi_i^* \geq 0 \quad (i = 1, \ldots, m).$$

$$\xi_i^* \left(1/\nu - d_i^*\right) = 0, \ \ \xi_i^* \geq 0, \ d_i^* \leq 1/\nu \quad (i = 1, \ldots, m).$$

この性質から，以下が成り立つ．

- $y_i \sum_j \alpha_j^* h_j(\boldsymbol{x}_i) > \rho^*$ ならば，$d_i^* = 0$.
- $0 < d_i^* < 1/\nu$ n ならば，$y_i \sum_j \alpha_j^* h_j(\boldsymbol{x}_i) = \rho^*$.
- $\xi_i^* > 0$ ならば，$d_i^* = 1/\nu$.

特に，ラベル付き事例 $(\boldsymbol{x}_i, y_i)$ の対応する重み $\boldsymbol{d}_i$ が非ゼロであるとき，$(\boldsymbol{x}_i, y_i)$ をサポートベクターと呼ぶ．ここで，マージンが ρ^* 未満（つまり $\xi_i^* > 0$）であるようなラベル付き事例は高々 ν 個であることに注意してほしい．なぜなら，そうでない場合，$\sum_i d_i^* > 1$ となり矛盾するからである．

さらに，主問題の最適解も疎な性質をもつ．

- $\gamma_{\boldsymbol{d}^*}(h_j) < \gamma^*$ ならば，$\alpha_j^* = 0$.

対応する係数 α_j が正であるとき，仮説 h_j が重要であると言うことにする．以上から，最適解はサポートベクターと重要な仮説のみを用いて再現することができることがわかる．

3 アルゴリズム

本節では，双対問題 (2) を解く手法について詳しく述べる．

3.1 LPBoost

まず，LPBoost [5] について述べる．事例の集合上の初期分布 $\boldsymbol{d}_1$（一様分布とする）が与えられた時，LPBoost は以下の動作を繰り返す．各ステップ t において，(i) LPBoost は分布 $\boldsymbol{d}_t$ に対して $\gamma_t + \varepsilon$ 以上の エッジ を持つような仮説 h_t を選ぶ．(ii) 次に，新しい制約 $\gamma_{\boldsymbol{d}}(h_t) \leq \gamma$ を現在の最適化問題に追加して解く．動作 (i) において条件を満たす仮説が存在しなければ終了する．図 1 に詳細をまとめる．

次に LPBoost が双対問題 (2) を近似的に解くことを示す．

定理 1. LPBoost はソフトマージン $\gamma^* - \varepsilon$ 以上の最終的な仮説を出力する．

証明. アルゴリズムの定義から，LPBoost が最終的な仮説を出力するとき，$\gamma_T \geq \max_{h \in \mathcal{H}} \gamma_{\boldsymbol{d}_T}(h) - \varepsilon$ が成り立つ．さらに，$\boldsymbol{d}_t$ は双対問題 (2) の実行可能解なので，$\max_{h \in \mathcal{H}} \gamma_{\boldsymbol{d}_T}(h) \geq \gamma^*$ が成り立つ．これらの事実を組み合わせると，$\gamma_T \geq \gamma^* - \varepsilon$ を得る． □

Algorithm 1 LPBoost(S,ε)

1. $\boldsymbol{d}_1$ を事例の集合 S 上の一様分布とする．$\gamma_1 = -1$ とする．
2. For $t = 1, \ldots,$
   (a) 分布 $\boldsymbol{d}_t$ に対して $\gamma_t + \varepsilon$ の エッジ を持つような仮説 h_t を選ぶ．
   (b) 条件を満たす仮説が $\mathcal{H}$ 中に存在しなければ $T = t - 1$ とし，終了．
   (c) 双対問題 (2) を限定された仮説集合 $\{h_1, \ldots, h_t\}$ に対して解く．得られた最適解を $(\gamma_{t+1}, \boldsymbol{d}_{t+1})$ とおく．

$$
\begin{aligned}
(\gamma_{t+1}, \boldsymbol{d}_{t+1}) = \arg \min_{\gamma, \boldsymbol{d} \in \mathcal{P}^m} \gamma \\
\text{sub. to} \\
\gamma_{\boldsymbol{d}}(h_j) \leq \gamma \quad (j = 1, \ldots, t) \\
\boldsymbol{d} \leq \frac{1}{\nu} \mathbf{1}.
\end{aligned}
$$

3. 最終的な仮説 $f(\boldsymbol{x}) = \sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x})$ を出力する，ここで，各 α_t $(t = 1, \ldots, T)$ は双対問題 (2) における Lagrange 乗数である．

3.2 提案手法

本節では，本提案手法 Sparse LPBoost について述べる．本手法は LPBoost に変更を加えたものであり，大まかに 2 つの違いがある．1 つ目の違いは，分布 $\boldsymbol{d}_t$ の定義域が事例の集合全てではなく，現在の仮説の重み付け $\boldsymbol{\alpha}_t$ に対するマージンが小さい事例のみをカバーする点である．2 つ目の違いは，Sparse LPBoost は各繰り返しにおいて 2 つ以上の仮説を選ぶ点である．Sparse LPBoost の詳細を図 2 に述べる．

次に，Sparse LPBoost の正当性を示す．

定理 2. Sparse LPBoost はソフトマージン $\gamma^* - \varepsilon$ 以上の最終的な仮説を出力する．

証明. $C = S - S_T$ とおく．また，$\gamma_T, \boldsymbol{d}_T$ をアルゴリズム終了時の双対問題の解とする．アルゴリズムの終了条件より，$\gamma_T + \varepsilon$ 以上の エッジをもつ仮説は存在しないので，$\gamma_T + \varepsilon \geq \max_{h \in \mathcal{H}} \gamma_{\boldsymbol{d}_T}(h)$ が成り立つ．さらに，$\boldsymbol{d}_T \in \mathcal{P}^{|S_T|}$ は事例の集合 S_T に対する双対問題 (2) の実行可能解であるので，$\max_{h \in \mathcal{H}} \gamma_{\boldsymbol{d}_T}(h) \geq \gamma^*$ を満たす．したがって，$\gamma_T \geq \gamma^* - \varepsilon$ が成り立つ．最後に，分布 $\boldsymbol{d}'_T = (\boldsymbol{d}_T, 0, \ldots, 0) \in \mathcal{P}^{|S|}$ を考える．この分布は C に属する事例にゼロの重みを与えるような分布である．このとき，明らかに $(\gamma_T, \boldsymbol{d}'_T)$ は，事例の集合 S に対し

Algorithm 2 Sparse LPBoost(S,ε)

1. (初期化) ν 個の事例を任意に選び，それらからなる事例の集合を S_1，S_1 上の一様分布を d_1 とする．また，$f_1(\boldsymbol{x}) = 0$，$\rho_1 = 1$，$\gamma_1 = -1$, $H_1 = \emptyset$ とする．
2. For $t = 1, \ldots,$
   (a) f_t に対してマージンが ρ_t 未満の事例からなる集合 S'_t を１つ選ぶ.
   (b) 条件を満たす S'_t が存在しなければ，$T = t-1$ とし，break.
   (c) $S_{t+1} = S_t \cup S'_t$ とおく.
   (d) For $t' = 1, \ldots,$
      i. 分布 $\boldsymbol{d}_{t'}$ に対して エッジが　$\gamma_{t'} + \varepsilon$ であるような仮説からなる集合 $H'_{t'}$ を１つ選ぶ.
      ii. 条件をみたす $H'_{t'}$ が存在しなければ break.
      iii. $H_{t'+1} = H_{t'} \cup H'_{t'}$ とおく.
      iv. 双対問題 (2) を S_t および　$H_{t'+1}$ に対して解く．$(\gamma_{t'+1}, \boldsymbol{d}_{t'+1})$ を得られた解で更新する．また，$f_{t+1}(\boldsymbol{x}) = \sum_{h \in H_{t'+1}} \alpha_h h(\boldsymbol{x})$，$\rho_{t+1}$ を対応する主問題の解とする，ここで，各 α_h は双対問題 (2) の Lagrange 乗数である.
3. $f_T(\boldsymbol{x}) = \sum_{h \in H_T} \alpha_h h(\boldsymbol{x})$　を出力.

て KKT 条件を満たす. □

3.3 仮説と事例を選択するためのヒューリスティクス

前に述べた Sparse LPBoost の説明では，仮説と事例の選択方法については指定していない．本節では，仮説と事例の選択に関するヒューリスティクスについて述べる．具体的には以下の３つの選択方法を考える.

閾値選択 $\gamma'_t + \varepsilon$ 以上の エッジを持つ仮説，および，マージンが ρ_t 以下の事例をそれぞれ１つずつ選択.

最大 / 最小選択　最大のエッジを持つ仮説，および，最小のマージンを持つ事例をそれぞれ１つ選択.

最大 / 最小- 指数選択 分布 $\boldsymbol{d}_{t'}$ に対して $\gamma'_t + \varepsilon$ 以上のエッジを持つ仮説の集合を $\widehat{\mathcal{H}}_{t'}$，$f_t$ に対してマージンが ρ_t 以下となるような事例の集合を $\widehat{S}_t$ とおく．このとき，エッジの高い順に上位 K 個の仮説，マージンの低い順に上位 L 個の事例をそれぞれ選ぶ，ここで　$K = \min\{|\widehat{\mathcal{H}}_{t'}|, 2^{t'}\}$，$L = \min\{|\widehat{S}_t|, 2^t\}$ とする.

では，どの選択方法をとるべきか議論する．以下では，例として，$\nu = 0.2m$ の場合を考えよう．この場合，ラベル付き事例の高々２割がマージン ρ^* 以下となり，よってこれらは正しく分類されないかもしれない．しかしながら，この設定により，高々２割の事例の誤分類を許すことで，全体のマージンを向上させることができる．よってラベルのノイズの大きいデータには妥当な設定といってよいだろう．また，簡単のため，事例に関する計算時間のみ考える．線形計画法の計算時間は m^k (k は定数) とする.

もし，閾値選択もしくは最大 / 最小選択を用いた場合，Sparse LPBoost の計算時間は少なくとも

$$\sum_{t=1}^{\nu} t^k > \int_{t=1}^{\nu} t^k dt = \frac{\nu^{k+1} - 1}{k+1} = \Omega(m^{k+1})$$

要する.

一方，最大 / 最小-指数選択を用いて，Sparse LPBoost が cm 個 $(0 < c \leq 1)$ の事例を選んで終了した場合，計算時間は高々,

$$\begin{aligned}
\sum_{t=1}^{\lceil \log(cm) \rceil} (\nu + 2^t)^k &= \sum_{t=1}^{\lceil \log(cm) \rceil} \sum_{s=0}^{k} \binom{k}{s} \nu^s 2^{t(k-s)} \\
&= \sum_{s=0}^{k} \binom{k}{s} \nu^s \sum_{t=1}^{\lceil \log(cm) \rceil} 2^{t(k-s)} \\
&\leq \sum_{s=0}^{k} \binom{k}{s} \nu^s 2^{(k-s)(\log(cm)+2)} \\
&= \sum_{s=0}^{k} \binom{k}{s} \nu^s (c'm)^{k-s} \\
&= (\nu + c'm)^k = (0.2 + c')^k m^k \\
&= O(m^k).
\end{aligned}$$

となる．また，同様の議論が仮説の選択方法に関しても成り立つ.

したがって，上記の３つの選択方法のうち，最大 / 最小-指数選択が，計算時間の観点で頑健であるといえる．以降では， Sparse LPBoost は最大 / 最小-指数選択を用いると仮定する．注意すべき点は， Sparse LPBoost の利点は，元の線形計画法とは最悪時の計算時間は同じでも，より小さい係数をもつことである．定数倍の計算時間ではあっても，実際のデータにおいては大きな実行時間の短縮が図れる可能性がある.

4 実験

人工データ, 実データ上で線形計画法, LPBoost, および Sparse LPBoost の実験的評価を行う．実験は Xeon 3.8GHz の CPU，メモリ 8Gb を持つワークステーション上で行った．実装には C++, 線形計画ソルバには CPLEX 11.0 を用いた．

4.1 人工データ

本実験で使用する人工データセットは $m = 10^4$ から 10^6 個の $\mathcal{X} = \{-1, +1\}^n$ 上のラベル付き事例から構成される．我々は線形関数 $f(\boldsymbol{x}) = x_1 + x_2 + \cdots + x_k + b$, を 1 つ固定する，ここで，$x_1, \ldots, x_k$ は事例 $\boldsymbol{x}$ の最初の k 個の次元であり，b は定数項である．関数 f は各事例 $\boldsymbol{x}$ のラベル (-1 or $+1$) を $f(\boldsymbol{x})$ の符号の正負によって与える．各パラメータは $n = 100$, $k = 10$ and $b = 5$ とした．各データセットに対して正例と負例が等確率になるようにランダムに事例を生成した．さらに，データにノイズを加えるため，0% または 5% の確率でラベルをランダムに反転させた．

各データセットに対して，$n+1$ 個の仮説を用意した．最初の n 個の仮説は n 個の次元それぞれに対応する，つまり，$h_j(\boldsymbol{x}) = x_j$ $(j = 1, \ldots, n)$ である．最後の仮説は常に $+1$ を返す（定数項に対応）．

また，ノイズなし，ありのそれぞれのデータセットに対して，パラメータ ν を $\nu = 1$ and $\nu = 0.2m$ に設定した．LPBoost and Sparse LPBoost に対しては $\varepsilon = 0.001$ とした．

ノイズなし，ありのデータセットに対する実験結果を表 1 にまとめる．ここで，計算時間は CPU 時間である．注目すべき点は，他のアルゴリズムに比べて，Sparse LPBoost は ν の選択に対してより頑健であることである．実際，異なる ν の値に対しても，計算時間は最速ないし，最速に近い．さらに，ノイズなし，ありのそれぞれのデータセットに対して，Sparse LPBoost は全事例数 m より少ない事例を選択している．したがって，双対問題 (2) における変数のサイズが減少し，問題を小さくすることにより計算時間の高速化が達成されている．特に，事例数が $m = 10^6$ の場合，LP ソルバーや LPBoost がメモリ不足で動作しないのに対して，Sparse LPBoost は動作しており, よりメモリを消費しないことがわかる．また，$\nu = 1$ と設定したノイズなしデータセットに対して，得られた分布 $\boldsymbol{d}$ の非ゼロ要素数は小さくなっている一方，$\nu = 0.2m$ としたノイズありデータセットに対して，得られた分布 $\boldsymbol{d}$ の非ゼロ要素数は増加している．結果として計算時間も増大している．これは，最適な分布が少なくとも，ν 個の非ゼロ要素を持つためである．

4.2 実データ

次に，実データセット上での実験結果を示す．実データセットには Reuters-21578 [1], RCV1 [9], news20 を用いた．

Reuters-21578 に対しては，部分データセット modified Apte (“ModApte”) を用いた．このデータセットは，トピックによってラベルづけされた 10170 個の事例からなる．我々は，“acq” という主要なトピックを正例，他のトピックを負例とし，2 値分類問題を構成した．仮説として，30,839 の単語に対応した決定株を用意した．ここで，各決定株は 1 つの単語に対応し，与えられた文書にその単語が含まれていれば $+1$，そうでなければ -1 を返す．

RCV1 データセット，news20 には，LIBSVM tools [4] で提供されているものを用いた．データ数と仮説の総数はそれぞれ，$m = 20,242$, $n = 47,236$, $m = 19,996$, $n = 1,355,193$ である．

各データセットに対して，さらに定数項に対応する仮説 -1 を用いる．LPBoost and Sparse LPBoost のパラメータについては，$\varepsilon = 10^{-4}$ とし，Reuters-21578, RCV1, news20 データセットそれぞれに対して $\nu = 0.2m$ とした．

実験結果を表 2 にまとめる．Sparse LPBoost は他の手法よりも数倍以上高速に動作していることがわかる．人工データでの結果と同様，Sparse LPBoost はより少ない事例（約 $0.6m$ から $0.8m$ 程度）で最適解を近似的に求めている．また，Sparse LPBoost は仮説間の疎性も利用しているように見える．実際，両データセットにおいて，Sparse LPBoost は 30,000 以上の仮説の中から数パーセントの仮説のみを用いている．

5 結論

本稿では，与えられた任意の $\varepsilon > 0$ に対して，ℓ_1 ソフトマージン最適化問題の ε 近似を求める手法を提案した．本手法は最適解における仮説間と事例間の疎性を利用することにより，標準的な線形計画ソルバや LPBoost よりも高速に動作する．

今後の課題として，計算時間の理論的保証が得られるように Sparse LPBoost を改良することが挙げられる．また，実用的な観点では，仮説と事例を選択するためのより良いヒューリスティクスの開発も重要である．

[1] http://www.daviddlewis.com/resources/testcollections/reuters21578.

		$\nu = 0$ (ノイズなし)			$\nu = 0.2m$ (ノイズあり)		
m	手法	計算時間 (秒)	$\#(d_i > 0)(\%)$	$\#(\alpha_j > 0)(\%)$	計算時間 (秒)	$\#(d_i > 0)(\%)$	$\#(\alpha_j > 0)$ (%)
10^4	LP	**5.99**	0.25	18.8	**2.8**	20.0	9.9
	LPB	11.64	0.19	17.8(69.3)	3.16	20.0	9.9(9.9)
	SLPB	6.15	0.25(23.0)	25(98)	2.99	20.0(47)	9.9(10.9)
10^5	LP	82.4	0.1	53.5	**33.78**	20	9.9
	LPB	120	0.02	15.8(61.4)	43.9	20	9.9(10.9)
	SLPB	**63.9**	0.03(18.9)	20.8(80)	53.1	20(58.7)	10.9(59.4)
10^6	LP	メモリ不足	n/a	n/a	メモリ不足	n/a	n/a
	LPB	メモリ不足	n/a	n/a	メモリ不足	n/a	n/a
	SLPB	**684**	0.002(25.2)	15.8(71.3)	**1679**	20(46.5)	90(91)

表 1: ノイズなし（$\nu = 1$），ノイズあり（$\nu = 0.2m$) データに対する実験結果．$\#(d_i > 0)$，および $\#(\alpha_j > 0)$ はそれぞれ，$\boldsymbol{d}$, $\boldsymbol{\alpha}$ における非ゼロ要素数の割合を表す．LPBoost (LPB) と Sparse LPBoost (SLPB) に対しては，選択された仮説や事例の個数の割合をかっこ内に示す．

データ	手法	計算時間（秒）	$\#(d_i > 0)(\%)$	$\#(\alpha_j > 0)(\%)$
Reuters-21578	LP	21.0	22.1	1.55
(m=10,170,n=30,839)	LPB	52.5	22.1	1.53(1.67)
	SLPB	**8.5**	22.2(68)	1.52(1.96)
RCV1	LP	2135	25.4	5.1
(m=20,242,n=47,237)	LPB	4154	24.3	3.7(4.0)
	SLPB	**690**	24.6(63.4)	3.9(4.6)
news20	LP	121525	25.4	0.16
(m=19,996,n=1,355,193)	LPB	7251	22.7	0.080(0.090)
	SLPB	**1223**	23.1(68.1)	0.088(0.117)

表 2: 実データセットに対する実験結果．$\#(d_i > 0)$，および $\#(\alpha_j > 0)$ はそれぞれ，$\boldsymbol{d}$, $\boldsymbol{\alpha}$ における非ゼロ要素数の割合を表す．LPBoost (LPB) と Sparse LPBoost (SLPB) に対しては，選択された仮説や事例の個数をかっこ内に示す．

謝辞

本研究は科研費若手研究 (B) 21700171 の援助によってなされた．また，本稿に有益なコメントを下さった豊橋技術科学大学の山本悠二氏に感謝します．

参考文献

[1] N. Balcan, A. Blum, and N. Srebro. A theory of learning with similarity functions. *Machine Learning*, 72(1-2):89–112, 2008.

[2] P. S. Bradley and O. L.Mangasarian. Massive data discrimination via linear support vector machines. *Optimization Methods and Software*, 13(1):1–10, 2000.

[3] Y. Censor and S. A. Zenios. *Parallel Optimization: Theory, Algorithms, and Applications*. Oxford University Press, 1998.

[4] C. C. Chang and C. J. Lin. Libsvm: a library for support vector machines. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`, 2001.

[5] A. Demiriz, K. P. Bennett, and J. Shawe-Taylor. Linear programming boosting via column generation. *Machine Learning*, 46(1-3):225–254, 2002.

[6] T. Graepel, R. Herbrich, B. Schölkopf, A. Smola, P. Bartlett, K. Müller, K. Obermayer, and R. Williamson. Classification on proximity data

with LP-machines. In *International Conference on Artificial Neural Networks*, pages 304–309, 1999.

[7] A. J. Grove and D. Schuurmans. Boosting in the limit: Maximizing the margin of learned ensembles. In *Proceedings of the fifteenth National Conference on Artificial Intelligence (AAAI-98)*, pages 692–698, 1998.

[8] M. Hein, O. Bousquet, and B. Schölkopf. Maximal margin classification for metric spaces. *Journal of Computer ans System Sciences*, 71:333–359, 2005.

[9] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 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] S. Nash and A. Sofer. *Linear and Nonlinear Programming.* Macgraw-Hill, 1996.

[12] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee. Boosting the margin: a new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651–1686, 1998.

[13] S. Sra. Efficient large scale linear progamming support vector machines. In *Machine Learning: ECML 2006*, pages 767–774, 2006.

[14] L. Wang, M. Sugiyama, C. Yang, K. Hatano, and J. Fung. Theory and algorithms for learning with dissimilarity functions. *Neural Computation*, 21(5):1459–1484, 2009.

[15] M. Warmuth, K. Glocer, and G. Rätsch. Boosting algorithms for maximizing the soft margin. In *Advances in Neural Information Processing Systems 20*, pages 1585–1592, 2008.

[16] 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.