順序決定木に対する正則化パラメータ推定の高速化

Efficient Algorithms for Estimating Regularization Parameters of Ordered Decision Trees

金森憲太朗 [1*]　石畠正和 [2]　湊真一 [2]　有村博紀 [2]
Kentaro Kanamori[1], Masakazu Ishihata[2], Shin-ichi Minato[2], Hiroki Arimura[2]

[1] 北海道大学 工学部
[1] School of Engineering, Hokkaido University
[2] 北海道大学 大学院情報科学研究科
[2] Graduate School of Inf. Sci. & Tech., Hokaido University

Abstract: In this paper, we study the regularized empirical risk minimization for the class of oblivious decision trees, and present an efficient exact algorithm for finding an optimal regularization parameter from given data. Based on geometric computation on stamp points, our algorithm computes optimal answers from the output of ODT algorithm (Osabe *et al.*, 2016). By experiments, our algorithm was faster than the conventional grid search in order of magnitude.

1　はじめに

教師あり学習のさまざまな局面で，データを生成する未知の確率分布 $\mathcal{D}$ に対して，与えられた学習アルゴリズム[1]の良さを，$\mathcal{D}$ から得られる有限なデータ集合 S から推測することが重要となる．

正則化経験損失最小化: 一般に，仮説の複雑さが増すほど，分布 $\mathcal{D}$ から観測された訓練データ集合 S に対する予測精度は向上するが，一方で，同一の分布 $\mathcal{D}$ に従う S と独立な未知データ集合 T（以後，テストデータ集合と呼ぶ）に対する予測精度が悪化する過剰学習（overfitting）が起きやすいことが知られている [6,8]．

これを防ぐための手法の一つが，学習において S に対する予測誤差に加え，仮説の複雑さの和を考慮して仮説を選択する**正則化経験損失最小化**（regularized empirical risk minimization．以後，RegERM と略す）を用いることである．この RegERM は，仮説空間 $\mathcal{H}$ と，訓練データ集合 S，非負実数パラメータ $\lambda \geq 0$ に対して，目的関数

$$obj(h \mid S, \lambda) \quad := \quad err(h \mid S) + \lambda \cdot size(h) \qquad (1)$$

を最小にする仮説を求める方法である．ここに，仮説 h に対して，$err(h \mid S)$ はその経験損失（経験誤差）であり，$size(h)$ は，仮説の複雑さである．本稿の決定木の族の場合，$size(h)$ は決定木の葉数である．

研究目的: 正則化パラメータの厳密推定: 式 (1) 中の実数パラメータ λ は学習アルゴリズムでは決定されない**超パラメータ**であるので，実際の機械学習データ応用においては，学習に先立って適切な λ の値を推定することが重要である．このために，アルゴリズムの期待予測誤差の推定値（ここでは，CV 誤差を用いる）を最小化するような λ の値を求める**交差検定誤差に関する最適正則化パラメータ推定**（以後，RegPE-CV と略す）が広く用いられている．

従来手法の問題点: しかし，RegPE-CV 問題において，λ の候補は無限に存在するため，最適解を厳密に求めるのは難しい．そこで，通常は，超パラメータの値をある刻み幅 δ ずつ増加させながら，最小値を探索する 1 次元のグリッドサーチ（grid search）が用いられることが多い．しかし，グリッドサーチは，厳密解が求まる保証がないことや，解像度 $1/\delta$ に比例して上限なく計算時間が増加するなどの問題をもつ．そこで，本研究では，正則化経験損失最小化における超パラメータ推定問題の高速な厳密解法について考察する．

研究結果: 本稿では，順序決定木（ordered/oblivious decision trees, ODT）の族に対して，長部ら [1,10] によって提案された経験損失の厳密最適化を行う学習アルゴリズム ODT を用いた RegPE-CV の厳密解法を考察する．

初めに 2 節で，本稿で必要な基本的な記法と定義を準備する．次に 3 節では，一般の仮説の族に対して，幾何学的な解釈から RegPE-CV 問題の解法を議論する．仮説のサイズ k と経験損失 e を x-軸と y-軸にも

*連絡先：北海道大学工学部情報エレクトロニクス学科
〒 060-0814 札幌市北区北 14 条西 9 丁目
E-mail: k11.silentliver@gmail.com

[1]本稿で学習アルゴリズムというとき，アルゴリズム自体に加えて，に仮説空間 $\mathcal{H}$ とパラメータの選択を含める．

つ二次元平面上で，仮説空間 $\mathcal{H}$ が定めるスタンプ集合 $Stamp(\mathcal{H})$ を入力とし，それがサイズ毎の最小経験損失解を含むとき，入力点集合の凸包計算を用いて最適な正則化パラメータを推定するアルゴリズム ConvexSearch を与える．定理 5 で正当性と計算量を与える．

次に順序決定木の族 $\mathcal{H}_{ODT}$ に対して，列挙型の決定木の厳密学習アルゴリズムである ODT アルゴリズム [10] を用いて，指定されたサイズ 1 から最大サイズ K までの最適決定木を含むスタンプ点集合を計算できることから，ConvexSearch を組み合わせて，次の主結果が得られる．

定理 1 与えられた変数順序 $(\mathcal{V}, <)$，最大サイズ K，葉の最小支持度 σ の制約のもとで，V-次の交差検定誤差に関する最適な正則化パラメータ $\hat{\lambda}$ の推定を，$O(\mathcal{T}_{ODT} + KV^2)$ 時間で効率よく実行するアルゴリズムが存在する．

ここに，$T_{ODT} = O(poly(||S||, K, V)\mathcal{F}_{S,\sigma})$ は，与えられた入力データ集合 S とパラメータ $|V|, K, \sigma$ から，ODT アルゴリズムが最適解を求めるのに必要な計算時間であり，$|\mathcal{F}_{S,\sigma}|$ は，S 上の頻出アイテム集合マイニング (FIM) における解集合のサイズであり，一般に入力サイズの指数である．ただし，多くのデータセットと適度な範囲のパラメータで FIM は実行可能であり，広く利用されている [3].

5 節では，3 節の提案手法についてグリッドサーチと比較する計算機実験を行った．いくつかの実データ集合上で，提案アルゴリズムが既存手法よりも高い精度の厳密解を求めることができることと，同時に計算時間も短縮できることを観察した．

2 準備

整数 $i \leq j$ に対して，$[i..j]$ で**整数の連続区間** $\{k \mid i \leq k \leq j\}$ を表す．確率変数 x が確率分布 $\mathcal{D}$ にしたがうことを $x \sim D$ と書き，関数 $f(x)$ の**期待値**を $\mathbb{E}_{x\sim\mathcal{D}}[f(x)] \in \mathbb{R}$ で表す．以下で触れない基本的な定義については，関連する教科書（ [6,8] 等）を参照されたい．

2.1 教師付き学習と誤差

$\mathcal{X}$ と $\mathcal{Y}$ を，それぞれ，**入力データ**と**出力ラベル**の集合とし，組 $(x, y) \in \mathcal{X} \times \mathcal{Y}$ を**ラベル付きデータ**，または**訓練データ**とよぶ．$\mathcal{X} \times \mathcal{Y}$ 上の**分類関数**とは，任意の二値関数 $h : \mathcal{X} \to \mathcal{Y}$ をいい，その族 $\mathcal{H}$ を**仮説空間**とよぶ．

与えられた**データ集合** $S \subseteq \mathcal{X} \times \mathcal{Y}$ に対して，**学習アルゴリズム** $\mathbf{Alg}$ は，$\mathcal{H}$ の仮説 $\hat{h} := \mathbf{Alg}(S \mid \mathcal{H})$ を返す．S に対する仮説 h の**経験誤差**（empirical error. 以後，**経験損失**とも呼ぶ）は，

$$err(h \mid S) := \frac{1}{|S|} \sum_{(x,y)\in S} I[y \neq h(x)]$$

で与えられる．ここに $\sharp err(h \mid S) := \sum_{(x,y)\in S} I[y \neq h(x)]$ は h の**誤分類数**と呼ばれる．

1 節の冒頭の式 (1) に示した正則化経験損失 $obj(h \mid S, \lambda) = err(h \mid S) + \lambda \cdot size(h)$ に対して，**正則化経験損失最小化問題**は次の問題である．

定義 1 正則化経験損失最小化問題 *(*REGERM*): 与えられた仮説空間* $\mathcal{H}$*, データ集合* S*，正実数* $\lambda \geq 0$ *に対して，正則化経験損失関数* $obj(h \mid S, \lambda)$ *を最小化する仮説* $\hat{h} \in \mathcal{H}$ *を求めよ．*

上の定義より，正則化経験損失最小化を行う学習アルゴリズムの解は，$\hat{h}^{\lambda}_{S,\mathcal{H}} := \hat{h} = \arg\min_{h\in\mathcal{H}} obj(h \mid S, \lambda)$ と書ける．

2.2 期待予測誤差と交差検定

期待予測誤差は，例を生成する未知の分布 $\mathcal{D}$ に対する学習アルゴリズムの予測精度を推定するための基本的な尺度である．まず，与えられた仮説 $\hat{h}$ の $\mathcal{H}$ に関する**期待予測誤差**は，$\mathbb{E}_{(x,y)\sim\mathcal{D}}[1[y \neq \hat{h}(x)]]$ と書ける．これより先の経験誤差 $err(\cdot \mid \cdot)$ を用いて，任意の仮説空間 $\mathcal{H}$ に関する任意のアルゴリズム $\mathbf{Alg}$ の**期待予測誤差**（expected prediction error）を，

$$\begin{aligned} EPE(\mathbf{Alg} \mid \mathcal{D}) &= EPE(\mathbf{Alg} \mid \mathcal{D}, \mathcal{H}) \\ &:= \mathbb{E}_{S,T\sim\mathcal{D}}[err(\hat{h} \mid T) \mid \hat{h} = \mathbf{Alg}(S)] \end{aligned} \tag{2}$$

と定める．

機械学習アルゴリズムの設計において，期待予測誤差の計算は重要であるが，一方でデータとラベルの生成分布 $\mathcal{D}$ は未知であるため，一般に直接の計算は困難である．これに対して，**交差検定**（cross validation） [6] は，期待予測誤差の推定に広く用いられている方法の一つである．

本稿では，超パラメータ λ のもとで REGERM を解く学習アルゴリズムを仮定するので，一般性を失うことなく，以後 $\mathbf{Alg}[\lambda]$ と表記する．$V \geq 2$ を任意の正整数[2]とする．観測データ集合 S を互いに排反な V 個のブロック $S_1, \ldots, S_V$ に分割し，第 $1 \leq n \leq N$ 番目のデータが所属するブロックの添え字を $v(n) \in [1..V]$ とする．各 $1 \leq v \leq V$ に対して，第 i 訓練データ集合を $R_v := \bigcup_{w=1,w\neq v}^{V} S_w$ と，第 i 仮説を REGERM により

$$\hat{h}_v := \hat{h}^{\lambda}_{R_v,\mathcal{H}} = \arg\min_{h\in\mathcal{H}} obj(h \mid R_v, \lambda)$$

[2]通常 $V = 5$ または 10 を用いる．

と定める．以上の準備のもとで，CV 誤差（期待予測誤差の推定値）を，式

$$\text{EPE-CV}(\mathbf{Alg}[\lambda] \mid S, \mathcal{H}) := \frac{1}{N}\sum_{n=1}^{N} I[y_n \neq \hat{h}_{v(n)}(x_n)]$$

で定める．

本稿で考察する問題: 本稿では，次の問題を考察する．

定義 2 交差検定による正則化パラメータ推定（RPE-CV）*: 与えられた仮説空間* $\mathcal{H}$ と，データ集合 S，正整数 $V \geq 2$ に対して，全ての正値 $\lambda > 0$ において，CV 誤差を最小化するパラメータ値 $\hat{\lambda}$

$$\hat{\lambda}(S, \mathcal{H}) \;\; := \;\; \arg\min_{\lambda > 0} \text{EPE-CV}(\mathbf{Alg}[\lambda] \mid S, \mathcal{H}) \quad (3)$$

を求めよ．

言いかえると，この RPE-CV 問題は，全ての正値 $\lambda > 0$ に対して，λ を用いた RegERM に基づく学習アルゴリズム $\mathbf{Alg}[\lambda]$ の推定される期待予測誤差を最小化する λ を求める問題である．

2.3 順序決定木

順序付き変数集合 $(V, <)$ とは，n 個の変数の集合 $V = \{x_1, \ldots, x_n\}$ と V 上の全順序 $<$ の組である．$(V, <)$ 上の **順序決定木** （oblivious decesion trees）とは，V の変数を用いた決定木で，T のすべての根から葉へのパスにおいて，V の変数が全順序 $<$ の昇順で現れるものをいう．以後，$\mathcal{ODT} = \mathcal{ODT}(V, <)$ で $(V, <)$ 上の順序決定木のなす族を表す．

制約として，順序決定木 h の**サイズ**を葉の総数 $size(h)$ とし，S での h の**支持度** $sup(h \mid S)$ を，全ての葉に対する割り当てられたデータ数の最小値と定める．長部らにより提案された ODT アルゴリズム [10] は，順序決定木の族 $\mathcal{ODT}$ に対して，経験損失の厳密最適化を行う列挙型の学習アルゴリズムである．

補題 1 (ODT アルゴリズム) 順序付き変数 $(\mathcal{V}, <)$ と，データ集合 S，最大サイズ $K \geq 1$，最小支持度 $\sigma \in [0, |S|]$ が与えられたとき，ODT アルゴリズムは，サイズ毎の経験誤差最小解の K 項組 $(\hat{h}_1, \ldots, \hat{h}_K)$ を出力する．ここに，各 $1 \leq k \leq K$ に対して，最小解 $\hat{h}_k$ は次の条件を満たす仮説である：

(i) $\hat{h}_k$ は，$size(h) = K$，$sup(h \mid S) \geq \sigma$, 変数順序 $(V, <)$ の制約を満たす順序決定木である．

(ii) 上の条件 *(i)* をみたす全ての順序決定木 $h'_k \in \mathcal{H}_{ODT}$ に対して，$err(\hat{h}_k \mid S) \leq err(h'_k \mid S)$.

さらに，アルゴリズムは $\mathcal{T}_{ODT} = O(poly(N, K, n)\mathcal{F}_{S,\sigma})$ 時間と $\mathcal{S}_{ODT} = O(poly(N, K, n))$ 領域で解を返す．ここに，$N = ||S||$，$n = |V|$ である．

補題 1 より，各 k について $obj(h \mid S, \lambda)$ を比較することで，次が直ちに示せる．

系 1 (RegERM) ODT を用いて，正則化経験損失最小化問題を，$O(\mathcal{T}_{ODT} + K)$ 時間と $O(\mathcal{S}_{ODT})$ 領域で解くことができる．

3 解の特徴付け

本節では，単一の訓練例集合 R 上の RegERM 問題の解の特徴付けを与える．これに基づいて，凸包計算[3]を用いた正則化パラメータ推定の高速化手法を与える．この結果を一般化して，次節では，交差検定における複数のブロックにおける解の特徴付けに拡張し，これに基づいて，本稿の目的である RPE-CV の高速なアルゴリズムを提案する．

本節では，一般の仮説空間 $\mathcal{H}$ を仮定するが，一般性を失うことなく，記法としては準備の 2.3 節の ODT アルゴリズムの説明で使用した記号をそのまま使用する．すなわち，$\mathcal{H}_k$ と $\hat{h}_k$ で，それぞれ，サイズちょうど k の部分空間と，補題 1 の条件 (i) と (ii) を満たす $\mathcal{H}_k$ の経験誤差最小解を表す．

以下では，とくにふれなければ，R と，$\lambda \geq 0$，$K \geq$ は任意の一つの訓練データ集合と，任意の正実数，任意の非負整数を表すと約束する．そこで，文脈から R が明らかならば，$err(h)$ や，$obj(h \mid \lambda)$ 等のように R を省略することがある．

3.1 固定した λ に対する解の特徴付け

仮説サイズが同じならば，$err(\cdot)$ と $obj(\cdot \mid \lambda)$ は大小関係 $\leq$ を保存する．

補題 2 任意の λ と任意の $h, h' \in \mathcal{H}_k$ に対して，$err(h) \leq err(h') \iff obj(h \mid \lambda) \leq obj(h' \mid \lambda)$.

すなわち，$\mathcal{H}_k$ に属する仮説の中で，仮説 $\hat{h}_k$ よりも目的関数を小さくする仮説は存在しない．

系 2 任意の $\lambda \geq 0$ と R に対して，目的関数 $obj(h \mid \lambda)$ を最小にする仮説 h は，仮説空間 $\hat{\mathcal{H}} := \{\hat{h}_k \mid k = 1, \ldots, K\}$ に含まれる．

よって，仮説空間 $\hat{\mathcal{H}}$ に含まれる仮説についてのみ目的関数を評価すれば良いことがわかる．

3.2 スタンプ点を用いた λ 全体に対する解の特徴付け

スタンプ平面は，x-軸をサイズ $k \in [1, K]$ とし，y-軸を経験誤差 $e \in [0, 1]$ とする離散二次元平面 $\mathcal{S} = \{1, \ldots, K\} \times [0, 1]$ である．各要素 $p = (k, e) \in \mathcal{S}$ を**スタンプ点**という．R を仮定したとき，仮説空間 $\mathcal{H}$ に対し

[3]あとで述べるが，正確には左下包 (left-lower hull) のみを用いる．

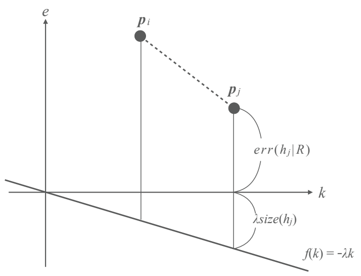

図 1: スタンプ点 p_i, p_j と直線 $f(k) = -\lambda k$．スタンプ点から $f(k)$ へ横軸に垂直に下ろした線分の長さが目的関数 $obj(h_j \mid \lambda) = err(h_j) + \lambda size(h_j)$ に対応しており，2 点の目的関数の大小関係は，λ と $\Lambda(h_i, h_j)$ の大小関係によって変化する．

て，そのスタンプ点集合を $\mathcal{S}(\mathcal{H}) := \{p(h) \mid h \in \mathcal{H}\}$ で定める．ここに，$p(h) = (size(h), err(h)) \in \{1, \ldots, K\} \times [0, 1]$ を**仮説 h のスタンプ点**と呼ぶ．以降で，文脈から明らかなときは，仮説 $h \in \mathcal{H}$ でそのスタンプ点 $p(h) \in \mathcal{S}$ と混同することがある．

最適仮説の族 $\hat{\mathcal{H}}$ に対して，$\mathcal{S}(\hat{\mathcal{H}})$ を最適スタンプ点集合と呼ぶ．$\mathcal{H}_k$ 中の最適仮説 $\hat{h}_k$ のスタンプ点を $\hat{p}_k := p(\hat{h}_k) = (k, \hat{e}_k)$ とすると，$\hat{e}_k$ は $\mathcal{H}_k$ の仮説によって達成される R 上の最小経験誤差である．以後，$me_k := e_k$ でサイズ k の最小経験誤差を表す．明らかに，$\hat{h}_k$ に対応するスタンプ点は，$\hat{p}_k := (k, me_k)$ と書ける．

ここで，図 1 に示すように，スタンプ点集合をプロットした二次元平面上で，直線 $L : f(k) = -\lambda \cdot k$ を考える．このとき，サイズ $1 \leq j \leq K$ の仮説 h に対応するスタンプ点 $p = (j, e)$ から直線 L へ k 軸に垂直に下ろした線分の長さを ℓ とすると，

$$\begin{aligned}\ell &= e + f(j) = e + \lambda \cdot j \\ &= err(h) + \lambda \cdot size(h) = obj(h \mid \lambda)\end{aligned}$$

と書ける．すなわち，スタンプ点 p と直線 $-\lambda \cdot k$ の距離が，仮説 h の目的関数 $odj(h \mid \lambda)$ に対応していることがわかる．

ここで，2 つのサイズが異なる任意の仮説 h, h' について，$p(h) = (k, e), p(h') = (k', e') \in \mathcal{H}$ とおく．今，$k < k'$ かつ $e \geq e'$ が成立すると仮定する．このとき，線分 $\overline{p(h), p(h')}$ の**傾き** $\Lambda(h, h')$ を，

$$\Lambda(h, h') := -\frac{e' - e}{k' - k} \geq 0$$

で定める[4]．以後，上の Λ を単に h と h' の**傾き**と呼ぶ．

[4]本稿では右下がりの線分のみを扱うので，便宜上，ここでの定義は，通常の線分の傾きに負号をつけたものにしている．

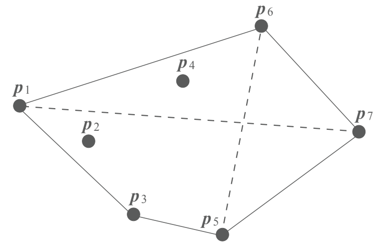

図 2: 例 1 の点集合 $Q = \{p_1, \ldots, p_7\}$ と，その凸包 $Conv(Q)$，下包 $Low(Q)$，左包 $Left(Q)$，左下包 $\mathcal{LLH}(Q)$ の例

補題 3 任意の $\lambda \geq 0$ と，仮説 h, h'（$size(h) < size(h')$）について以下が成立する：

$$\begin{aligned}obj(h \mid \lambda) > obj(h' \mid \lambda) &\Leftrightarrow 0 \leq \lambda < \Lambda(h, h') \\ obj(h \mid \lambda) = obj(h' \mid \lambda) &\Leftrightarrow \lambda = \Lambda(h, h') \\ obj(h \mid \lambda) < obj(h' \mid \lambda) &\Leftrightarrow \Lambda(h, h') < \lambda\end{aligned}$$

3.3 最適スタンプ点集合に対する左下包

ここで，図 2 に示すように，二次元平面上の点集合 Q に対する**凸包** $Conv(Q)$ は，Q の各点を内部に含む最小の凸多角形をなす Q の部分集合である．$Conv(Q)$ の最左点と最右点（または，最上点と最下点）を結んだ線分で分けて得られる 2 つの凸多角形のうち下側の（左側の）凸多角形の辺に含まれる Q の点の集合を**下包**（または，**左包**）と呼び，$Low(Q)$（または，$Left(Q)$）で表す．さらに，Q の**左下包**（left lower hull）を，$\mathcal{LLH}(Q) := Low(Q) \cap Left(Q)$ で定める．定義より，$\mathcal{LLH}(Q) \subseteq Conv(Q) \subseteq Q$ である．

例 1 図 *2* に，点集合 $Q = \{p_1, \ldots, p_7\}$ とその凸包，下包，左包，左下包の例を示す．ここに，凸包，下包，左包，左下包は，それぞれ，$Conv(Q) = \{p_1, p_3, p_5, p_6, p_7\}$ であり，$Low(Q) = \{p_1, p_3, p_5, p_7\}$, $Left(Q) = \{p_1, p_3, p_5, p_6\}$ である．$\mathcal{LLH}(Q) = \{p_1, p_3, p_5\}$ である．

定義より，$\mathcal{LLH}(Q)$ は，Q の最左点と最下点を含むことが容易に言える．上記の例でいうと，最左点 p_1 と最下点 p_5 が $\mathcal{LLH}(Q)$ に含まれていることがわかる．

アルゴリズムでは 3 点の位置関係が重要である．ここで，3 点 $p_1 = (x_1, y_1), p_2 = (x_2, y_2), p_2 = (x_3, y_3)$ の**符号付面積**（signed area）[9] を $A = Area(p_1, p_2, p_3) = (x_2 - x_1)(y_3 - y_2) - (x_3 - x_2)(y_2 - y_1)$ と定義する．符号付面積 A は，3 点 p_1, p_2, p_3 が左回りなら $A \geq 0$ をとり，右回りなら $A \leq 0$ をとる．等号が成立するのは，3 点が一直線上にあるとき，そのときだけである．

補題 4 任意の点 $q \in Q$ について，$p \in \mathcal{LLH}(Q)$ と以下の条件は同値である : 任意の点 $p, r \in \mathcal{LLH}(Q)$, $(p.x \le r.x)$ に対して，「$p.x \le q.x \le r.x \Rightarrow Area(p, q, r) \ge 0$」かつ，「$r.x \le q.x \Rightarrow r.y \ge q.y$」.

以上の議論から，最適スタンプ点集合 $\mathcal{S}(\hat{\mathcal{H}})$ に対して，その左下包 $\mathcal{LLH}(\mathcal{S}(\hat{\mathcal{H}}))$ に含まれないスタンプ点に対応する仮説が，目的関数を最小化することはないことを示す.

定理 *2* 任意の $\lambda > 0$ と任意の仮説 $\hat{h} \in \mathcal{H}$ について，次が成り立つ : $\hat{h}$ が，$\mathcal{H}$ 上で正則化経験損失関数 $obj(h \mid S, \lambda)$ を最小化するならば，$p(\hat{h}) \in \mathcal{LLH}(\mathcal{S}(\mathcal{H}))$ が成立する.

証明: 定理の対偶として，$p(h) \notin \mathcal{LLH}(\mathcal{S}(\mathcal{H}))$ ならば，ある仮説 $h_* \in \mathcal{H}$ が存在して，$obj(h_* \mid \lambda) < obj(h \mid \lambda)$ が成立すること（$*$）を示す.

まず，$p(h_k) \notin \mathcal{LLH}(\mathcal{S}(\mathcal{H}))$ を仮定する．すると，補題 *4* の対偶をとることで，ある最適仮説 $\hat{h}_l, \hat{h}_r \in \hat{\mathcal{H}}$ と対応する点 $\hat{p}_l, \hat{p}_r$ が存在して，以下のいずれかが成立することが示せる :

1. $l \le k \le r$ かつ $Area(\hat{p}_l, \hat{p}_k, \hat{p}_r) < 0$

2. $r \le k$ かつ $me_r < me_k$.

1. の場合について，主張（$*$）を背理法で証明する．すなわち，ある $\lambda \ge 0$ が存在して，$obj(\hat{h}_k \mid \lambda) \le obj(\hat{h}_l \mid \lambda)$ かつ $obj(\hat{h}_k \mid \lambda) \le obj(\hat{h}_r \mid \lambda)$ を仮定すると，

$$-\frac{me_r - me_k}{r-k} \le \lambda \le -\frac{me_k - me_l}{k-l} \tag{4}$$

が成立する．また，$Area(\hat{p}_l, \hat{p}_k, \hat{p}_r) < 0$ より，

$$-\frac{me_k - me_l}{k-l} < -\frac{me_r - me_k}{r-k} \tag{5}$$

が成立する．不等式 *(4)*，*(5)* より，条件を満たす λ は存在しない.

次に，*2.* の場合について，主張（$*$）を証明する．このとき，$r \le k$ かつ $me_r < me_k$ であるから，任意の λ に対して $obj(\hat{h}_r \mid \lambda) < obj(\hat{h}_k \mid \lambda)$ が成立する.

以上により，任意の λ に対して，仮説 $\hat{h}_k$ よりも目的関数を小さくする仮説が存在し，その仮説のスタンプ点は $\mathcal{LLH}(\mathcal{S}(\hat{\mathcal{H}}))$ に含まれることが示される. □

定理 *2* より，RegERM 問題に対する解を見つけるには，$\mathcal{LLH}(\mathcal{S}(\hat{\mathcal{H}}))$ 上の点だけを探せばよい.

よって，考慮すべき仮説の集合として $\hat{\mathcal{H}}_{\mathcal{LLH}}(R)$ を

$$\hat{\mathcal{H}}_{\mathcal{LLH}}(R) := \{h \in \hat{\mathcal{H}} \mid p(h) \in \mathcal{LLH}(\mathcal{S}(\hat{\mathcal{H}} \mid R)\}$$

で定めることとする．ただし，R が明らかな場合は $\hat{\mathcal{H}}_{\mathcal{LLH}} = \hat{\mathcal{H}}_{\mathcal{LLH}}(R)$ と書くこととする.

3.4 最適な λ の推定区間

補題 *5* 任意の仮説 $\hat{h}_i, \hat{h}_j, \hat{h}_k \in \hat{\mathcal{H}}_{\mathcal{LLH}}$ について，$i < j < k$ ならば，$0 \le \Lambda(\hat{h}_i, \hat{h}_j) \le \Lambda(\hat{h}_i, \hat{h}_k) \le \Lambda(\hat{h}_j, \hat{h}_k)$ が成立する.

元の最適仮説の k 座標の昇順列 $(\hat{h}_1, \ldots, \hat{h}_K)$ に対して，最適スタンプ点集合 $\mathcal{S}(\hat{\mathcal{H}})$ に対する左下包 $\mathcal{LLH}(\mathcal{S}(\hat{\mathcal{H}}))$ 上の点を k 座標の昇順で並べた部分列を $(\hat{h}_{\kappa(1)}, \ldots, \hat{h}_{\kappa(M)})$ とおく．ここに，$M = |\mathcal{LLH}(\mathcal{S}(\hat{\mathcal{H}}))| \le K$ は，左下包の要素数である．ここで，$\kappa : [1, M] \to [1, K]$ は添え字関数である.

このとき，以下が成立する.

定理 *3* 任意の仮説 $\hat{h}_k \in \hat{\mathcal{H}}_{\mathcal{LLH}}$ について，$1 \le m \le M$ を $k = \kappa(m)$ となる添え字とする．このとき，仮説 $\hat{h}_k$ が RegERM 問題の解となる必要十分条件は，

$$\Lambda(\hat{h}_k, \hat{h}_{\kappa(m+1)}) \le \lambda \le \Lambda(\hat{h}_{\kappa(m-1)}, \hat{h}_k).$$

証明: 仮説 $\hat{h}_k$ が RegERM 問題に対する解であると仮定すると，任意の仮説 $\hat{h}_{\kappa(l)}, \hat{h}_{\kappa(r)} \in \hat{\mathcal{H}}_{\mathcal{LLH}}$ $(l < m < r)$ に対して，補題 *3* より $\Lambda(\hat{h}_k, \hat{h}_{\kappa(r)}) \le \lambda \le \Lambda(\hat{h}_{\kappa(l)}, \hat{h}_k)$ が成立する．ここで $\Lambda(\hat{h}_k, \hat{h}_{\kappa(r)})$ について，補題 *5* より

$$\begin{aligned}\Lambda(\hat{h}_k, \hat{h}_{\kappa(r)}) &\le \Lambda(\hat{h}_k, \hat{h}_{\kappa(r-1)}) \\ &\le \Lambda(\hat{h}_k, \hat{h}_{\kappa(r-2)}) \le \cdots \le \Lambda(\hat{h}_k, \hat{h}_{\kappa(m+1)})\end{aligned}$$

となる．同様にして $\Lambda(\hat{h}_{\kappa(l)}, \hat{h}_k) \ge \Lambda(\hat{h}_{\kappa(m-1)}, \hat{h}_k)$ となるので，$\Lambda(\hat{h}_k, \hat{h}_{\kappa(m+1)}) \le \lambda \le \Lambda(\hat{h}_{\kappa(m-1)}, \hat{h}_k)$ が成立する. □

左下包上の点の添え字 $m = 1, \ldots, M-1$ に対して，$\Lambda^{(m)} := \Lambda(\hat{h}_{\kappa(m)}, \hat{h}_{\kappa(m+1)}) \in \mathbb{R}$ とする．これは，左下包上の左上から m 番目の辺 $\hat{p}_{\kappa(m)}, \hat{p}_{\kappa(m+1)}$ が一意に定める辺の傾きに対応する.

このとき，この訓練集合 R 上の選択境界 B を

$$B := \{\Lambda^{(1)}, \Lambda^{(2)}, \ldots, \Lambda^{(M-1)}\} \tag{6}$$

として定める.

補題 *6* B の各要素について，$\Lambda^{(1)} \ge \Lambda^{(2)} \ge \cdots \ge \Lambda^{(M-1)} \ge 0$ が成立する.

定理 *3* より，$m = 1, \ldots, M$ に対して選択区間 $SR^{(m)}$ を

$$SR^{(m)} := \begin{cases} [0, \Lambda^{(M-1)}), & if\, m = M \text{ のとき,} \\ [\Lambda^{(m)}, \Lambda^{(m-1)}), & if\, 1 < m < M \text{ のとき,} \\ [\Lambda^{(1)}, \infty) & if\, m = 1 \text{ のとき,} \end{cases}$$

で定める．$\mathcal{SR} = \mathcal{SR}(\mathcal{B}) = \{SR^m\}_{m=1}^{M}$ とおくと，$\mathcal{SR}$ は $[0, \infty)$ の分割となる．ここで，$[0, \infty)$ 上の同値関係 $\equiv_{\mathcal{SR}}$ を，$\lambda \equiv_{\mathcal{SR}} \lambda' \iff (\exists SR \in \mathcal{SR})\ \{\lambda, \lambda'\} \subseteq SR$ と定める．R 上の RegERM 問題に対する解となる仮説を $\hat{h}^{\lambda} = \hat{h}_R^{\lambda}$ で表す．定理 *3* から次の系が導ける.

系 3 任意の λ, λ' に対して $\lambda \equiv_{\mathcal{SR}} \lambda' \iff obj(\hat{h}^\lambda \mid R) = obj(\hat{h}^\lambda \mid R)$ が成立する．

系 *4* から，同値類 $[\lambda]_{\mathcal{SR}} := \{ \lambda' \mid \lambda \equiv_{\mathcal{SR}} \lambda' \}$ は，$obj(\hat{h}^\lambda \mid R)$ が同じ値をとる λ のなす集合であることがわかる．それぞれの区間 $SR^{(m)}$ に対して RegERM 問題に対する解となる仮説 $\hat{h}_{\kappa(m)}$ を対応づけることができる．

系 4 任意の $1 \le m \le M$ に対して，左下包上の仮説 $\hat{h}_{\kappa(m)} \in \hat{\mathcal{H}}_{\mathcal{LLH}}(R)$ が RegERM 問題に対する解となることは，$\lambda \in SR^{(m)}$ が成立することと同値である．

4 アルゴリズム

本節では，前節で与えた RPE-CV 問題の解の特徴付けから，本稿の目的である最適スタンプ点集合から正則化パラメータを高速に推定するアルゴリズム ConvexSearch を与える．

4.1 全てのブロックに対する RPE-CV 問題の解の特徴付け

交差検定により，全てのブロック $v = 1, \ldots, V$ に対して，*3* 節で説明した方法を用いて，v 番目の訓練例 R_v から選択境界 B_v が求まったとする．このとき，全ブロックに対する選択境界 $\mathcal{B}$ を，集合和 $\mathcal{B} := B_1 \cup \cdots \cup B_V$ で定める．

ここで集合 $\mathcal{B}$ の要素を値の昇順に並べた列を $\lambda_1 \le \cdots \le \lambda_{|\mathcal{B}|}$ とおく．l 番目の要素を λ_l として，全ブロックに対する選択区間 $SR^{(l)}$ を，

$$SR^{(l)} := \begin{cases} [0, \lambda_1), & if\ l = 1 \text{ のとき}, \\ [\lambda_{l-1}, \lambda_l), & if\ 1 < l < |\mathcal{B}| \text{ のとき}, \\ [\lambda_{|\mathcal{B}|}, \infty), & if\ l = |\mathcal{B}| + 1 \text{ のとき}, \end{cases}$$

で定め，それらがなす集合を

$$\mathcal{SR} = \mathcal{SR}(\mathcal{B}) := \{SR^{(1)}, \ldots, SR^{(|\mathcal{B}|+1)}\} \quad (7)$$

とする．

補題 7 選択区間の集合 $\mathcal{SR}$ について，$|\mathcal{SR}| = O(KV)$．

証明: $|\mathcal{SR}| = |\mathcal{B}| + 1 \le |B_1| + \cdots + |B_V| + 1$ かつ，任意の B_v について $|B_v| = |\mathcal{LLH}(\mathcal{S}(\hat{\mathcal{H}} \mid R)| - 1 \le K - 1$ より成立する． □

$\mathcal{SR}$ は $[0, \infty)$ の分割なので，前節の系 *4* にしたがって同値関係 $\equiv_{\mathcal{SR}}$ を定める．同様に，前節で導入した各ブロックごとの区間集合を $\{\mathcal{SR}_v\}_{v=1}^V$ で表す．V 個のブロックに対する $(R_v)_{v=1}^V$ 上の RegERM 問題に対する任意の解のなす組を $\mathbf{h}^\lambda = (\hat{h}_v^\lambda)_{v=1}^V$ で表し，その目標関数値を $\mathbf{obj}(\mathbf{h}^\lambda) = (obj(\hat{h}_v^\lambda \mid R_v))_{v=1}^V$ で表す．このとき，系 *4* の一般化として次の定理が成立する．

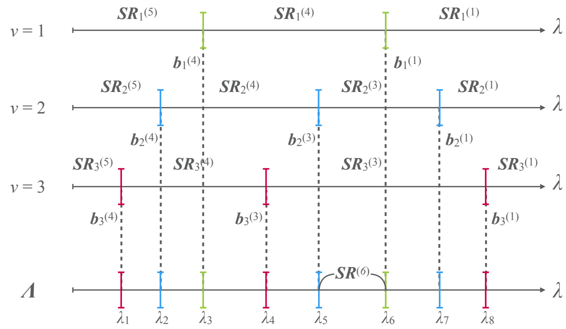

図 3: $\lambda \in SR^{(m)}$ の元で RPE-CV 問題の解となる仮説が対応づけられる．全ての v について選択境界 B_v を併合すると，各ブロックにおいて $\lambda \in SR^{(l)}$ の元で RPE-CV 問題の解となる仮説の V 個の組が対応づけられる区間 $SR^{(l)}$ がわかる．図の例の場合，$\lambda \in SR^{(6)}$ と $\{\hat{h}_1, \hat{h}_2, \hat{h}_3\} = \{\hat{h}_4, \hat{h}_3, \hat{h}_3\}$ が対応づけられる．

定理 4 任意の λ, λ' に対して $\lambda \equiv_{\mathcal{SR}} \lambda' \iff \mathbf{obj}(\mathbf{h}^\lambda) = \mathbf{obj}(\mathbf{h}^{\lambda\Gamma})$ が成立する．

証明: $\mathcal{SR}$ の構成から，$\lambda \equiv_{\mathcal{SR}} \lambda' \iff \bigwedge_{v=1}^V \lambda \equiv_{\mathcal{SR}_v} \lambda'$ であることが言えるので直ちに定理が示される． □

V 個の仮説の組 $\mathbf{h}^\lambda = (\hat{h}_1^\lambda, \ldots, \hat{h}_V^\lambda)$ を一つ対応づける．このとき，任意の選択区間 $SR^{(l)} \in SP$ に対して，それが含む任意の λ について組 $\mathbf{h}^\lambda$ は同一であることが言える．(図 *3*)．

よって，各選択区間に対応する V 個の仮説の組 $\{\hat{h}_1, \ldots, \hat{h}_V\}$ が高々 $O(KV)$ 個存在するので，それらの全ての仮説の組について CV による期待予測誤差の推定値を計算し，その最小値を達成した仮説の組に対応する選択区間が RPE-CV 問題に対する解となる．

4.2 アルゴリズム

各ブロック $v \in \{1, \ldots, V\}$ に対して，仮説集合 $\hat{\mathcal{H}}_v = \hat{\mathcal{H}}(R_v)$ と，$\hat{\mathcal{H}}_v$ の各仮説の訓練データ集合 R_v に対する分類誤差，及びテストデータ集合 S_v に対する誤分類数の集合をそれぞれ $TR_v := \{err(h) \mid h \in \hat{\mathcal{H}}_v\}$, $TS_v := \{\#err(h \mid S_v) \mid h \in \hat{\mathcal{H}}_v\}$ とする．

Algorithm 1 に，最適スタンプ点集合から左下包計算を用いて RPE-CV 問題に対する最適解を計算するアルゴリズムを示す．

関数 ConvexSearch では，以下の手順により RPE-CV 問題に対する最適解を計算する．

まず，各ブロック v において，*4* 行目で最適スタンプ点集合に対する左下包を関数 GrahamScanLLH により求める．これは，凸包計算アルゴリズムである Graham's scan に基づいたアルゴリズムで，最適スタンプ点集合に対しては計算時間 $O(K)$ で左下包を求めることができる．

Algorithm 1 入力として仮説集合 $\hat{\mathcal{H}}$ と，訓練データ R_v に対する分類誤差及びテストデータ S_v に対する誤分類数を受け取り，RPE-CV 問題に対する解となる λ の区間 SR を求めるアルゴリズム．

Input: $\hat{\mathcal{H}} = \{\hat{\mathcal{H}}_v\}_{v=1}^V, TR = \{\{err(h) \mid h \in \hat{\mathcal{H}}_v\}\}_{v=1}^V, TS = \{\{\#err(h \mid S_v) \mid h \in \hat{\mathcal{H}}_v\}\}_{v=1}^V$

Output: SR

1: **function** ConvexSearch(TR, TS)
2: **for** $v = 1$ to V **do**
3: $St_v := \{(size(h), err(h)) \mid h \in \hat{\mathcal{H}}_v\}$;
4: $\mathcal{LLH}_v \leftarrow$ GrahamScanLLH(St_v);
5: $\hat{\mathcal{H}}_v^{\mathcal{LLH}} := \{h \in \hat{\mathcal{H}}_v \mid p(h) \in \mathcal{LLH}_v\}$;
6: $B_v \leftarrow$ CalcBound($\mathcal{LLH}_v$);
7: **end for**
8: $\mathcal{B} := \bigcup_{v=1}^V B_v$;
9: $\mathcal{SR} \leftarrow$ CalcRange($\mathcal{B}$);
10: **for** $r \in \mathcal{SR}$ **do**
11: **for** $v = 1$ to V **do**
12: $\hat{h}_v \leftarrow$ SelectOptHypo($B_v, \hat{\mathcal{H}}_v^{\mathcal{LLH}}, r$);
13: **end for**
14: EPE-CV$_r := \sum_{v=1}^V \#err(\hat{h}_v \mid S_v)$;
15: **end for**
16: $r^* := \arg\min_{r \in \mathcal{SR}}$ EPE-CV$_r$;
17: **return** r^*;
18: **end function**
19: $SR \leftarrow$ ConvexSearch($\hat{\mathcal{H}}, TR, TS$);

次に，求めた最適スタンプ点集合の左下包から，*6* 行目で，関数 CalcBound によって選択境界 B_v（式 *(6)*）を計算する．

以上の操作を $v = 1, \ldots, V$ に対して行い，得られた $B_1, \ldots, B_V$ から *9* 行目で，関数 CalcRange によって全ブロックに対する選択区間 SR（式 *(7)*）を求める．

12 行目で，関数 SelectOptHypo は，与えられた $\lambda \in SP^{(l)}$ の元で，RegERM 問題に対する解となる仮説を返す．*14* 行目で，各選択区間 $SP^{(l)} \in SR$ における交差検定による期待予測誤差の推定値 EPE-CVを求める．

最後に *14* 行目で，EPE-CVの最小値を達成した選択区間を，RPE-CV 問題の最適解として出力する．

定理 *5* Algorithm 1は，入力として $TR = \{TR_1, \ldots, TR_V\}$ と $TS = \{TS_1, \ldots, TS_V\}$ を受け取り，RPE-CV 問題に対して最適な正則化パラメータ λ を $O(KV^2)$ 時間で正しく出力する．ここに，V は交差検定の分割ブロック数であり，K は仮説の最大サイズである．

定理 *5* と補題 *1* から *1* 節の定理 *1* が示される．

5 実験

5.1 データと方法

UCI ML レポジトリ[5]のデータセットの離散化版（*CP4IM*[6]）を用いた．表 *1* に使用したデータセットとその特徴の一覧を示す．

表 1: 実験に用いたデータセットの一覧

データセット名	データ数	属性数	目標クラス数
chess	3196	74	2
g-credit	1000	113	2
mushroom	8124	120	2
vote	435	49	2

表 2: アルゴリズムの実行時間の比較

Algorithm	δ	Grid num	Time(sec)
NAIVE	0.5	3	109.9
NAIVE	0.2	6	219.5
NAIVE	0.1	11	403.3
NAIVE	0.05	21	732.6
NAIVE	0.01	101	3667.9
FAST	NA	NA	36.7

本実験では，次の *2* つのプログラムを用いて，最適な正則化パラメータ λ，すなわち，RPE-CV 問題の最適解を求め，計算時間と *EPE-CV* を測定した．なお，正則化経験損失最小化には，*1* 節で紹介した *ODT* アルゴリズムの長部ら *[10]* による *C++*実装を用いた．以下では，$[\lambda_{\min}, \lambda_{\max}]$ で λ の推定区間を表す．

- PecvOdtNaive（Naive）*: 1* 節のグリッドサーチを *Python* で実装し，*ODT* アルゴリズムに対して適用したもの．刻み目 δ の値を，最大値 $\lambda = 1$ まで変えて実験を行った．

- PecvOdtFast (Fast)*: 3* 節の提案アルゴリズムを *Python* で実装し，*ODT* アルゴリズムに対して適用したもの．

パラメータとして最大サイズ $K = 10$ と交差検定の分割数 $V = 10$ を用いた．実験環境は，*PC*（*CPU Intel Core i5 2.9GHz, Memory 8GB, OS macOS Sierra 10.12.6*），コンパイラ *g++ ver4.2.1*，*Python ver3.6.0* を用いた．

5.2 実験 1：グリッドサーチと提案手法の比較

表 *2* と表 *3* に，データセット *mushroom* に対して，PecvOdtNaive と PecvOdtFast で最適解 $\hat{\lambda}$ を推定したときの計算時間と EPE-CVの最小値を示す．最小頻度は $\sigma = 1000$ を用いた．

表 *2* では，PecvOdtNaive は，δ の値が小さくなるほど実行時間が増加した．一方，PecvOdtFast では，*ODT* アルゴリズムが一度しか実行されないため，実行時間は短くなった．

表 *3* では，PecvOdtFast は，δ の値に関係なく EPE-CVの厳密な最小値を求めている．一方，PecvOdtNaive では，δ の値によって発見された EPE-CVの最小値が異なり，いずれの値も PecvOdtFast が計

[5]http://archive.ics.uci.edu/ml/
[6]https://dtai.cs.kuleuven.be/CP4IM/datasets/

表 3: アルゴリズムの EPE-CVの比較

Algorithm	δ	Grid num	$\hat{\lambda}$	EPE-CV
NAIVE	0.5	3	0.0	0.0706
NAIVE	0.2	6	0.0	0.0706
NAIVE	0.1	11	0.1	0.0649
NAIVE	0.05	21	0.1	0.0649
NAIVE	0.01	101	0.1	0.0649
FAST	NA	NA	[0.0118, 0.0121]	0.0567

表 4: 正則化の有無による EPE-CV$_2$ と最適仮説の平均サイズの比較

Dataset name	σ	λ	EPE-CV$_2$	Ave. size
chess	300	0	0.099	5.6
		[0.0037,0.0766]	0.098	4.2
g-credit	150	0	0.275	4.9
		[0.0037,0.0060]	0.275	4.5
mushroom	750	0	0.028	5.8
		[0.0001,0.0062]	0.028	5.8
vote	25	0	0.050	4.4
		[0.0029,0.3140]	0.043	2.2

算した値よりも大きくなっていることがわかる．実際，PecvOdtFast が見つけた最適区間の幅は $\Delta = 0.0121 - 0.0118 = 0.0003$ であり，Naive は最小の刻み幅でも 0.01 なので最適解を見つけるのは難しい．

5.3 実験 2：ODT アルゴリズムに対する正則化の評価

本稿で主題とした *RPE-CV* 法で求めた λ の有効性を確かめた．表 *4* に，*4* つのデータセット上で，比較対象として $\lambda = 0$ と，*2* 重交差検定により内側の交差検定で得た $\hat{\lambda}$ を用いて，外側の交差検定で得た *10* 次交差検定で求めた EPE-CV$_2$ と得られた仮説たちの平均サイズ *(Ave. size)* を示す．ここに，$\lambda = 0$ は経験誤差最小化に対応する．

$\lambda = \hat{\lambda}$ の元で仮説を評価した場合の方が仮説の平均サイズが小さくなっていることがわかる．加えて，EPE-CV$_2$ の値も小さくなるか同等程度となっており，より小さいサイズの仮説を選択しつつ未知のデータに対する分類精度も向上あるいは維持できている．

6 おわりに

本稿では，仮説空間に対するスタンプ点集合の概念を用いて，正則化経験損失最小化における超パラメータ推定の効率良いアルゴリズムを与えた．スタンプ点の概念は，最初に *Fukuda* ら *[2]* によって導入されたものである．さらに，長部ら *[1, 10]* が考察した順序決定木の族に対する経験損失の厳密最適化学習アルゴリズム ODT を用いて，実際に提案手法の有効性を検証した．

今後の課題として，提案手法の *LASSO* 解やルール集合 *[4, 5, 7]* に対する拡張が挙げられる．*ODT* のような厳密最適化手法が利用できない場合のランダムサンプリングを用いたスタンプ点集合の近似計算も興味深い．

謝辞

小宮山 純平，原 聡，瀬々 潤、津田 宏治，寺田 愛花，美添 一樹氏の諸氏，および，湊基盤 *(S)* プロジェクトメンバーには，貴重なコメントとご教示をいただきました．本研究は，*JSPS* 科研費 基盤研究 *JPA16H01743*，挑戦的萌芽研究 *JP15K12022* の助成および，北大 *GI-CoRE*「ビッグデータ・サイバーセキュリティ研究拠点」の支援を部分的に受けました．ここに謝意を表します．

参考文献

[1] *H. Arimura, K. Osabe, and T. Uno. Optimization and enumeration of decision trees from massive data sets. In* Proc. 21st Conf. of the Int'l Federation of Oper. Res. Soc. (IFORS 2017)*, pages ME–18, Quebec, July 16-21 2017.*

[2] *T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. Mining optimized association rules for numeric attributes. In* Proc. PODS 1996*, pages 182–191, 1996.*

[3] *J. Han, M. Kamber, and J. Pei.* Data mining: concepts and techniques*. Elsevier, 3rd edition, 2011.*

[4] *S. Hara and M. Ishihata. Approximate and exact enumeration of rule models. In* Proc. AAAI 2018 (to appear)*, Feb. 2018.*

[5] *S. Hara and T. Maehara. Enumerate lasso solutions for feature selection. In* Proc. AAAI 2017, San Francisco, USA.*, pages 1985–1991, 2017.*

[6] *T. Hastie, R. Tibshirani, and J. Friedman.* The Elements of Statistical Learning*. Springer Series in Statistics. 2001.*

[7] *J. Komiyama, M. Ishihata, H. Arimura, T. Nishibayashi, and S. Minato. Statistical emerging pattern mining with multiple testing correction. In* Proc. ACM KDD 2017, Halifax, August*, pages 897–906, 2017.*

[8] *M. Mohri, A. Rostamizadeh, and A. Talwalkar.* Foundations of Machine Learning*. The MIT Press, 2012.*

[9] 浅野 哲夫．計算幾何―理論の基礎から実装まで．共立出版*, 2007.*

[10] 長部和仁，宇野毅明，有村博紀．アイテム集合列挙に基づく最適な順序付き決定木の高速発見．人工知能基本問題研究会*, 102:32–39, 2016.*