情 報 学 基 礎 22－5
（1991. 7. 10）

決定木による分類規則の学習について - 理論的側面から -

榊原 康文
（株）富士通研究所 国際情報社会科学研究所

決定木の学習について，理論的結果と実際的システムについて紹介する．理論的結果では，Valiant によって導入された確率的学習モデル（PAC 学習モデル）の上で，ノイズを含んだデータからの学習可能性について考察する．実際的システムとしては，Quinlan による情報理論に基づくエントロピーを用いた ID3 について紹介する．さらに現在最も重要な研究課題であると思われる，新しい属性の発見について簡単にふれる．

On Learning Classification Rules Using Decision Trees

Yasubumi Sakakibara
International Institute for Advanced Study of Social Information Science (IIAS-SIS)
FUJITSU LABORATORIES LTD.
140, Miyamoto, Numazu, Shizuoka 410-03, Japan
E-mail : yasu%iias.flab.fujitsu.co.jp@uunet.uu.net

We consider the learning problem of decision trees from the theoretical point of view and practical point of view. We investigate the polynomial-time learnability of decision trees in the presence of noise on the distribution-independent model of concept learning from random examples introduced by Valiant. We survey the practical learning system for decision trees, ID3, by Quinlan which uses an information based method. We also briefly mention the problem of feature discovery.

1 はじめに

機械に学習能力を持たせることを目的とする機械学習の研究では, 帰納的推論と呼ばれる, 与えられた個々の事実から一般的な規則を導き出す推論が重要な学習方法として研究されている. 最近特に, 帰納的に推論（学習）するアルゴリズムを, 効率の良さ（効率の良いアルゴリズムとは, 関連するパラメータの多項式時間で実行するアルゴリズム）を問題にしながら設計し, その理論的解析を行なう, いわゆる計算論的学習理論と呼ばれる研究が盛んである. 本稿では, この帰納的推論の学習対象として, 分類規則を表現する方法の一つである決定木 (decision tree) を取り上げる. 決定木は, 世の中の物（事例）が属性とその値の対の集合で定義されている場合に, これらをいくつかのクラスに分類するための規則を表現する方法の一つである. そしてその決定木の学習は, 医療診断システムなどの大量のデータを処理する分野などで用いられ, 例えばそこでの学習問題は, 過去の患者の事例から病名を診断する決定木を学習するという問題になる.

本稿では, この決定木の学習アルゴリズムについて, 理論的結果と実際的システムについて紹介する. 理論的結果では, Valiant によって導入された確率的学習モデル（PAC 学習モデル）[13] の上で, 学習可能性について考察する. 特に, 大量のデータからの学習という設定においては, データにノイズが含まれるという仮定は実際的であるので, ここではノイズを含んだデータからの学習について考察する. 実際的システムとしては, Quinlan による情報理論に基づくエントロピーを用いた ID3 [6] について紹介する. さらに現在最も重要な研究課題であると思われる, 新しい属性の発見 [5] について簡単にふれる.

2 決定木

決定木 (decision tree) は, 世の中の物（事例）が属性とその値の対の集合で定義されている場合に, これらをいくつかのクラスに分類するための規則を表現する方法の一つである.

$$\text{物（事例）} = \{(\text{属性}1, \text{値}), (\text{属性}2, \text{値}), \ldots\}$$

そしてその決定木の学習は, 医療診断システムなどの大量のデータを処理する分野などで用いられ, 例えばそこでの学習問題は, 過去の患者の事例から病名を診断する決定木を学習するという問題になる.

ここでは, 各属性はブール属性と仮定し, それらを 2 つ（0 または 1）のクラスに分類する規則を考えることにする. すなわち, これらの規則はブール関数を表す.

今, n 個のブール変数（このような変数の集合を $V_n = \{x_1, x_2, \ldots, x_n\}$ で表す）があると仮定する. $X_n = \{0,1\}^n$ とおく. X_n の要素を**割り当て**と呼ぶ. すると任意のブール関数は, X_n から $\{0,1\}$ への関数として定義される.

決定木とは, 各内部ノードにブール変数が, 各葉に 0 または 1 がラベル付けされた 2 分木である. 決定木は次のようにブール関数を定義する. 1 つの割り当て $\vec{a} \in X_n$ は, 決定木の根から葉への 1 つのユニークなパスを決定する: 各内部ノードにおいて, そのノードにラベル付けされているブール変数の値がその割り当てにおいて 0 ならば左（1 ならば右）の枝をたどる. たどり着いた葉の値が, その割り当てに対する関数の値となる.

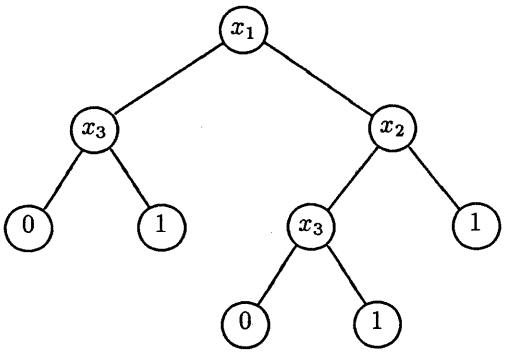

図 1: A decision tree representation for $x_1x_2 \vee x_3$.

3 決定木の学習

まず初めに, 与えられた有限個の具体例から決定木を構築する単純な学習アルゴリズムについて説明し, 非形式的な議論をすることによって, 直観的理解を得ることにする.

今, 割り当て $\vec{a} \in X_n$ とそれに対する値 $l \in \{0,1\}$ の対 $\langle \vec{a}, l \rangle$ (これを, 例と呼ぶ) の有限集合 S (これを**サンプル**と呼ぶ) が与えられた時, S を正しく分類する決定木を求めるアルゴリズムを考える. この方法は, 与えられたサンプルに矛盾しない推測を求めるというもので, 学習アルゴリズムが一般的に用いる常套手段である.

次の記号の準備をして以下にそのアルゴリズムを与える. $x = x_i \in V_n$ と仮定する. S_0^x (S_1^x) は, S 中の例 $\langle \vec{a}, l \rangle$ で $\vec{a} = (a_1, \ldots, a_n)$ かつ $a_i = 0$ ($a_i = 1$) であるものすべての集合を表すとする. S_0^x と S_1^x が共に非空である時, 変数 x は **informative** であるという.

ALGORITHM *FINDT*(S)

Input: A sample S.

Output: A decision tree T.

Procedure:

1. If all pair $\langle \vec{a}, l \rangle$ in S has $l = 1$, stop and return the decision tree $T = 1$;
   If all pair $\langle \vec{a}, l \rangle$ in S has $l = 0$, stop and return the decision tree $T = 0$;
2. For some informative variable $x \in V_n$
   (a) Let T_0^x =*FINDT*(S_0^x) and T_1^x =*FINDT*(S_1^x);
   (b) Stop and return the decision tree with root labelled x, left subtree T_0^x and right subtree T_1^x;

アルゴリズム *FINDT*(S) が必ず停止して, サンプル S に矛盾しない決定木を出力することは, 用意に示すことができる. 本稿で示される学習アルゴリズムは, すべてこのアルゴリズムを基本にしている.

アルゴリズム $FINDT(S)$ では，ステップ2においてどの変数を選択するかによって，得られる決定木のサイズやアルゴリズムの計算量が大きく変わってくる．以下では，決定木にランクという制限を入れることによって，変数の選択を決める方法を用いた効率的な学習に関する理論的な結果と，情報理論に基づくヒューリスティックを使って変数を選択する方法を用いた実際的な学習システム ID3 について考察する．特に，理論的結果においては，ノイズを含んだデータから学習する問題を考える．

4 学習可能性

Valiant によって導入された確率的学習モデル（PAC 学習モデル）[13] の上で，ノイズに強い効率的な学習アルゴリズムについて考察する．

4.1 PAC 学習モデル

いま学習アルゴリズムが学習対象とするブール関数のクラスを F_n で表し，これから学習される F_n 中の**未知の関数**を f_U で表す．F_n のことを**仮説空間**とも呼ぶ．以下に定義される学習モデルでは，このように予め固定されたブール関数のクラスから未知の関数が選ばれると仮定する．

いくつかの用語の定義をする．f_U の**例**とは，対 $\langle \vec{a}, f_U(\vec{a}) \rangle$ のことである．ここで，$\vec{a} \in X_n$．**サンプル**とは，例の有限集合のことである．あるブール関数 g が**例 $\langle \vec{a}, l \rangle$ に矛盾しない**とは，$g(\vec{a}) = l$ であるときをいう．g が**サンプルに矛盾しない**とは，そのサンプル中のすべての例に g が矛盾しないときをいう．

Valiant によって導入された PAC 学習モデル [13] では，まず定義域 X_n 上にある固定された確率分布 D を仮定する．この確率分布 D は任意に選ばれ，また学習機械にはこの D に関する情報は一切与えられない．このようにどんな確率分布の下でもうまく学習することを要請するので，このモデルは distribution-independent モデルとも呼ばれている．学習アルゴリズムは，入力を取らない**サンプリングオラクル** $EX()$ と呼ばれるブラックボックスにアクセスすることができる．$EX()$ が呼ばれると，分布 D に従ってある要素 $\vec{a} \in X_n$ が独立に取り出され，例 $\langle \vec{a}, f_U(\vec{a}) \rangle$ が学習アルゴリズムに返される．学習の成功基準は二つの実数，**正確さのパラメータ** ϵ と**信頼性のパラメータ** δ に関して定義されている．ただし，$0 < \epsilon \leq 1$，$0 < \delta \leq 1$．二つのブール関数 f と g の間の確率分布 D に関する異なり具合を

$$d(f,g) = \sum_{f(\vec{a}) \neq g(\vec{a})} \mathrm{Pr}_D(\vec{a})$$

で定義する．ここで，$\mathrm{Pr}_D(\vec{a})$ は確率分布 D に従って要素 $\vec{a}$ が取り出される確率を表す．$EX()$ の一回の呼び出しにおいて，f と g の値が異なる要素が得られる確率はちょうど $d(f,g)$ である．**ϵ 近似な関数**とは，$d(g, f_U) \leq \epsilon$ である関数 g をいう．学習アルゴリズムがブール関数のクラス F_n を PAC(probably approximately correctly) **学習する**とは，学習アルゴリズムが常に停止して次の条件を満たすブール関数 g を出力することを言う．

$$Pr(d(g, f_U) \leq \epsilon) \geq 1 - \delta.$$

すなわち，δ で保証された高い確率で，推測されたブール関数 g と未知のブール関数との間には，ϵ で保証される範囲内の違いしかないということである（ϵ 近似な関数を出力する）．

またこのような学習アルゴリズム A が**多項式時間PAC学習する**とは, A の計算時間が n, $1/\epsilon$ と $1/\delta$ の多項式で押さえられることを言う.

効率的なPAC学習可能性は, 学習対象とするブール関数のクラス F_n のサイズ $|F_n|$ にかかってくる. ブール関数のクラス F_n は, ある定数 t に対して $\ln(|F_n|) = O(n^t)$, すなわち $\ln(|F_n|)$ が n の多項式で押さえられる時, **多項式サイズ**であるという.

Ehrenfeucht と Haussler [3] は, 決定木に次に定義されるランクという制限を入れることによって, ランクが高々 r の決定木のクラスは, 多項式時間PAC学習可能であることを示した. ただしこの結果は, サンプルにノイズが含まれない場合の結果である.

決定木 T の**ランク**, $r(T)$, は, 次のように再帰的に定義される:

1. $T = 0$ または $T = 1$ ならば $r(T) = 0$
2. r_0 が T の左の部分木のランクで r_1 が右の部分木のランクの時,

$$r(T) = \begin{cases} \max(r_0, r_1) & \text{if } r_0 \neq r_1 \\ r_0 + 1 & \text{otherwise.} \end{cases}$$

4.2 ノイズデータからの学習

PAC学習モデルの上で提案されている Angluin と Laird による**分類ノイズモデル** (Classification noise process) [1] と呼ばれるノイズモデルについて考察し, 分類ノイズが存在する場合に効率的学習アルゴリズムを構築するための手法として, Noise-tolerant Occam algorithm (NTOA) を提案する. そしてブール関数のクラスに対してこの NTOA が存在するならば, それを使ってその関数のクラスに対するノイズに強い多項式時間PAC学習アルゴリズムが構築できることを示す. 尚, 本稿では紙面の都合上, 結果を示すだけにとどまり, これらの証明は一切省く. 興味ある読者は文献 [10, 9, 11] を参照されたい.

分類ノイズが存在する場合には, サンプリングオラクル $EX_\eta()$ (分類ノイズが存在するということを示すために, 添字 η (ノイズの割合) を付ける) から次のように例が取られて来ると仮定する.

$EX()_\eta$ が呼ばれると, 分布 D に従ってある要素 $\vec{a} \in X_n$ が独立に取り出されるが, 例が学習アルゴリズムに返される時に, 各例に対して独立に, 確率 η で, $f_U(\vec{a}) = 1$ の時に $\langle \vec{a}, 0 \rangle$ が, $f_U(\vec{a}) = 0$ の時に $\langle \vec{a}, 1 \rangle$ が返される.

ノイズの割合 η は, $1/2$ 未満であると仮定する. またノイズの割合に関する情報, ここではノイズのある上限 $\eta_b \geq \eta$, が学習アルゴリズムに与えられると仮定する. さらに多項式時間PAC学習アルゴリズムの実行時間は, $1/(1-2\eta_b)$ の多項式となることを許す.

学習アルゴリズムがブール関数のクラス F_n を**ノイズデータから多項式時間PAC学習する**とは, サンプリングオラクル $EX_\eta()$ が与えられた時に, 学習アルゴリズムが常に停止して

$$Pr(d(g, f_U) \leq \epsilon) \geq 1 - \delta$$

なるブール関数 g を出力し, かつその計算時間が n, $1/\epsilon$, $1/\delta$ と $1/(1-2\eta_b)$ の多項式で押さえられることを言う.

ノイズがない場合に, PAC学習アルゴリズムが用いる基本的な戦略は, 与えられたサンプルに矛盾しないブール関数を求めて出力するという方法である [2]. しかし, ノイズが存在

する場合には，この方法は使えない．すなわち，ノイズが影響して，サンプルに矛盾しない関数を F_n 中に見つけることができない場合がある．そこで，以下に与える方法は，サンプルに矛盾しないブール関数を求める代わりに，サンプル中の（すべての例にではなく）ほとんどの例に矛盾しない関数を求めるという戦略を取る．

記号の用意をする．ブール関数 f とサンプル S に対して，$F(f,S)$ は f が矛盾する S 中の例の個数を表すと定義する．したがって，$F(f,S)=0$ ならば，f はサンプル S に矛盾しない関数である．

ブール関数のクラス F_n に対する **Noise-tolerant Occam algorithm (NTOA)** とは，$EX_\eta()$ から取られた m 個の例からなるサンプル S とパラメータ ϵ,δ,η_b が与えられた時，

1. 次の条件を満足するブール関数 $g\in F_n$ を $1-\delta/2$ 以上の確率で出力し，

$$\frac{F(g,S)}{m} \leq \eta_b + \frac{\epsilon(1-2\eta_b)}{4},$$

2. その実行時間が n と m の多項式で押さえられる

アルゴリズムをいう．

これより次の結果を得ることができる．

定理 1 F_n は多項式サイズであると仮定する．この時，ブール関数のクラス F_n に対して $NTOA$ が存在するならば，F_n はノイズデータから多項式時間 PAC 学習可能である．必要とされるサンプルの大きさは，

$$m \geq \frac{8}{\epsilon^2(1-2\eta_b)^2}\ln\left(\frac{2|F_n|}{\delta}\right)$$

である．

4.3 効率的学習アルゴリズム

決定木に対する NTOA を構築し，決定木がノイズデータから効率良く学習可能であることを示す．ランクが r の決定木のクラス r-DT(n) に対する NTOA を図 2に示す．

アルゴリズム $FINDT$ と $RFINDT$ との違いは，まずノイズデータに対処するために，$FINDT$ のステップ 1が，$RFINDT$ のステップ 1, 2に拡張されている．そこで使われる定数 Q_F と Q_I を $RFINDT$ の再帰的呼び出しで保持するために，2 つの引数が拡張されている．また，$FINDT$ のステップ 2における変数の選択に対しては，ランクが高々 r の決定木を求めるようにするために，$RFINDT$ の再帰的呼び出しにおいて，引数にランク $r-1$ を持たせる拡張と，ステップ 4cを設ける拡張がされている．

補題 2 $Q_F=\eta_b+\frac{\epsilon(1-2\eta_b)}{8}$, $Q_I=\frac{\epsilon(1-2\eta_b)}{4(en/r)^r}|S|$ とする．もしアルゴリズム $RFINDT(S,r,Q_F,Q_I)$ が決定木 T を出力するならば，T はランクが高々 r の決定木であり，かつ

$$\frac{F(T,S)}{|S|} \leq \eta_b + \frac{\epsilon(1-2\eta_b)}{4}$$

である．

ALGORITHM *NODT*
Input:

- A sample S of m examples drawn from $EX_\eta()$ subject to classification noise,
- Integers $n, r \geq 0$ and positive fractions ϵ, and η_b, with $0 \leq \eta \leq \eta_b < 1/2$.

Output:

A decision tree T of rank at most r such that $F(T,S)/m \leq \eta_b + \epsilon(1-2\eta_b)/4$ if one exists, else "none".

Procedure:

1. Calculate the following:

$$\begin{aligned} Q_F &= \eta_b + \frac{\epsilon(1-2\eta_b)}{8}; \\ Q_I &= \frac{\epsilon(1-2\eta_b)}{4(en/r)^r}|S|; \end{aligned}$$

2. Call $RFINDT(S, r, Q_F, Q_I)$;
3. Let $T = RFINDT(S, r, Q_F, Q_I)$;
4. Output T and halt.

Subprocedure $RFINDT(S, r, Q_F, Q_I)$:

1. If $F(1,S)/|S| \leq Q_F$, stop and return the decision tree $T = 1$;
   If $F(0,S)/|S| \leq Q_F$, stop and return the decision tree $T = 0$;
2. If $|S| \leq Q_I$ and $F(1,S) \leq F(0,S)$, stop and return the decision tree $T = 1$;
   If $|S| \leq Q_I$ and $F(0,S) \leq F(1,S)$, stop and return the decision tree $T = 0$;
3. If $r = 0$, stop and return "none";
4. For each informative variable $x \in V_n$
   (a) Let $T_0^x = RFINDT(S_0^x, r-1, Q_F, Q_I)$ and $T_1^x = RFINDT(S_1^x, r-1, Q_F, Q_I)$;
   (b) If both recursive calls are successful (i.e., neither T_0^x ="none", nor T_1^x ="none"), then stop and return the decision tree with root labelled x, left subtree T_0^x and right subtree T_1^x;
   (c) If one recursive call is successful but the other is not, then
      i. Reexecute the unsuccessful recursive call with rank bound r instead of $r-1$;
      ii. If the reexecuted call is now successful, then let T be the decision tree with root labelled x, left subtree T_0^x and right subtree T_1^x, else let T ="none";
      iii. Stop and return T;
5. Stop and return "none".

図 2: Efficient robust learning of decision trees of rank r

補題 3 $Q_F = \eta_b + \frac{\epsilon(1-2\eta_b)}{8}$, $Q_I = \frac{\epsilon(1-2\eta_b)}{4(en/r)^r}|S|$ とする. また S は, 未知のランクが r の決定木に対するサンプリングオラクル $EX_\eta()$ から得られた m 個の例からなるサンプルとし,

$$m \geq \frac{128r(en/r)^r}{\epsilon^3(1-2\eta_b)^3}\ln\left(\frac{2en}{r\delta}\right)$$

とする. この時少なくとも $1-\delta/2$ の確率で, $RFINDT(S, r, Q_F, Q_I)$ は決定木 T を出力する.

補題 4 $EX_\eta()$ から得られた任意の非空のサンプル S と $r \geq 0$ に対して, $RFINDT(S, r, Q_F, Q_I)$ の実行時間は $O(|S|(n+1)^{2r})$ である.

以上の結果より, 次の定理を得る.

定理 5 *NODT* は, ランクが r の決定木のクラスに対する *NTOA* である.

定理 6 r-$DT(n)$ ノイズデータから多項式時間 *PAC* 学習可能である.

5 実際的システム: ID3

Quinlan によって導入された ID3 では, アルゴリズム $FINDT(S)$ における変数の選択を行なうステップ 2において, 情報理論に基づくエントロピーを用いて変数を選択する.

与えられたサンプル S 中で, 値 0 を取る (クラス 0 に分類される) 例 $\langle\vec{a}, 0\rangle$ の集合を S_0, 値 1 を取る例 $\langle\vec{a}, 1\rangle$ の集合を S_1 で表し, $n = |S_0|$, $p = |S_1|$ とする. ID3 の方法は次の 2 つの考え方を仮定する.

1. S に対する決定木は, S 中の任意の要素を確率 $|S_0|/|S|$ でクラス 0 に, 確率 $|S_1|/|S|$ でクラス 1 に分類すると考える. つまり, (0 または 1 に) 分類される要素の集合の大きさを確率と同一視することにする.

2. 決定木は, 割り当てを入力するとその値 (分類されるクラス) を返すので, メッセージ 0 または 1 を伝達する確率的な情報機械と見なすことができる. このとき, 決定木が伝達する情報量 (エントロピー) は,

$$I(p,n) = -\frac{p}{p+n}\log\frac{p}{p+n} - \frac{n}{p+n}\log\frac{n}{p+n}$$

で与えられる.

今, 変数 $x = x_i \in V_n$ が根にラベル付けされる変数として選ばれ, x は S を 2 つの集合 S_0^x と S_1^x に分けたとする. S_i^x 中で値 0 を取る例の個数を n_i, 値 1 を取る例の個数を p_i とする ($i \in \{0,1\}$). S_i^x に対する決定木の情報量は $I(p_i, n_i)$ で与えられる. したがって, 変数 x を根にもつ決定木の情報量は, 加重平均

$$E(x) = \sum_{i=0}^{1}\frac{p_i+n_i}{p+n}I(p_i,n_i)$$

となる. よって, 変数 x を選択したことによる情報量の増加は

$$gain(x) = I(p,n) - E(x)$$

となる. そこで ID3 は, 変数を選択する基準として, $gain(x)$ を最大にする変数 x を選ぶという方法を用いる.

6 属性の発見

決定木の表現方法を用いる上での問題点として, replication 問題というものがある. これは, 決定木の内部に同じパターンの部分木が何度も現れて木のサイズが大きくなってしまう問題である. これは, 決定木の各ノードに一つの変数（属性）しか割り当てられないことから起こる問題である. 例えば, 項の間で共有される変数が無いような選言標準形のブール式を決定木を用いて表現しようとすると, その中でサイズが最小の決定木でも, 同じパターンの部分木が何度も現れてしまう. この問題は, リテラルの連言 (conjunction) を内部ノードに割り当てる拡張した決定木を用いれば解決する.

Pagallo と Haussler [5] は, 学習中にこのような適切なリテラルの連言を見つけて, それを新しい属性として属性集合に加えて決定木を構築する方法を提案している. 彼らは, FRINGE, GREEDY3, GROVE と呼ばれる 3 つのアルゴリズムを提案している. これらの 3 つのアルゴリズムは, 決定木を構築するアルゴリズムとして, 基本的には *FINDT* を用いる（正確には, GREEDY3 と GROVE は, 決定リスト [8] と呼ばれる特殊な決定木しか構築しない）が, 学習中に新しい属性としてリテラルの連言を見つけ出し, それを属性集合 V_n に加えて, *FINDT* のステップ 2 における属性の選択の候補とするところが新しい. 3 つのアルゴリズムは, リテラルの連言を発見する際に, 異なる方法を用いる.

例えば FRINGE は, まず, ID3 で最初の属性（変数）だけを用いて決定木を生成する. そして 1 とラベル付けされた葉に至る長さ 2 以上のパス上の（葉に近い）最後の 2 つのノードに割り当てられている属性から（それらの否定を取るなどして）連言を新しく作り出す. 次に, 得られた新しい属性を加えた属性集合を用いて, ID3 で決定木を作り直す. これを繰り返すと, 徐々に大きな連言が得られる.

7 まとめ

本稿では, 決定木の学習について, PAC 学習モデルの上でノイズデータからの学習可能性に関する理論的結果と, 情報理論に基づくエントロピーを用いた学習システム ID3 と属性の発見の話題について簡単に触れた. しかし, ここで紹介した研究は, 決定木の学習に関する研究のほんの一部であって, 取り上げなかった重要な研究としては, 連続関数や確率的概念の学習への拡張 [4] や, 逐次呈示データからの学習 (incremental learning) [12], 情報理論における最小記述原理 (MDLP) を用いた学習 [7] などがある.

また今後の課題として, 例えば, ID3 システムは（ある条件の下で）多項式時間 PAC 学習アルゴリズムとなっているかを調べることは, 興味深いと思われる.

参考文献

[1] D. Angluin and P. Laird. Learning from noisy examples. *Machine Learning*, 2:343–370, 1988.

[2] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36:929–965, 1989.

[3] A. Ehrenfeucht and D. Haussler. Learning decision trees from random examples. *Information and Computation*, 82:231–246, 1989.

[4] M. Kearns and R. E. Schapire. Efficient distribution-free learning of probabilistic concepts. In *Proceedings of 31st IEEE Symposium on Foundations of Computer Science*, pages 382–391. IEEE Computer Society Press, 1990.

[5] G. M. Pagallo and D. Haussler. Boolean feature discovery in empirical learning. *Machine Learning*, 5:71–99, 1990.

[6] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[7] J. R. Quinlan and R. L. Rivest. Inferring decision trees using the minimum description length principle. *Information and Computation*, 80:227–248, 1989.

[8] R. L. Rivest. Learning decision lists. *Machine Learning*, 2:229–246, 1987.

[9] Y. Sakakibara. An efficient robust algorithm for learning decision lists. Research Report 105, IIAS-SIS, FUJITSU LIMITED, 1990.

[10] Y. Sakakibara. Occam algorithms for learning from noisy examples. In *Proceedings of 1st Workshop on Algorithmic Learning Theory (ALT'90)*, pages 193–208. Ohmsha, Ltd, 1990.

[11] Y. Sakakibara. Algorithmic learning of formal languages and decision trees. Unpublished manuscript, 1991.

[12] P. E. Utgoff. Incremental induction of decision trees. *Machine Learning*, 4:161–186, 1989.

[13] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27:1134–1142, 1984.