人 工 知 能 87−3
（1993. 3. 4）

決定木学習によるデータ分類と派生語の抽出

小柴健史　　榊原康文
（株）富士通研究所　国際情報社会科学研究所

文書データなどを分類する表現の一つとして決定木がある．この場合，決定木は文字列上に定義されるが，決定木を構成する上でノードに割り当てられる文字列をどのように決定するかが問題になる．ここでは，文字列の選択方法も含めて帰納的に決定木を構成する学習アルゴリズムを用いる．このアルゴリズムでは，ノードに割り当てられる文字列の比較に派生語的関係を利用する．

Classifying Document Data and Extracting Derivative Keywords by Learning Decision Trees

Takeshi Koshiba　　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: koshiba@iias.flab.fujitsu.co.jp
yasu@iias.flab.fujitsu.co.jp

A decision tree is a useful representation for classifying document data, where attributes are defined over strings. It is one of serious problems in constructing a decision tree to determine an attribute as a label of an internal node. In this paper, we use a learning algorithm which constructs decision trees recursively. We propose a derivative matching in process of learning decision trees, and consider a performance of derivative matching.

1 はじめに

近年，文書データの分類法やそのデータからキーワードを抽出する方法等の研究・開発が，情報検索の分野の研究において盛んに行なわれている．特に，CD-ROM などのように容易に大量データを保存できるような媒体が利用でき，また，大量のデータの中から必要な情報を効率的に取り出すことが要求されている今日においては計算機による処理技術に期待を寄せるところが大きいといえる．こうした問題に対して自然言語処理やデータ圧縮等の技術を利用して様々な研究がなされている [RL92]．しかしながら，そこで利用されている手法は往々にして複雑で，必ずしも効率的なものとは言い難いのが現状である．

本稿では，機械学習の研究成果を文書データの自動分類へ応用する一つの方法を提案する．情報検索においては，必要な情報を取り出すための適当なキーワードをいかにして抽出するかが重要な課題であり，また，難しい問題でもある．この問題に対して機械学習のアルゴリズムを適用させることを考える．また，機械学習のアルゴリズムが抽出したキーワード群の中で幾つかのキーワードが互いに類似していることが多いことに着目し，類似した幾つかのキーワードをまとめて扱うための方法を提案する．この方法を用いたアルゴリズムが効率的に動作するかについて考察をする．

2 決定木によるデータ分類

ここで扱う決定木は文書データを分類する一つの表現である．まず，決定木による分類の対象となる個々のデータは属性とその値との対からなる集合で表現されるものとする．このような構造を持つデータを事例と呼び，事例の集合をサンプルと呼ぶ．文書データを分類することが目的であるので属性として文字列上の属性を対象にする．つまり，文字列が持っている特徴・性質に関するものである．例えば,「文字列があるキーワードを部分文字列として含むか」というのも文字列上の属性である．

本稿では，文字列上の属性として「文字列があるキーワードの派生語を部分文字列として含むか」という属性を対象にしている．単純な文字列比較では，その属性について真となる事例の数と偽となる事例の数が極端に偏っているので，この単純文字列比較の属性を用いて得られる決定木は偏ったものものになりがちである．このような問題の対処法として，属性の真偽によって分類される事例数が極端に偏らない属性を導入すればよく，派生語を用いた文字列比較の属性はこの性質をある程度満足するものといえる．

文字列 v に対して，キーワード w は,「文字列 v はキーワード w の派生語を部分文字列として含むか」という属性に対応している．つまり，キーワードを属性と同一視できる．次に，文字列 u,v が互いに派生語であるという関係を定義する．ただし，派生語であるという関係を辞書等により参照することは考えず，文字列上のマッチングにより定義する．形式的に定義すると，ある $u,v \in \Sigma^*$ に対して u,v が互いに派生語とは，

- $len(min(u,v)) < 4$ の場合

  $u = v$ のとき，

- $len(min(u,v)) = 4$ の場合

$$\exists t \in \{\varepsilon\} \cup \Sigma \cup \Sigma^2 \left[min(u,v)t = max(u,v) \right] \text{ のとき},$$

- $len(min(u,v)) > 4$ の場合，$s = min(u,v)$ とし，$s = s'a$ $(a \in \Sigma)$ となる s' に対し，

$$\exists t \in \Sigma \cup \Sigma^2 \cup \Sigma^3 \left[s't = max(u,v) \right] \text{ のとき},$$

である．ただし，$min(u,v)$ は $u, v \in \Sigma^*$ に対して短い方の文字列，$max(u,v)$ は u, v のうち $min(u,v)$ と等しくない方の文字列とする．また，$len(u)$ は $u \in \Sigma^*$ の長さとし，uv は $u, v \in \Sigma^*$ の連接とする．例えば，"book" と "books" は互いに派生語であり，"study" と "studies" も互いに派生語である．

さらに，Σ はスペース文字を含むものとし，スペース文字を表すアルファベットを Σ_s とする．**単語**とは $(\Sigma \backslash \Sigma_s)^*$ の要素のこととする．このとき，**キーワード**とは Σ^* の要素とする．**キーワード属性**とはキーワードが定める属性のことで，キーワード v に対してキーワード属性を K_v と表記する．キーワード属性は Σ^* の要素を引数とする述語であると定義でき，ある $u \in \Sigma^*$ に対して，

$$K_v(u) \text{が真} \quad \Leftrightarrow \quad \begin{array}{l} \text{「文字列 } u \text{ は } v \text{ と互いに派生語である} \\ \text{単語を部分文字列として含んでいる.」} \end{array} \tag{1}$$

のように意味付けすることができる．文字列 $v \in (\Sigma \backslash \Sigma_s)^+$ に対して，つまり，単語をキーワードとするキーワード属性は式 (1) の意味付けで問題ないが，スペース文字を含むキーワードに対しては曖昧になるのでより形式的に定義する．一般のキーワード $v \in \Sigma^*$ は，

$$v = s_0 w_1 s_1 w_2 s_2 \cdots w_{n-1} s_{n-1} w_n s_n$$
$$(\text{ただし } s_0, s_n \in \Sigma_s^*,\ s_1, \ldots, s_{n-1} \in \Sigma_s^+,\ w_1, \ldots, w_n \in (\Sigma \backslash \Sigma_s)^+)$$

という形式をしている．単語の定義から v の中の単語数は少なくとも n だが，単語になり得るもので長さにおいて極大なものを単語であるとすれば，文字列 v の中の単語数は n であるといえる．このとき，ある文字列 $u \in \Sigma^*$ に対して，

$$K_v(u) \text{が真} \quad \Leftrightarrow \quad \left\{ \begin{array}{l} \text{「文字列 } u \text{ は } v \text{ の中の } n \text{ 個の単語と互いに派生語} \\ \text{である単語を含む.」} \\ \qquad \text{かつ} \\ \text{「各単語 } w_i \text{ と互いに派生語である単語を } w_i' \text{ とし，} \\ \text{順序集合 } (\{w_i : 1 \le i \le n\}, <_v) \text{ と } (\{w_i' : 1 \le i \le \\ n\}, <_u) \text{ とが同型である．(ただし，} <_v \text{ は } v \text{ の中で} \\ \text{の } w_i \text{ の出現位置上の関係，} <_u \text{ についても同様).」} \\ \qquad \text{かつ} \\ \text{「文字列 } u \text{ 上で 各 } w_i' \text{ の間にある文字はすべて } \Sigma_s \\ \text{の要素である.」} \end{array} \right.$$

と定義できる．つまり，K_v に対して真となる文字列 u は

$$r_0(t_1r_1)^*w_1's_1'w_2's_2'\cdots s_{n-1}'w_n'(r_2t_2)^*r_3$$
$$\left(\begin{array}{c}\text{ただし，}r_1,r_2,s_1',\ldots s_n'\in\Sigma_s^+,\ r_0,r_3\in\Sigma_s^*,\ t_1,t_2\in(\Sigma\backslash\Sigma_s)^+\text{で，}\\ w_i'\text{は}w_i\text{と互いに派生語である単語}(1\leq i\leq n)\end{array}\right)$$

という形式のもののみである．例えば $v=$ “Information System” のキーワード属性 K_v に対して，$u=$ “Advanced Information Systems Engineering” は真となる．

文書データ分類はキーワードを属性とした決定木で表現される．この決定木を**文書分類木**と呼ぶ．文書分類木とは，各内部ノードにはキーワードがラベルとして，各葉には文字列が分類されるクラス名がラベルとして付けられている二分木のことである．分類したい文字列に対して，各内部ノードにラベル付けされているキーワードのキーワード属性が真なら右の子ノードへ，偽なら左の子ノードへ進むことが分類作業に対応している．つまり，文書分類木のあるパスはある文字列に対して文書分類木を用いて分類する過程に対応している．

一般に文書データは，あるアルファベット上の文字列と考えることができる．したがって，実際の文書データ分類に際して文書分類木を用いて分類作業をすることができる．ある文書データを用いた文書分類木の例を図 1 に示す．この文書分類木を用いると，“Systems and Control” というタイトルの文献は分類項目 4 に分類されることが分かる．

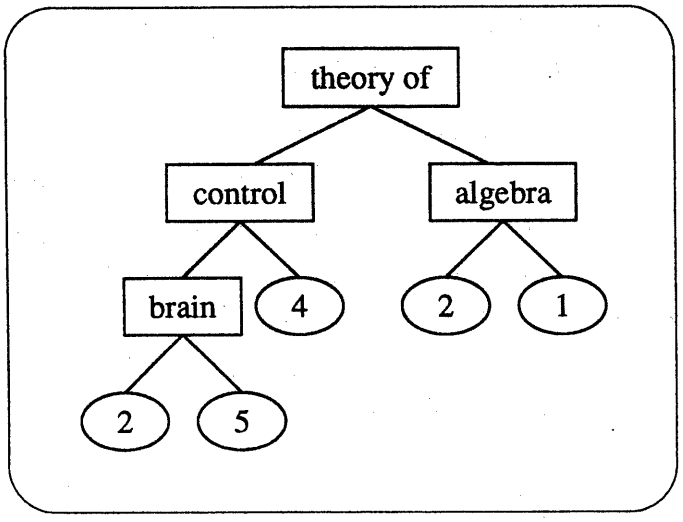

図 1: 簡単な文書分類木の例

3 文書分類木学習アルゴリズム

文書分類木を学習するアルゴリズムとは，既に分類された文書データをサンプルとして入力し文書分類木を帰納的に構成するアルゴリズムである．分類された文書とは，文献

のタイトルとその文献の属する分野 (分類項目) との組 (例えば, ⟨"Montague Grammar", LANG⟩) とする. (実際の分類作業は本のタイトルのみではなく内容を吟味して行なうものである. 文書データのサンプルとしては文献の要旨と分類項目の組からなる集合であるとしてもよいが簡単のために文献タイトルのみを利用する). ここで用いる文書分類木学習アルゴリズムに流れについて簡単に示す.

step.1. 大量の文書データの中から自動的に属性 (キーワード) を抽出し, 属性の集合を構成する.

step.2. 各属性について評価関数を利用してノードのラベルとなる属性を決定する.

step.3. 選択された属性の真偽によりサンプルを二分する. 分割されたサンプルに対して,

(a) ほぼ分類項目が均一になったら, ノードのラベルとして分類項目を書き込んだ後, そのサンプルに対しては手続きを停止する.

(b) そうでないサンプルに対しては step.2 に戻って手続きを繰り返す.

まず, step.1 であるが, 実際にはキーワード属性の候補として単純にすべての単語の並びを選ぶ. 例えば, 単語数が k である文献が m あったとすると, 高々 $m \cdot k(k-1)/2$ の数のキーワード属性を用いることになる. 例えば, "Machines Languages and Complexity" というタイトルの文献 1 つに対して,

"machines",
"languages",
"and",
"complexity",
"machines languages",
"languages and",
"and complexity",
"machines languages and",
"languages and complexity",
"machines languages and complexity"

の 10 個のキーワードを得ることになる. この方法でキーワードを得ることは多少実用面での問題点がある. それは, サンプルに対して属性数が非常に大きくなることである. しかも, あるキーワード属性が他のキーワード属性とサンプル中のすべての文書に対して同一の値をとる場合がかなりある. このことは実験的にも確かめることができる. これに対しては, [AD91] にあるように冗長な属性を調べ考慮の対象外として学習を開始するという手法もある.

step.2 で用いられる評価関数を以下に示す. S をサンプル, $v \in \Sigma^*$ をキーワードとする. このとき,

$$
\begin{aligned}
S_v^1 &= \{(u,\ell) \in S : K_v(u) \text{が真}\} \\
S_v^0 &= \{(u,\ell) \in S : K_v(u) \text{が偽}\} \\
D(S,c) &= \{(u,\ell) \in S : c = \ell\}
\end{aligned}
$$

ALGORITHM *DocTree*

Input:

サンプル S,
キーワードに関するパラメータ $kl_{\min}$ と $kl_{\max}$,
ノイズに関するパラメータ $prnrt$ と $nsrt$.

Output:

サンプル S に対する文書分類木 T.

Procedure:

1. キーワードをサンプル S から抽出．キーワードの集合 $\mathcal{K}$ を以下に示す．

$$\mathcal{K} = \{v : kl_{\min} \leq v\text{の中の単語数} \leq kl_{\max},\ v\text{は}u((u,l) \in S)\text{に対する部分単語列}\}$$

2. 副手続き $CalcTree(S, \mathcal{K}, prnrt, itnsrt)$ を呼び出し文書分類木 T を計算し，出力する．

SubProcedure $CalcTree(X, \mathcal{K}, prnrt, itnsrt)$:

1. ある ℓ_i が以下の条件を満たすとき，$T = \ell_i$ を出力し停止する．

$$\frac{|X| - |D(X, \ell_i)|}{|X|} \leq nsrt$$

2. $|X| \leq prnrt$ であるとき，

$$|D(X, \ell_k)| = \max\{|D(X, \ell_i)| : 1 \leq i \leq m\}$$

を満たす $T = \ell_k$ を出力し停止する．

3. (a) X に対して informative なすべてのキーワードについて，評価関数 $Eval(v, X)$ を計算する．
   (b) $Eval(v, X)$ を最小にする最長のキーワードを選択し v_c とする．
   (c) キーワード属性 K_{v_c} によって X を二分し，

$$T_0 = CalcTree(X^0_{v_c}, \mathcal{K}, prnrt, itnsrt)$$
$$T_1 = CalcTree(X^1_{v_c}, \mathcal{K}, prnrt, itnsrt)$$

   を計算する．
   (d) v_c を決定木の根のラベル，T_0 を左部分木，T_1 を右部分木，とした決定木 T を出力し停止する．

図 2: 文書分類木の学習アルゴリズム

とする．$S_v^1 \neq \varnothing$ かつ $S_v^0 \neq \varnothing$ であるとき K_v は **informative** であるという．今，分類項目が m 個あるとし，それぞれの分類項目名を $\ell_1, \ell_2, \ldots, \ell_m$ とする．X を事例の有限集合としたとき，ここで用いる評価関数 $Eval$ を，

$$J(X) = \varphi\left(\frac{|D(X,\ell_1)|}{|X|}, \frac{|D(X,\ell_2)|}{|X|}, \ldots, \frac{|D(X,\ell_m)|}{|X|}\right)$$

の形をした関数を用いて，

$$Eval(v,X) = \frac{|X_v^0|}{|X|}J(X_v^0) + \frac{|X_v^1|}{|X|}J(X_v^1)$$

と定義する．ただし，$\varphi(x_1, x_2, \ldots, x_m)$ の条件として，

1. $\sum_{i=1}^{m} x_i = 1, \quad 0 \leq x_i \leq 1 (1 \leq i \leq m)$,
2. 任意の $m-2$ の変数を固定して残りの 2 変数 x_i, x_j について変移させることを考える．x_i の変移を t であらわしたとき，

$$\frac{\partial^2 \varphi}{\partial t^2} \leq 0 \ ,$$

が満たされているものとする．この条件を満たす関数 $J(X)$ を **Jensen 型関数**と呼ぶ．Jensen 型関数の例として，Quinlan の ID3[Qui86] で用いられているエントロピー関数がある．そのエントロピー関数を以下に示す．

$$J(X) = -\sum_{i=1}^{m} \frac{|D(X,\ell_i)|}{|X|} \log_2 \frac{|D(X,\ell_i)|}{|X|}$$

この，評価関数を用いて属性を選択することになるが最大の評価を得る属性が唯一であるとは限らない．むしろ，実際は複数存在することが少なくない．文書分類木の学習アルゴリズムでは，最大評価を得た属性が複数ある場合は，そのキーワードの文字列長が最長のキーワード属性を選択する．例えば，“Neural Network” と “Neural” のキーワード属性がともに最大評価のとき，“Neural Network” の方を選択する．これは，数単語で一つの意味をなす言葉を抽出するのを目的としている．

step.3 では，文書データ内に含まれるノイズに対応するために，アルゴリズムの入力として与えられる 2 つのパラメータ $nsrt$ と $prnst$ を利用している [Sak91]．これらのパラメータは前者がノイズの割合，後者が枝刈のためのもので，これらの値は Valiant の PAC 学習モデル [Val84] で形式的に決定できるが，本稿ではこのことは議論しない．

図 2 に具体的にアルゴリズムを示す．

4 実験的考察

本節では，図 2 の文書分類木の学習アルゴリズムを用いた実験結果とそれに対して考察を与える．ただし，文書分類木学習アルゴリズムが用いる評価関数として Quinlan の

01#Algebraic Curves Over Finite Fields
01#Algorithmic Algebraic Number Theory
02#Automata Languages And Programming
02#Artificial Neural Networks
02#Machines Languages And Complexity
03#Montague Grammar
04#A Science Of Generic Design Volume I
04#Foundations Of Robotics
05#Dynamics Of Proteins And Nucleic Acids
07#Human Judgment

図 3: 入力サンプルの一部分

分類項目	内容
00. 参考	ハンドブック，辞典，辞書，説明書など
01. 数理科学	数学，数理科学など
02. 情報	情報全般，情報科学，計算機科学など
03. 言語科学	言語学全般，記号論，統語論など
04. システム科学	システム理論，制御工学，ロボティクスなど
05. 生物科学	生物工学，神経回路，DNA／遺伝子など
06. 人文科学	哲学，心理学，認知科学など
07. 社会科学	社会科学全般，政策，経営など
08. 環境科学	環境科学など
09. 教育	教育学，図書館学など
10. 工学全般	電気工学，電子工学，電波工学，機械工学など
11. 物理科学	物理科学全般

図 4: 文書の分類項目

エントロピー関数を用いる．実験は，某図書室にある洋書とその分類データを入力サンプルとして与えて行なった．文書データの一部分を図 3 に示す．データの形式は “分類項目#文献タイトル” となっている．また，分類項目とその内容については図 4 に示す．

文献数 566 冊を二分し 466 冊を入力サンプルとして文書分類木学習アルゴリズムに与え，残りの 100 冊でえられた文書分類木がどの程度精度があるのかを測るためのテストデータとして用いた．具体的には，以下の実験を行なった．

1. $prnst = 5$, $nsrt = 20\%$ とした場合，派生語のキーワード属性と単純なキーワード属性を用いた場合の分類精度と文書分類木の大きさの比較．
2. $prnst = 3$, $nsrt = 5\%$ とした場合，派生語のキーワード属性と単純なキーワード属性を用いた場合の分類精度と文書分類木の大きさの比較．
3. $prnst = 10$, $nsrt = 15\%$ とした場合，派生語のキーワード属性と単純なキーワード属性を用いた場合の文書分類木の大きさの比較．

実験結果を表 1 に示す．

	派生語の利用した文字列比較		通常の文字列比較	
	分類精度	文書分類木の大きさ	分類精度	文書分類木の大きさ
実験 1	73%	85	72%	87
	66%	67	67%	67
実験 2	73%	199	74%	197
実験 3	—	93	—	125

表 1: 実験結果

実験 1 に関しては 566 冊の文献を 466 冊と 100 冊 に二分する分け方を替えて 2 通りのテストデータを試した．実験 2 では，実験 1 における 1 つ目のテストデータをそのまま利用した．また，実験 3 では，文書分類木の大きさを比較するために 566 冊すべてを入力サンプルとして学習アルゴリズムに与えた．ただし，文書分類木の大きさとは決定木のノード数のこととする．

この実験結果を学習された文書分類木の正解率という点から判断すると派生語を利用したキーワード属性の導入が効果的であるとは言い難い．また，実験 1 や実験 2 の結果を見る限りでは文書分類木の大きさという点でも差異はほとんど見られない．しかし，実験 3 の結果によると文書分類木の大きさがかなり小さくなっている．実験 1, 2 と実験 3 との違いは入力サンプルの大きさなので，この差が文書分類木の大きさの差に影響していると考えられる．

また，この実験結果のように両者にあまり差異が見られなかった原因として以下のようなことが考えられる．

1. 入力サンプルとして与える文書データ数が少な過ぎる．このために，派生語を利用した属性と単純な文字列比較の属性で真偽値がことなる事例数が少ないことが考えられる．今回の実験では 500 冊前後であるが，実際には数万のデータ数が望ましい．

2. 各分野に特有な専門書が含まれていて，キーワード属性の数がサンプルの大きさに比べて多過ぎる．このために，分類規則の一般化というよりも例外処理のような状況になっているものと考えられる．

3. 文献のタイトルのみでは分類規則を学習するのには情報が少ない．

5 おわりに

単純な文字列比較から発展させて派生語を利用した文字列比較を用いる属性を利用して，洋書における分類規則を学習するアルゴリズムを考えたが，今後の課題として以下のようなものが挙げられる．

1. 現在のアルゴリズムでは，キーワード属性を大量に構成してしまうために，文書分類木を計算するのに時間がかかっているので，[AD91] などを利用して属性数を減らす方法を考える．

2. 複数の単語からなる属性を考えたが，実際にはこの属性が効果的には選択されなかった．むしろ，「単語 A と単語 B を含むか」といったような，幾つかの属性の合成属性を考える．

参考文献

[AD91] Hussein Almuallim and Thomos G. Dietterich. Learning with many irrelevant features. In *Proceedings of the 9th National Conference on Artificial Intelligence*, Vol. 2, pp. 547–552. Morgan Kaufmann, 1991.

[Kos92] Takeshi Koshiba. A note on local strategies to construct decision trees. Research Report IIAS-RR-92-14E, IIAS-SIS, FUJITSU LABORATORIES LTD., 1992.

[Qui86] J. Ross Quinlan. Induction of decision trees. *Machine Learning* **1**(1), pp. 81–106, 1986.

[RL92] Ellen Riloff and Wendy Lehnert. Classifying texts using relevancy signatures. In *Proceedings of the 10th National Conference on Artificial Intelligence*, pp. 329–334, 1992.

[Sak91] Yasubumi Sakakibara. Algorithmic learning of formal languages and decision trees. Research Report IIAS-RR-91-22E, IIAS-SIS, FUJITSU LABORATORIES LTD., 1991.

[SM92] 榊原康文, 三末和男. 決定木の学習による文書データの分類と日本語キーワードの抽出. 人工知能研究会報告書 82-1, 情報処理学会, 1992.

[Val84] Leslie G. Valiant. A theory of the learnable. *Communications of the ACM* **27**(11), pp. 1134–1142, 1984.