

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2007-213441

(P2007-213441A)

(43) 公開日 平成19年8月23日(2007.8.23)

(51) Int. Cl.	F I	テーマコード (参考)
<b>G06N 5/04 (2006.01)</b>	G06N 5/04 550N	
	G06N 5/04 550C	
	G06N 5/04 580A	

審査請求 未請求 請求項の数 15 O L (全 38 頁)

(21) 出願番号	特願2006-34343 (P2006-34343)	(71) 出願人	391041062
(22) 出願日	平成18年2月10日 (2006.2.10)		福島県
			福島県福島市杉妻町2番16号
		(74) 代理人	100118094
			弁理士 殿元 基城
		(72) 発明者	趙 強福
			会津若松市一箕町松長一丁目17番地25号 会津大学教員公舎D205

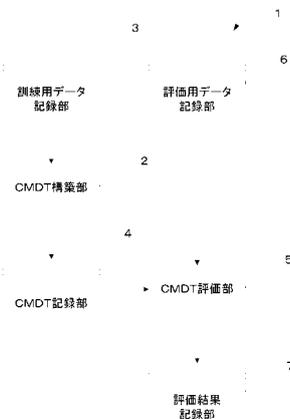
(54) 【発明の名称】 多変数決定木構築システム、多変数決定木構築方法および多変数決定木を構築するためのプログラム

(57) 【要約】

【課題】 多変数決定木 (MDT) の構築のための計算量および計算時間を短縮させることができ、さらにMDTにおける判断内容を容易に理解すること。

【解決手段】 本発明に係る多変数決定木構築システム1は、要素データを備えた複数の訓練用データを用いて、データの分割を行うための多変数テスト関数が非終端節点毎に設けられた多変数決定木を構築するシステムである。多変数決定木構築システム1は、非終端節点においてデータが分割されるべきグループを示すグループラベル情報を非終端節点毎に訓練用データに付与するグループラベル付与手段2を備えている。多変数テスト関数生成手段2はグループラベル情報に基づいて、分類データの修正を行い、修正がなされた分類データに基づいて非終端節点毎に前記多変数テスト関数を生成する。

【選択図】 図1



## 【特許請求の範囲】

## 【請求項 1】

要素データを備えた複数の訓練用データを用いて、データの分割を行うための多変数テスト関数が非終端節点毎に設けられた多変数決定木を構築する多変数決定木構築システムであって、

前記非終端節点においてデータが分割されるべきグループを示すグループラベル情報を、当該非終端節点毎に前記訓練用データに付与するグループラベル付与手段と、

前記多変数テスト関数は、前記要素データに対応するデータ情報と前記グループラベルを示すラベル情報とを有する複数の分類データからなり、前記要素データの要素数に基づいて当該要素数に対応する複数次元の特徴空間を構成し、前記訓練用データの要素データの値を前記特徴空間の空間座標として判断するとともに、前記分類データのデータ情報の値を前記特徴空間の空間座標として判断することによって、前記訓練用データの空間座標までの距離が最小となる最近傍の分類データを求め、当該訓練用データと求められた最近傍の分類データとが同一のグループラベルとなるように前記分類データの空間位置を修正することにより前記分類データのデータ情報の修正を行い、修正がなされた分類データのデータ情報とラベル情報とに基づいて前記非終端節点毎に前記多変数テスト関数を生成する多変数テスト関数生成手段と

を備えることを特徴とする多変数決定木構築システム。

## 【請求項 2】

前記訓練用データは前記多変数決定木により最終的に分割されるべきクラスを示すクラス情報を有し、

前記グループラベル付与手段は、前記クラス情報に基づいて前記訓練用データのグループラベルを決定し、当該クラス情報により前記グループラベルを決定することができない訓練用データが存在する場合には、既にグループラベルが付与された訓練用データであってグループラベルを決定することができない訓練用データに最近傍となる訓練用データと同じグループラベルを、前記グループラベルを決定できなかった訓練用データに付与することを特徴とする請求項 1 に記載の多変数決定木構築システム。

## 【請求項 3】

前記多変数テスト関数生成手段により生成された多変数テスト関数の分割性能を情報利得に基づいて判断し、当該分割性能が既定値未満である場合には当該多変数テスト関数が生成された非終端節点を終端節点に変更する早期停止判断手段

を備えることを特徴とする請求項 1 または請求項 2 に記載の多変数決定木構築システム。

## 【請求項 4】

前記訓練用データは前記多変数決定木により最終的に分割されるべきクラスを示すクラス情報を有し、

グループラベル付与手段により前記訓練用データに前記グループラベルを付与する前に、該当する節点が終端節点であるか非終端節点であるかを判断し、当該節点が終端節点である場合には当該終端節点の分割結果を前記訓練用データが有するクラス情報に基づいて決定する終端節点判別手段

を備えることを特徴とする請求項 1 ないし請求項 3 のいずれか 1 項に記載の多変数決定木構築システム。

## 【請求項 5】

前記多変数テスト関数生成手段は、生成される多変数テスト関数に含まれる分類データの数と分類データのラベル情報とが不明である場合に、該当する節点の多変数テスト関数を  $R^4$ -Rule 学習則を用いて生成する

ことを特徴とする請求項 1 ないし請求項 4 のいずれか 1 項に記載の多変数決定木構築システム。

## 【請求項 6】

要素データを備えた複数の訓練用データを用いて、データの分割を行うための多変数テ

10

20

30

40

50

スト関数が非終端節点毎に設けられた多変数決定木を構築する多変数決定木構築方法であって、

前記非終端節点においてデータが分割されるべきグループを示すグループラベル情報を、当該非終端節点毎にグループラベル付与手段が前記訓練用データに付与するグループラベル付与ステップと、

多変数テスト関数生成手段が、前記訓練用データの要素データの要素数に基づいて当該要素数に対応する複数次元の特徴空間を構成し、前記訓練用データの要素データの値を前記特徴空間の空間座標として判断するとともに、前記要素データに対応するデータ情報と前記グループラベルを示すラベル情報とを有する分類データを、当該分類データのデータ情報の値に基づいて前記特徴空間の空間座標として判断し、前記訓練用データの空間座標と前記分類データの空間座標との距離が最小となる最近傍の分類データを求め、当該訓練用データと求められた最近傍の分類データとが同一のグループラベルとなるように前記分類データの空間位置を修正することにより前記分類データのデータ情報の修正を行い、修正がなされた分類データのデータ情報とラベル情報とに基づいて前記非終端節点毎に前記多変数テスト関数を生成する多変数テスト関数生成ステップと

10

を備えることを特徴とする多変数決定木構築方法。

【請求項 7】

前記訓練用データが前記多変数決定木により最終的に分割されるべきクラスを示すクラス情報を有し、

前記グループラベル付与ステップにおいて、前記グループラベル付与手段は、前記クラス情報に基づいて前記訓練用データのグループラベルを決定し、当該クラス情報により前記グループラベルを決定することができない訓練用データが存在する場合には、既にグループラベルが付与された訓練用データであってグループラベルを決定することができない訓練用データに最近傍となる訓練用データと同じグループラベルを、前記グループラベルを決定できなかった訓練用データに付与する

20

ことを特徴とする請求項 6 に記載の多変数決定木構築方法。

【請求項 8】

早期停止判断手段が、前記多変数テスト関数生成手段により生成された多変数テスト関数の分割性能を情報利得に基づいて判断し、当該分割性能が既定値未満である場合には当該多変数テスト関数が生成された非終端節点を終端節点に変更する終端節点変更ステップ

30

を備えることを特徴とする請求項 6 または請求項 7 に記載の多変数決定木構築方法。

【請求項 9】

前記訓練用データが前記多変数決定木により最終的に分割されるべきクラスを示すクラス情報を有し、

グループラベル付与ステップにおいて前記訓練用データに前記グループラベルを付与する前に、終端節点判別手段が該当する節点が終端節点であるか非終端節点であるかを判断し、当該節点が終端節点である場合には当該終端節点の分類結果を前記訓練用データが有するクラス情報に基づいて決定する終端節点判別ステップ

を備えることを特徴とする請求項 6 ないし請求項 8 のいずれか 1 項に記載の多変数決定木構築方法。

40

【請求項 10】

前記多変数テスト関数生成ステップにおいて、生成される多変数テスト関数に含まれる分類データの数と分類データのラベル情報とが不明である場合には、前記多変数テスト関数生成手段が、該当する節点の多変数テスト関数を  $R^4$ -Rule 学習則を用いて生成することを特徴とする請求項 6 ないし請求項 9 のいずれか 1 項に記載の多変数決定木構築方法。

【請求項 11】

要素データを備えた複数の訓練用データを用いて、データの分割を行うための多変数テスト関数が非終端節点毎に設けられる多変数決定木を構築するために、コンピュータに、前記非終端節点においてデータが分割されるべきグループを示すグループラベル情報を

50

、当該非終端節点毎にグループラベル付与手段が前記訓練用データに付与するグループラベル付与ステップと、

多変数テスト関数生成手段が、前記訓練用データの前記要素データの要素数に基づいて当該要素数に対応する複数次元の特徴空間を構成し、前記訓練用データの要素データの値を前記特徴空間の空間座標として判断するとともに、前記要素データに対応するデータ情報と前記グループラベルを示すラベル情報とを有する分類データを、当該分類データのデータ情報の値に基づいて前記特徴空間の空間座標として判断し、前記訓練用データの空間座標と前記分類データの空間座標との距離が最小となる最近傍の分類データを求め、当該訓練用データと求められた最近傍の分類データとが同一のグループラベルとなるように前記分類データの空間位置を修正することにより前記分類データのデータ情報の修正を行い

10

修正がなされた分類データのデータ情報とラベル情報とに基づいて前記非終端節点毎に前記多変数テスト関数を生成する多変数テスト関数生成ステップと

【請求項 1 2】

前記訓練用データが前記多変数決定木により最終的に分割されるべきクラスを示すクラス情報を有し、

前記コンピュータに、

前記グループラベル付与ステップにおいて、前記グループラベル付与手段により前記クラス情報に基づいて前記訓練用データのグループラベルを決定させ、当該クラス情報により前記グループラベルを決定させることができない訓練用データが存在する場合には、既にグループラベルが付与された訓練用データであってグループラベルを決定することができない訓練用データに最近傍となる訓練用データと同じグループラベルを、前記グループラベルを決定できなかった訓練用データに付与させる

20

ことを特徴とする請求項 1 1 に記載の多変数決定木を構築するためのプログラム。

【請求項 1 3】

前記コンピュータに、

早期停止判断手段により前記多変数テスト関数生成手段によって生成された多変数テスト関数の分割性能を情報利得に基づいて判断させ、当該分割性能が既定値未満である場合には当該多変数テスト関数が生成された非終端節点を終端節点に変更させる終端節点変更ステップ

30

を実行させることを特徴とする請求項 1 1 または請求項 1 2 に記載の多変数決定木を構築するためのプログラム

【請求項 1 4】

前記訓練用データが前記多変数決定木により最終的に分類されるべきクラスを示すクラス情報を有し、

前記コンピュータに、

グループラベル付与ステップにおいて、前記訓練用データに前記グループラベル付与する前に、終端節点判別手段により該当する節点が終端節点であるか非終端節点であるかを判断させ、当該節点が終端節点である場合には当該終端節点の分類結果を前記訓練用データが有するクラス情報に基づいて決定させる終端節点判別ステップ

40

を実行させることを特徴とする請求項 1 1 ないし請求項 1 3 のいずれか 1 項に記載の多変数決定木を構築するためのプログラム。

【請求項 1 5】

前記コンピュータに、

前記多変数テスト関数生成ステップにおいて、生成される多変数テスト関数に含まれる分類データの数と分類データのラベル情報とが不明である場合には、前記多変数テスト関数生成手段により該当する節点の多変数テスト関数を前記  $R^4$ -Rule 学習則を用いて生成させる

ことを特徴とする請求項 1 1 ないし請求項 1 4 のいずれか 1 項に記載の多変数決定木を構築するためのプログラム。

50

## 【発明の詳細な説明】

## 【技術分野】

## 【0001】

本発明は、要素データを備えた複数の訓練用データを用いて、データの分割を行うための多変数テスト関数が非終端節点毎に設けられた多変数決定木を構築する多変数決定木構築システム、多変数決定木構築方法および多変数決定木を構築するためのプログラムに関する。

## 【背景技術】

## 【0002】

近年、コンピュータを用いた判断処理が日常的に使用されるようになってきた。コンピュータによる一般的な判断方法には、いわゆる *if - then* ルールが用いられている。多数の *if - then* ルールを効率よく、理解しやすくまとめる方法の一つとして、決定木がある。

## 【0003】

図22は、決定木(ツリー構造)の一例を示している。図22に示す決定木は決定結果(ラベル)として *Class 0* , *Class 1* を持つ終端節点(*c1 ~ c4*)と、単一変数テスト関数(UTF: Univariate Test Function)を使って局所的な分類判断(分割判断)を行う非終端節点(*a1* , *b1* , *b2*)とにより構成されている。コンピュータが何らかの判断を行う場合には、最上位にある非終端節点 *a1* (ルート)より単一テスト関数による判断に基づいて子節点(下位節点)へと順々に分類処理を進めて、最終的に終端節点における決定結果(ラベル)に基づいて判断を行う。

## 【0004】

例えば、入力データ:  $X = (0.1, 0.8)$  として、図22に示す決定木を用いて *Class 0* 又は *Class 1* の分類を行う場合を考える。まず、コンピュータは、最上位にある非終端節点 *a1* (ルート)におけるテスト関数:  $X_1 < 0.5?$  に基づく判断を行う。入力データ:  $X = (0.1, 0.8)$  より第1の  $X$  要素( $x_1$ ) =  $0.1$  は、 $0.5$  よりも小さくなるので  $x_1 < 0.5$  の条件を満たすものと判断され、ルートの下位の非終端節点 *b1* へと処理が移行する。

## 【0005】

そしてコンピュータは、非終端節点 *b1* におけるテスト関数:  $x_2 < 0.5?$  に基づく判断を行う。入力データ:  $X = (0.1, 0.8)$  より第2の  $X$  要素( $x_2$ ) =  $0.8$  は、 $0.5$  よりも大きいので、 $x_2 < 0.5?$  の条件を満たさず、非終端節点 *b1* の下位の終端節点 *c2* において決定結果として *Class 1* を備える終端節点 *c2* へと処理が移行する。コンピュータは、終端節点 *c2* において決定結果として *Class 1* を取得することにより、入力データ:  $X$  が *Class 1* に分類されるものと判断する。

## 【0006】

このように、各非終端節点で単一変数テスト関数を用いて分類(分割)処理を行うことによって、コンピュータの判断内容を *if - then* ルールで示すことができるので、処理内容が理解しやすくなると共に、判断処理の修正を簡単に行うことができるという利点がある。

## 【0007】

なお、このような単一変数テスト関数に対応する決定結果の境界は、座標軸に平行なものとなる(図23参照)ので、通常決定木は *APDT* (Axis-Parallel Decision Tree) と呼ばれる。*APDT* を構築する既存の方法として、*CART* (例えば、特許文献1参照)や *C4.5* (例えば、非特許文献2参照)等が知られている。

## 【0008】

*APDT* の構築における終端節点の判別は、通常、割り当てられたデータが全て同じクラスに属しているか、あるいは大部分のデータが既に同じクラスに属しているかによって行う。終端節点のクラスは多数決で決められる。

10

20

30

40

50

## 【 0 0 0 9 】

非終端節点におけるテスト関数を評するためには、一般的に評価関数を用いた評価が行われている。評価関数は、現在まで何種類も提案されているが、どれを使っても構築された決定木の性能はあまり変わらないことが知られている（非特許文献1）。C4.5においては、評価関数として情報利得率（IGR: Information Gain Ratio）が使用されている。

## 【 0 0 1 0 】

情報利得率は、現在節点に割り当てた訓練用データの集合を  $S$ 、そのうち  $i$  番目のクラスに属するデータの数を  $n_i$  とする。与えられたデータのクラスを識別するために必要とされる平均情報量は以下のように定義する：

## 【 数 1 】

$$Info(S) = - \sum_{i=1}^{N_c} \frac{n_i}{|S|} \times \log_2 \left( \frac{n_i}{|S|} \right)$$

・・・・・・・・（1）

ただし、 $N_c$  はクラスの数、 $|S|$  は  $S$  のサイズである。

## 【 0 0 1 1 】

あるテスト関数  $F$  を基に  $S$  を  $N$  個のグループ  $S_1, S_2, \dots, S_N$  に分割した場合、情報利得（IG: Information Gain）は次式で求められる。

$$IG(F) = Info(S) - Info(F, S)$$

・・・・・・・・（2）

ただし、

## 【 数 2 】

$$Info(F, S) = \sum_{k=1}^N \frac{|S_k|}{|S|} \times Info(S_k)$$

・・・・・・・・（3）

と定義する。情報利得（IG）もテスト関数の分割能力を評価する一つの基準であるが、情報利得を用いて決定木の分割能力を評価すると、決定木のバランスがあまりよくなることが知られている。

## 【 0 0 1 2 】

そのため、情報利得の代わりとなる評価関数として、IGR が提案されている。テスト関数  $F$  の IGR は以下の式で示される。

## 【 数 3 】

$$IGR = \frac{IG(F)}{SplitInfo(F)}$$

・・・・・・・・（4）

ただし、

10

20

30

40

【数 4】

$$SplitInfo(F) = - \sum_{k=1}^N \frac{|S_k|}{|S|} \times \log_2 \left( \frac{|S_k|}{|S|} \right).$$

..... (5)

A P D Tにおけるテスト関数は、上述のように  $X_i < a$  の形式を通常とることとなる。ここで  $X_i$  は  $i$  番目の特徴で、 $a$  は閾値を意味している。従って A P D T を構築する際にテスト関数を求めることは、評価関数を最適にするように、 $i$  と  $a$  とを求めることに等しい。この最も単純な方法は、全ての特徴とその特徴が取り得る全ての値を調べ尽くす方法である。実際、最適なテスト関数を求めるための計算量は、

$$Cost (A D P T) = O (N_d \times N_t \times m)$$

..... (6)

で示される。

【0013】

ここで  $N_d$  は特徴空間の次元（特徴の数）、 $N_t$  は現在節点に割り当てられたデータの数、 $m$  は特徴が取り得る値の数で、記号  $O()$  は「比例する」と読むことができる。最悪の場合は  $m = N_t$  である。

【0014】

A P D T は簡単に *if - then* ルールに直すことができるので、理解しやすい学習モデルとして様々な分野で応用されている。しかしながら、単一変数テスト関数を用いて判断処理を行う A P D T では、判断を行うためのデータ数が一定以上になると認識率などの性能が飽和してしまうとともに、決定木のサイズ（節点の数等）がデータ数に比例して大きくなってしまいう傾向にあった（例えば、非特許文献 3 参照）。このため、決定木のサイズが大きくなり節点数が増加すると、*if - then* ルールは非常に長くなり、理解が困難なものになってしまうという問題があった。

【0015】

一方で、決定木のサイズを減らす方法として、各非終端節点において多変数テスト関数（M T F : Multivariate Test function）を用いる方法も提案されている。多変数テスト関数を利用した決定木の中でよく知られているものが O D T（Oblique Decision Tree）である。O D T では次式に示すテスト関数が用いられている。

【数 5】

$$F(X) = \sum_{i=1}^{N_d} w_i x_i - \theta$$

..... (7)

【0016】

ここで、 $N_d$  は特徴（テスト関数において分類が行われる入力データの要素）の数、 $x_i$  は  $i$  番目の特徴、 $w_i$  は  $i$  番目の重み係数、 $\theta$  は閾値である。通常、 $F(X) < 0$  の場合、 $x$  を左子節点に割り当て、 $F(X) \geq 0$  の場合、 $x$  を右子節点に割り当てる。このような  $F(X)$  に対応する決定境界は一般の超平面となるので、A P D T よりも O D T の方が効率よくデータを分類することができる。

【0017】

O D T を構築する方法がいくつか提案されているが、その中で最も効率がよいと思われる方法は O C 1 である（例えば、非特許文献 4 参照）。O C 1 では、まず最適な U T F を

10

20

30

40

50

求め、そこから局所検索を行ってよりよいM T Fを求める。局所検索が局所最適値 (Local Optimal) におちついた場合、小さな外乱を用いてよりよい最適値を求めることによってO D Tを構築する。

【非特許文献1】L. Brieman, J. H. Friedman, R. A. Olshen and C. J. Stong, Classification and Regression Trees, Pacific Grove, CA: Wadsworth & Brooks Advanced Books and Software, 1984.

【非特許文献2】J. R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufman Publishers, 1993.

【非特許文献3】T. Oates and D. Jensen, "The effects of training set size on decision tree complexity," The 14-th International Conference on Machine Learning, pp. 254-262, 1997. 10

【非特許文献4】S. K. Murthy, S. Kasif and S. Salzber, "A system for induction of oblique decision trees," Journal of Artificial Intelligence Research, No. 2, p. 112, 1994.

【発明の開示】

【発明が解決しようとする課題】

【0018】

しかしながら、O D Tのような多変数テスト関数を利用する多変数決定木 (M D T: Multivariate Decision Tree) では、その判断方法がブラックボックス化してしまうという問題があった。例えば、(7)式に示す多変数テスト関数を用いることにより、データXが超平面の下側 ( $F(X) < 0$ ) ならばクラス0と判断し、超平面の上 ( $F(X) \geq 0$ ) ならばクラス1と分類する場合には、この分類自体は正しいものであっても、それが何を意味するかを判断することが容易ではない。 20

【0019】

さらに、多変数テスト関数を用いて決定木を構築するために、莫大な計算量が必要となるという問題があった。例えば、最も簡単な多変数決定木であるO D Tの構築であって、最適な多変数テスト関数を求める問題はNP-完全問題となり、計算量がパラメータの数に対して指数関数的に増大してしまうという問題があった。上述したO C 1の場合では、ヒューリスティックな探求法を採用することにより、テスト関数を求める計算量を

$$\text{Cost}(\text{ODT}) = O[N_d \times N_t^2 \times \log_2(N_t)] \quad 30$$

..... (8)

に減らしている。ここで $N_d$ は特徴空間の次元、 $N_t^2$ は現在節点に割り当てられたデータ数である。しかしながら、O C 1の中に確率的方法が含まれるので、計算量が非常に多くなる場合がある。また、O C 1に使われている方法は、O D Tを求めるのに提案されていたものであり、一般のM D Tの構築には使えない。

【0020】

本発明は、上記問題に鑑みてなされたものであり、決定木の構築のための計算量および計算時間を短縮させることができ、さらに決定木における判断内容を容易に理解することが可能な多変数決定木を構築することができる多変数決定木構築システム、多変数決定木構築方法および多変数決定木を構築するためのプログラムを提供することを課題とする。 40

【課題を解決するための手段】

【0021】

上記課題を解決するために、本発明に係る多変数決定木構築システムは、要素データを備えた複数の訓練用データを用いて、データの分割を行うための多変数テスト関数が非終端節点毎に設けられた多変数決定木を構築する多変数決定木構築システムであって、前記非終端節点においてデータが分割されるべきグループを示すグループラベル情報を、当該非終端節点毎に前記訓練用データに付与するグループラベル付与手段と、前記多変数テスト関数は前記要素データに対応するデータ情報と前記グループラベルを示すラベル情報とを有する複数の分類データからなり、前記要素データの要素数に基づいて当該要素数に対応する複数次元の特徴空間を構成し、前記訓練用データの要素データの値を前記特徴空間 50

の空間座標として判断するとともに、前記分類データのデータ情報の値を前記特徴空間の空間座標として判断することによって、前記訓練用データの空間座標までの距離が最小となる最近傍の分類データを求め、当該訓練用データと求められた最近傍の分類データとが同一のグループラベルとなるように前記分類データの空間位置を修正することにより前記分類データのデータ情報の修正を行い、修正がなされた分類データのデータ情報とラベル情報とに基づいて前記非終端節点毎に前記多変数テスト関数を生成する多変数テスト関数生成手段とを備えることを特徴とする。

**【0022】**

また、多変数決定木構築システムは、前記訓練用データは前記多変数決定木により最終的に分割されるべきクラスを示すクラス情報を有し、前記グループラベル付与手段は、前記クラス情報に基づいて前記訓練用データのグループラベルを決定し、当該クラス情報により前記グループラベルを決定することができない訓練用データが存在する場合には、既にグループラベルが付与された訓練用データであってグループラベルを決定することができない訓練用データに最近傍となる訓練用データと同じグループラベルを、前記グループラベルを決定できなかった訓練用データに付与することを特徴とするものであってもよい。

10

**【0023】**

さらに、多変数決定木構築システムは、前記多変数テスト関数生成手段により生成された多変数テスト関数の分割性能を情報利得に基づいて判断し、当該分割性能が既定値未満である場合には当該多変数テスト関数が生成された非終端節点を終端節点に変更する早期停止判断手段を備えることを特徴とするものであってもよい。

20

**【0024】**

また、多変数決定木構築システムは、前記訓練用データは前記多変数決定木により最終的に分割されるべきクラスを示すクラス情報を有し、グループラベル付与手段により前記訓練用データに前記グループラベルを付与する前に、該当する節点が終端節点であるか非終端節点であるかを判断し、当該節点が終端節点である場合には当該終端節点の分割結果を前記訓練用データが有するクラス情報に基づいて決定する終端節点判別手段を備えることを特徴とするものであってもよい。

**【0025】**

さらに、多変数決定木構築システムは、前記多変数テスト関数生成手段は、生成される多変数テスト関数に含まれる分類データの数と分類データのラベル情報とが不明である場合に、該当する節点の多変数テスト関数を  $R^4$ -Rule学習則を用いて生成することを特徴とするものであってもよい。

30

**【0026】**

本発明に係る多変数決定木構築方法は、要素データを備えた複数の訓練用データを用いて、データの分割を行うための多変数テスト関数が非終端節点毎に設けられた多変数決定木を構築する多変数決定木構築方法であって、前記非終端節点においてデータが分割されるべきグループを示すグループラベル情報を、当該非終端節点毎にグループラベル付与手段が前記訓練用データに付与するグループラベル付与ステップと、多変数テスト関数生成手段が、前記訓練用データの要素データの要素数に基づいて当該要素数に対応する複数次元の特徴空間を構成し、前記訓練用データの要素データの値を前記特徴空間の空間座標として判断するとともに、前記要素データに対応するデータ情報と前記グループラベルを示すラベル情報とを有する分類データを、当該分類データのデータ情報の値に基づいて前記特徴空間の空間座標として判断し、前記訓練用データの空間座標と前記分類データの空間座標との距離が最小となる最近傍の分類データを求め、当該訓練用データと求められた最近傍の分類データとが同一のグループラベルとなるように前記分類データの空間位置を修正することにより前記分類データのデータ情報の修正を行い、修正がなされた分類データのデータ情報とラベル情報とに基づいて前記非終端節点毎に前記多変数テスト関数を生成する多変数テスト関数生成ステップとを備えることを特徴とする。

40

**【0027】**

50

また、多変数決定木構築方法は、前記訓練用データが前記多変数決定木により最終的に分割されるべきクラスを示すクラス情報を有し、前記グループラベル付与ステップにおいて、前記グループラベル付与手段は、前記クラス情報に基づいて前記訓練用データのグループラベルを決定し、当該クラス情報により前記グループラベルを決定することができない訓練用データが存在する場合には、既にグループラベルが付与された訓練用データであってグループラベルを決定することができない訓練用データに最近傍となる訓練用データと同じグループラベルを、前記グループラベルを決定できなかった訓練用データに付与することを特徴とするものであってもよい。

**【0028】**

さらに、多変数決定木構築方法は、早期停止判断手段が、前記多変数テスト関数生成手段により生成された多変数テスト関数の分割性能を情報利得に基づいて判断し、当該分割性能が既定値未満である場合には当該多変数テスト関数が生成された非終端節点を終端節点に変更する終端節点変更ステップを備えるものであってもよい。

10

**【0029】**

また、多変数決定木構築方法は、前記訓練用データが前記多変数決定木により最終的に分割されるべきクラスを示すクラス情報を有し、グループラベル付与ステップにおいて前記訓練用データに前記グループラベルを付与する前に、終端節点判別手段が該当する節点が終端節点であるか非終端節点であるかを判断し、当該節点が終端節点である場合には当該終端節点の分類結果を前記訓練用データが有するクラス情報に基づいて決定する終端節点判別ステップを備えるものであってもよい。

20

**【0030】**

さらに、多変数決定木構築方法は、前記多変数テスト関数生成ステップにおいて、生成される多変数テスト関数に含まれる分類データの数と分類データのラベル情報とが不明である場合には、前記多変数テスト関数生成手段が、該当する節点の多変数テスト関数を  $R^4$ -Rule学習則を用いて生成することを特徴とするものであってもよい。

**【0031】**

本発明に係る多変数決定木を構築するためのプログラムは、要素データを備えた複数の訓練用データを用いて、データの分割を行うための多変数テスト関数が非終端節点毎に設けられる多変数決定木を構築するために、コンピュータに、前記非終端節点においてデータが分割されるべきグループを示すグループラベル情報を、当該非終端節点毎にグループラベル付与手段が前記訓練用データに付与するグループラベル付与ステップと、多変数テスト関数生成手段が、前記訓練用データの前記要素データの要素数に基づいて当該要素数に対応する複数次元の特徴空間を構成し、前記訓練用データの要素データの値を前記特徴空間の空間座標として判断するとともに、前記要素データに対応するデータ情報と前記グループラベルを示すラベル情報とを有する分類データを、当該分類データのデータ情報の値に基づいて前記特徴空間の空間座標として判断し、前記訓練用データの空間座標と前記分類データの空間座標との距離が最小となる最近傍の分類データを求め、当該訓練用データと求められた最近傍の分類データとが同一のグループラベルとなるように前記分類データの空間位置を修正することにより前記分類データのデータ情報の修正を行い、修正がなされた分類データのデータ情報とラベル情報とに基づいて前記非終端節点毎に前記多変数テスト関数を生成する多変数テスト関数生成ステップとを実行させることを特徴とする。

30

40

**【0032】**

また、多変数決定木を構築するためのプログラムは、前記訓練用データが前記多変数決定木により最終的に分割されるべきクラスを示すクラス情報を有し、前記コンピュータに、前記グループラベル付与ステップにおいて、前記グループラベル付与手段により前記クラス情報に基づいて前記訓練用データのグループラベルを決定させ、当該クラス情報により前記グループラベルを決定させることができない訓練用データが存在する場合には、既にグループラベルが付与された訓練用データであってグループラベルを決定することができない訓練用データに最近傍となる訓練用データと同じグループラベルを、前記グループラベルを決定できなかった訓練用データに付与させることを特徴とするものであってもよ

50

い。

【0033】

さらに、多変数決定木を構築するためのプログラムは、前記コンピュータに、早期停止判断手段により前記多変数テスト関数生成手段によって生成された多変数テスト関数の分割性能を情報利得に基づいて判断させ、当該分割性能が既定値未満である場合には当該多変数テスト関数が生成された非終端節点を終端節点に変更させる終端節点変更ステップを実行させることを特徴とするものであってもよい。

【0034】

また、多変数決定木を構築するためのプログラムは、前記訓練用データが前記多変数決定木により最終的に分類されるべきクラスを示すクラス情報を有し、前記コンピュータに、グループラベル付与ステップにおいて、前記訓練用データに前記グループラベル付与する前に、終端節点判別手段により該当する節点が終端節点であるか非終端節点であるかを判断させ、当該節点が終端節点である場合には当該終端節点の分類結果を前記訓練用データが有するクラス情報に基づいて決定させる終端節点判別ステップを実行させることを特徴とするものであってもよい。

10

【0035】

さらに、多変数決定木を構築するためのプログラムは、前記コンピュータに、前記多変数テスト関数生成ステップにおいて、生成される多変数テスト関数に含まれる分類データの数と分類データのラベル情報とが不明である場合には、前記多変数テスト関数生成手段により該当する節点の多変数テスト関数を前記 R<sup>4</sup>-Rule 学習則を用いて生成させることを特徴とするものであってもよい。

20

【発明の効果】

【0036】

本発明に係る多変数決定木構築システム等を用いることによって、非終端節点毎に多変数テスト関数により分類されるべきグループラベルの情報をグループラベル付与手段が各訓練用データに付与するため、非終端節点毎にグループラベルを用いて学習的に多変数テスト関数を生成することができる。このようにグループラベルを用いてテスト関数を求めることによって、テスト関数を求める問題を教師付き学習問題として帰着させることができるので、多変数決定木の構築を高速に行うことが可能となる。

【0037】

さらにグループラベルの取り得る値を適切に調整することによって、非終端節点における分割数等を調整することができるので、使用目的に適した木構造となるように多変数決定木の構築を行うことが可能である。

30

【0038】

また、本発明に係る多変数決定木構築システム等では、多変数テスト関数の分割性能を情報利得に基づいて判断し、当該分割性能が既定値未満である場合には当該多変数テスト関数が生成された非終端節点を終端節点に変更して不要節点の生成を防止するため、多変数決定木の規模が肥大化することを防止することができる。このため、構築された多変数決定木の構造が複雑になりにくく、理解しやすい決定木を構築することができると共に、決定木構築に要する処理速度の向上および処理負担の軽減を実現することが可能となる。

40

【0039】

さらに、上述した多変数テスト関数の分割性能評価は、各非終端節点において一回のみ行うので、APDTやODTのように大量のテスト関数を生成した後に全てのテスト関数に対して評価を行う場合に比べて、決定木を効率的に構築することが可能となる。

【0040】

また、データの要素データに基づく空間位置と分類データのデータ情報に基づく空間位置との距離により最適な分類データを求めて、その分類データのラベル情報に基づいてデータの分類を行うので、多変数テスト関数を用いた判断方法を容易に理解することができる、ODTのように判断方法がブラックボックス化してしまうことを回避することができる。

50

## 【0041】

さらに、多変数テスト関数に含まれる分類データの数と分類データのラベル情報とが不明な場合であっても、 $R^4$ -Rule学習則を用いて多変数テスト関数を生成することができるため、分割精度の高い多変数テスト関数を生成することが可能となる。さらに、 $R^4$ -Rule学習則を用いるか、それとも特徴空間の空間座標に基づいて最近傍の分類データを求めるLVQ学習則を用いるか、あるいはその他の学習則を用いるかは、各非終端節点において多変数テスト関数を生成する際に非終端節点毎に選択することができるため、適用される訓練用データや多変数テスト関数の条件等に応じて柔軟に多変数決定木を構築することが可能となる。

## 【発明を実施するための最良の形態】

10

## 【0042】

以下、本発明に係る本発明に係る多変数決定木構築システムを、図面を用いて説明する。図1は、多変数決定木構築システム1の概略構成を示したブロック図である。

## 【0043】

多変数決定木構築システム1は、ユーザーが理解可能な多変数決定木(CMDT: Comprehensive Multivariate Decision Tree)を構築するCMDT構築部2と、CMDTの構築に用いられる訓練用データが記録される訓練用データ記録部3と、CMDT構築部2により構築されたCMDTを記録するCMDT記録部4と、CMDT記録部4に記録されたCMDTを評価するCMDT評価部5と、CMDT評価部5での評価に用いられる評価用データが記録される評価用データ記録部6と、CMDT評価部5により評価された評価結果が記録される評価結果記録部7とを有している。

20

## 【0044】

訓練用データ記録部3、CMDT記録部4、評価用データ記録部6、評価結果記録部7はそれぞれ、メモリ、ハードディスク、フレキシブルディスク、光学記録装置(例えば、CD-ROM、DVDROM等)等のデータを記録・読み出し可能な装置で構成され、必要に応じてこれらに記録されたデータを読み出したり、書き込んだりすることが可能な構成となっている。

## 【0045】

ここで、訓練用データとは、多変数テスト関数を作成するために必要とされるデータ群であり、各データは、 $(x_1, x_2, \dots, x_n, \text{クラス})$ の形で記録される。ここで、 $x_1, x_2, \dots$ は、分類を行うために用いられる要素データであり、クラスは分類(分割)されるべき分類情報(分割情報、クラス情報)を示している。CMDT構築部2は、各データを読み取り、例えばデータの第1要素 $= x_1$ 、第2要素 $= x_2$ 、 $\dots$ 、第 $n$ 要素 $= x_n$ となる場合には、そのデータが“クラス”で示される決定結果に振り分けられる(分割される)CMDTを生成する。つまり、CMDT構築部2は、訓練用データの要素データとしての判断条件 $(x_1, x_2, \dots, x_n)$ と、これらの判断条件 $(x_1, x_2, \dots, x_n)$ に基づいて求められる判断結果(クラス)とにより、判断条件から判断結果を判断することが可能な判断基準としてCMDTを構築する。

30

## 【0046】

また、評価用データも、訓練用データと同様のデータ形式を備えるデータ群であり、CMDT構築部2により構築されたCMDTの分類(分割)精度を判断するために用いられる。評価用データも既知の要素データとクラスとを備えており、CMDT評価部5は、評価用データの要素データに基づいてCMDTによって分類(分割)された分類(分割)結果と、各評価用データのクラスとが一致するか否かを比較することによって、CMDTの評価を行う。

40

## 【0047】

訓練用データと評価用データとは、異なるデータが用いられるが、上述したように、要素データとクラスとを備える点で共通しているため、実際にCMDTの構築および評価を行う場合には、共通したデータを複数の部分に分け、一部を評価用データとして用い、残りのデータを訓練用データとして用いることによってCMDTの構築・評価が行われる。

50

## 【 0 0 4 8 】

なお、説明の便宜上、訓練用データ記録部 3 と、C M D T 記録部 4 と、評価用データ記録部 6 と、評価結果記録部 7 とを別々の記録装置として図 1 に示したが、全ての記録部または一部の記録部を、同一の記録装置によって構成してもよい。さらに、各記録部は、必ずしも物理的に C M D T 構築部 2 や C M D T 評価部 5 に繋がっている必要はなく、ネットワークを介してデータの送受信ができるような関係であってもよい。

## 【 0 0 4 9 】

C M D T 構築部 2 は、計算・処理全般を司る制御部 ( C U : C o n t r o l U n i t )、演算処理において必要なデータを一時的に記録するメモリ ( T M : T e m p o r a r y M e m o r y )、C U における演算処理をプログラムとして記録するメモリ ( P M : P r o g r a m M e m o r y ) 等を備える。なお、これらの T M や P M 等は、上述した訓練用データ記録部 2 や C M D T 記録部 4 等に用いられる記録装置と兼用するものであってもよい。

## 【 0 0 5 0 】

C M D T 構築部 2 は、図 2 に示すように、C M D T を構築する機能に応じて、終端節点判断機能 1 0 と、終端節点ラベル決定機能 1 1 と、グルーブラベル決定機能 1 2 と、C M T F 生成機能 1 3 と、早期停止判断機能 1 4 とを有しており、これらの機能を用いることによって C M D T を構築する。

## 【 0 0 5 1 】

図 3 は、C M D T 構築部 2 が C M D T を構築する過程を示したフローチャートである。C M D T 構築部 2 は、決定木を構築するために、各非終端節点に対して好適な C M T F ( 理解可能な多変数テスト関数 ) を生成し、この C M T F での判断に従って振り分けられる子節点 ( 下位節点 ) においてさらに好適な C M T F を生成して、最も下位の非終端節点まで、同様の C M T F の生成を再帰的に行うことによって、C M D T を構築する。

## 【 0 0 5 2 】

図 3 に示すように、C M D T を構築する過程において、C M D 構築部 2 は、終端節点判断機能 1 0 により C M T F を生成しようとする現在の節点が終端節点か否かを判断する ( ステップ S 1 )。終端節点であると判断した場合 ( ステップ S 1 で Y e s の場合 ) には、終端節点ラベル決定機能 1 1 により終端節点のラベルを決定し ( ステップ S 2 )、処理を終了する。

## 【 0 0 5 3 】

現在の節点が終端節点でないと判断した場合 ( ステップ S 1 で N o の場合 )、C M D T 構築部 2 は、グルーブラベル決定機能 1 2 によって、訓練用データのグルーブラベルを決定する ( ステップ S 3 )。その後、C M D T 構築部 2 は、C M T F 生成機能 1 3 によりその非終端節点における C M T F を生成する ( ステップ S 4 )。その後、C M D T 構築部 2 は、生成された C M T F の分割性能評価を早期停止判断機能 1 4 に基づいて行い ( ステップ S 5 )、分割性能が規定の評価値  $T_0$  以下であるか否かを判断することによってテスト関数の性能を評価し ( ステップ S 6 )、分割性能が評価値  $T_0$  を満たしていない場合 ( ステップ S 6 において Y e s の場合 ) には、現在の節点を終端節点に変更 ( ステップ S 7 ) し、処理を終了する。分割性能が評価値  $T_0$  を満たしている場合 ( ステップ S 6 において N o の場合 ) には、生成されたテスト関数の性能が十分なものであると判断して、C M T F によって訓練用データを複数のグループに分割し、各グループの訓練用データに基づいて新しい子節点 ( 下位節点 ) を作成し、この子節点を現在節点として上述した処理を再帰的に実行する ( ステップ S 8 )。

## 【 0 0 5 4 】

図 3 に示した C M D T の構築する過程は、単一変数テスト関数 ( U T F ) を用いて構築される通常の決定木 ( A P D T ) の構築過程にも似ている。しかしながら、A P D T を構築する際には、基本的に可能なかぎり全てのテスト関数に対して、その評価値 ( 情報利得率など ) を調べている。また O D T を構築する場合も、やはり大量のテスト関数を生成し、各テスト関数の評価を行ってその中で最もよいテスト関数を求めている。これに対して

10

20

30

40

50

、本発明に基づいてC M T Fを構築する場合には大量のテスト関数を調べる代わりに、1つのテスト関数だけを学習によって生成するため、A P D TやO D Tを構築する場合に比べて効率的に決定木を構築することが可能となる。

【0055】

次に、上述した各処理をより詳細に説明する。

【0056】

まず、終端節点判断機能における終端節点判断において、該当する節点に適用される訓練用データが全て同一クラスである場合は、訓練用データを分割する必要がないので、C M D T構築部2が現在の節点は終端節点であると判断する。現在節点が終端節点であると判断された場合、C M D T構築部2はその終端節点のラベルを訓練用データの“クラス”に設定する。この設定によって、終端節点により分類されたデータの決定結果（分類結果、分割結果）が、“クラス”に決定されることとなる。

10

【0057】

次に、現在節点が終端節点でないと判断した場合、C M D T構築部2は、グループラベル決定機能12により、訓練用データのグループラベルを決定する。C M D Tの各非終端節点にあるC M T Fは、現在節点に割り当てたデータを複数のグループ（例えばNグループ）に分割することを目的としている。このため、訓練用データに現在節点において分割されるべきグループラベルの情報（ラベル情報）を与えておく必要がある。しかしながら、訓練用データは上述したように、要素データとクラスラベルの情報とは備えているが、グループラベル情報は備えていない。このためC M D T構築部2が、グループラベル決定機能12を用いて現在節点において分割されるべきラベル情報を各訓練用データに付与する。このラベル情報は教師信号としての役割を有し、C M T Fを学習により求めるために利用されることとなる。

20

【0058】

図4は、訓練用データを2つのグループに分類（分割）するための処理を示したフローチャートである。このグループラベルを用いて生成されるC M T Fは2分木に対応するものとなる。なお、図4では説明の便宜上2つのグループに分類する方法を示しているが、分類するグループは2グループに限定されるものではなく、2以上のグループに分類する場合であっても、同様の処理を行うことによって複数のグループラベルを決定することができる。

30

【0059】

まず、C M D T構築部2は、現在節点に割り当てた訓練用データの集合Sと、現在節点の子節点（下位節点）となる左子節点と右子節点に割り当てるデータの集合 $S_1$ 、 $S_2$ とを用意する（ステップS11）。なお、集合 $S_1$ 、 $S_2$ は空集合である。

【0060】

次に、C M D T構築部2は、全ての訓練用データのクラスの中から、データ数の多い2つのクラス $C_1$ と $C_2$ を求める（ステップS12）。この $C_1$ と $C_2$ とを主要クラスと呼ぶ。そしてC M D T構築部2は、主要クラス $C_1$ を有する訓練用データを集合Sから集合 $S_1$ に移動し、主要クラス $C_2$ を有する訓練用データを集合Sから集合 $S_2$ に移動する（ステップS13）。

40

【0061】

その後、C M D T構築部2は、集合Sが空集合であるか否かの判断を行う（ステップS14）。集合Sが空集合である場合（ステップS14でYesの場合）には、訓練用データが全て2つラベルに該当する集合 $S_1$ と $S_2$ とに分類されたものと判断されるので、グループラベル決定処理を終了し、図3に示すC M T Fを生成する処理へ処理を移動する。

【0062】

集合Sが空集合でない場合（ステップS14でNoの場合）には、集合 $S_1$ と集合 $S_2$ とに分類されていない訓練用データが存在することとなるため、以下に示す処理（ステップS15～S19）を行うことによって残った訓練用データを、集合 $S_1$ か集合 $S_2$ かのどちらかに振り分ける。

50

## 【0063】

まず、C M D T構築部2は、集合Sから訓練用データを1つ取り出してそれをXとする（ステップS15）。そして、C M D T構築部2は、Xと同じクラス情報を有する訓練用データが集合S<sub>1</sub>と集合S<sub>2</sub>に移動された訓練用データの中に存在するか判断する（ステップS16）。同一のクラス情報を有する訓練用データが集合S<sub>1</sub>、S<sub>2</sub>の訓練用データから見つかった場合（ステップS16でYesの場合）、C M D T構築部2は、その訓練用データをYとする（ステップS17）。

## 【0064】

同一のクラス情報を有する訓練用データが集合S<sub>1</sub>、S<sub>2</sub>の訓練用データから見つからなかった場合（ステップS16でNoの場合）、C M D T構築部2は、集合S<sub>1</sub>、S<sub>2</sub>の訓練用データから、最近傍となる訓練用データを求めて、その訓練用データをYとする（ステップS18）。ここで、最近傍となるデータとは、訓練用データの要素データを特徴空間の空間座標として判断し、この空間座標までの距離が最も近くなるデータを意味するが、その詳細については、次述するC M T Fを生成する処理において説明する。

## 【0065】

そして、C M D T構築部2は、ステップS17またはステップS18において求められたYと同一の集合にXを移動させ（ステップS19）、以下集合Sが空集合となるまで同様の処理を繰り返す。

## 【0066】

このようにして訓練用データが集合S<sub>1</sub>とS<sub>2</sub>と割り振られた場合、集合S<sub>1</sub>に移動された訓練用データのグループラベルは例えばラベル0に決定され、集合S<sub>2</sub>に移動された訓練用データのグループラベルは、例えばラベル1に決定される。次のC M T Fを生成する処理において、C M D T構築部2は、このグループラベルを教師信号としてC M T Fの生成を行う。

## 【0067】

このようにC M D T構築部2が、非終端節点毎にC M T Fにより分類されるべきグループラベルの情報を各訓練用データに付与するため、非終端節点毎にグループラベルを用いて学習的に多変数テスト関数を生成することができる。このようにグループラベルを用いてC M T Fを求めることによって、テスト関数を求める問題を教師付き学習問題として帰着させることができるので、C M D Tの構築を高速に行うことが可能となる。

## 【0068】

さらにグループラベルの取り得る値を適切に調整することによって、非終端節点における分割数等を調整することができるので、使用目的に適した木構造となるようにC M D Tの構成を行うことが可能となる。

## 【0069】

図5は、C M D T構築部2がC M T Fを生成する過程における判断を模式的に示したブロック図である。C M D T構築部2は、C M T F生成機能13に基づいて、C M T Fを生成する方法を、L V Q学習則26、R<sup>4</sup>-Rule学習則27、その他の学習則28から節点毎に選択して、該当する節点（現在節点）におけるC M T Fを生成する。

## 【0070】

具体的に選択は、図6に示すフローチャートに基づいて行われる。C M D T構築部2は、C M T Fを生成する節点における多変数テスト関数のサイズ（規模）が固定（指定）されている場合、つまり固定型の多変数テスト関数（固定型最近傍識別器）を生成する場合（ステップS21）には、その節点におけるC M T FをL V Q学習則26により生成する（ステップS22）。

## 【0071】

C M T Fを生成する節点における多変数テスト関数のサイズ（規模）が固定（指定）されていない場合、つまり可変型の多変数テスト関数（可変型最近傍識別器）を生成する場合（ステップS23）には、C M D T構築部2は、その節点におけるC M T FをR<sup>4</sup>-Rule学習則により生成する（ステップS24）。

10

20

30

40

50

## 【 0 0 7 2 】

さらに、固定型最近傍識別器や可変型最近傍識別器に該当する多変数テスト関数とは異なるテスト関数を生成する場合には、例えば、ニューラルネットワーク、サポートベクトルマシンなどのテスト関数を使用したい場合、他の学習則を利用してその節点における C M T F を生成する（ステップ S 2 5）。

## 【 0 0 7 3 】

上述したように、どの学習則を用いて C M T F が生成されるかは、節点毎に選択することができるので、各節点に割り当てたデータの複雑さなどによって各節点の C M T F の規模を決めれば、汎用性が高く規模が小さい多変数決定木を構築することができる。

## 【 0 0 7 4 】

次に、上述した L V Q 学習則 2 6、R<sup>4</sup>-Rule 学習則 2 7 について説明する。その他の学習則 2 8 は上述したように、ニューラルネットワーク、サポートベクトルマシンなどの公知の学習則を用いるため、ここでの詳しい説明は省略する。

## 【 0 0 7 5 】

[ L V Q 学習則を用いた C M T F の生成 ]

L V Q 学習則 2 6 および R<sup>4</sup>-Rule 学習則 2 7 を用いて C M T F を生成する場合、C M D T 構築部 2 は、最近傍識別器（以下、N N C という）という多変数テスト関数を生成する。この N N C が C M T F に該当するものである。

## 【 0 0 7 6 】

「背景技術」において説明したように、多変数テスト関数を利用した決定木の中でよく知られている O D T (Oblique Decision Tree) の多変数テスト関数は ( 7 ) 式で示されるものである。このテスト関数はブラックボックス化してしまうという問題があり、分類自体が正しいものであっても、それが何を意味するか判断することは容易ではなかった。これに対して N N C は、人間らしい判断が可能な多変数テスト関数である。なお、N N C を非終端節点におけるテスト関数として用いた決定木を N N C - T r e e という。

## 【 0 0 7 7 】

まず、N N C について説明する。N N C は複数のプロトタイプ（分類データ）により構成される。プロトタイプとは、訓練用データ（入力データ）と同様の（対応する）データ形式からなるデータ情報を有している。データ情報は、特徴空間において空間座標として示すことができるデータである。また、各プロトタイプはラベル（ここでラベルとは、N N C - T r e e を構築する際におけるグループラベルを示している。クラスラベルは既知のものであるが、グループラベルはグループラベル決定機能 1 2 により各訓練用データに付与される）を備えており、この点で、プロトタイプは既知のデータであるともいえる。

## 【 0 0 7 8 】

未知のデータ X を分類する場合、C M D T 構築部 2 は、X に最も類似しているプロトタイプ Y を探し出して X を Y と同じラベルに分類する。類似するか否かの判断は、特徴空間における X と Y との距離 D によって求める。通常はユークリッド距離を用いるが、他の距離を使ってもかまわない。特徴空間の次元を N d とすると、X と Y との 2 点間のユークリッド距離 D は次の式で示される。

## 【 数 6 】

$$D = \sqrt{\sum_{i=1}^{N_d} (x_i - y_i)^2}$$

・・・・・・・・ ( 9 )

この 2 点間距離が短ければ短いほど X と Y とが類似する度合いが高いと判断できる。

## 【 0 0 7 9 】

図 7 は、( 9 ) 式により訓練用データ（入力データ）X に最適なプロトタイプ Y を求め

10

20

30

40

50

る過程を説明するために用意した図であり、理解しやすいように2次元の特徴空間を一例として示している。訓練用データ $X = (0.1, 0.8)$ とし、プロトタイプ $Y$ として $P_1 \sim P_4$ の4つの既知のプロトタイプを用いる。なお、 $P_1$ と $P_4$ とはラベル1、 $P_2$ と $P_3$ とはラベル0を備えるものとする。

#### 【0080】

まず、C M D T構築部2は、訓練用データ $X$ と全てのプロトタイプ $P_1 \sim P_4$ との距離を求める。図7から明らかなように、訓練用データ $X$ からの距離が最も近いプロトタイプ( $X$ の最近傍)は $P_1$ であるため、C M D T構築部2は、訓練用データ $X$ をプロトタイプ $P_1$ と同じラベル1に属するものと判断し、訓練用データ $X$ をラベル1に分類する。

#### 【0081】

このように、N N Cを利用したデータの分類・認識では、プロトタイプを前例として捉え、訓練用データとプロトタイプとの2点間距離に基づいてグループ(グループラベル)を判断(分類)することができる。すなわち、未知の訓練用データ $X$ が前例(プロトタイプ $Y$ )に似ていれば、訓練用データ $X$ はその前例(プロトタイプ $Y$ )と同じグループに分類されると判断することができる。従って、N N Cは「人間らしい」判断ができ、判断基準を理解しやすい多変数テスト関数であるといえる。なお、N N Cは、多数の単一テスト関数(U T F)の集まりに相当するので、非終端節点においてN N Cをテスト関数として用いることによって決定木における節点数を少なくすることができ、理解しやすい決定木を構築することが可能となる。

#### 【0082】

次に、C M D T構築部2において、N N Cを生成する方法をより詳細に説明する。

#### 【0083】

まず、本実施形態においてC M D T構築部2により作成するN N Cは、予め作成されるN N Cのサイズ(N N Cに含まれるプロトタイプの数)とN N Cにおいて使用されるプロトタイプのラベルとが既知のものとする。上述したように、C M D T構築部2は、この節点において生成するN N Cが固定型最近傍識別器の場合に、L V Q学習則を選択するため、前提としてN N CのサイズとN N Cにおいて使用されるプロトタイプのラベルとが既知のものであることが望ましい。N N Cのサイズとプロトタイプのラベルとが既知のものであれば、サイズとラベルが決まっていなくても速くN N Cを構築することができる。

#### 【0084】

ただし、サイズとクラスが既知のものでない場合であっても、通常十分に大きいN N Cのサイズを仮定し、ランダムにプロトタイプのラベルを決めるか又は各ラベルに同じ数のプロトタイプを割り振る方法を用いることによってL V Q学習則26を利用することができる。このようにしてサイズを仮定し、ラベルを決定した場合であっても、訓練用データを用いてN N Cを修正(更新)することによってN N Cの精度を向上させることができる。

#### 【0085】

N N Cを修正(更新)して精度を向上させるために、C M D T構築部2は複数エポック(その節点に適用される全ての訓練用データを1回使用することを1エポックという)訓練用データを読み出してプロトタイプの修正(更新)を繰り返し実行する。C M D T構築部2は、エポック数が規定値より多くなった場合にプロトタイプの修正(更新)を終了して、N N Cの生成つまりC M T Fの生成を完了する。

#### 【0086】

また、C M D T構築部2は、各プロトタイプを修正(更新)する方法として、学習率という概念を用いて、プロトタイプの修正を行う。この学習率は通常、 $0 < \text{学習率} < 1$ の初期値を取り、更新により徐々に減少する値である。

#### 【0087】

プロトタイプの修正(更新)を行う場合、まずC M D T構築部2は、訓練用データ $X$ (訓練用データの1つ)の最近傍となるプロトタイプ $P_0$ を求め、求められたプロトタイプ

10

20

30

40

50

のラベルと訓練用データ X のラベルとを比較する。プロトタイプ P 0 のラベルと訓練用データ X のラベルとが同じである場合には、このプロトタイプ P 0 の修正（更新）を行うことなく、次の訓練用データを読み取り同様の処理を続ける。プロトタイプ P 0 のクラスと訓練用データ X のラベルとが異なる場合、C M D T 構築部 2 は、最近傍のプロトタイプ P 0 以外のプロトタイプとして、訓練用データ X のラベルと同じラベルを持つプロトタイプの中から訓練用データ X に最も近いプロトタイプ P 1 を求める。そして、C M D T 構築部 2 は、プロトタイプ P 0 とプロトタイプ P 1 とを、

$$\begin{aligned} P 0^{new} &= P 0^{old} - (X - P 0^{old}) \cdot \dots \cdot (10) \\ P 1^{new} &= P 1^{old} + (X - P 1^{old}) \cdot \dots \cdot (11) \end{aligned}$$

に修正（更新）する。なお、 $\alpha$  は  $0 < \alpha < 1$  の値を示している。

10

#### 【0088】

また、(10)式は、プロトタイプ P 0 を訓練用データ X の要素データとプロトタイプ P 0 のデータ情報との差の  $\alpha$  倍だけ訓練用データ X の空間位置より遠ざける計算式を示し、(11)式は、プロトタイプ P 1 を訓練用データ X の要素データとプロトタイプ P 1 のデータ情報との差の  $\alpha$  倍だけ訓練用データ X の空間位置に近づける計算式を示している。

#### 【0089】

このように、1つの訓練用データ X を用いて、ラベルの正しいプロトタイプ P 1 が訓練用データ X に近づくようにプロトタイプ P 1 の修正を行うと共に、ラベルの異なるプロトタイプ P 0 が訓練用データ X から遠ざかるようにプロトタイプ P 0 の修正を行うことによって、N N C の分割精度の向上を図り、さらに各プロトタイプが最適な位置に修正される速度（収束速度）を向上させる。

20

#### 【0090】

また C M D T 構築部 2 は、さらに効率よくプロトタイプの修正（更新）を行うために、全ての訓練用データに対して使用確率  $p$  を導入し、プロトタイプの修正（更新）に使用する訓練用データの使用回数の調整を行う。

#### 【0091】

具体的に C M D T 構築部 2 は、訓練用データ X の使用確率  $p(X)$  の初期値を  $p(X) = 1$  とし、訓練用データ X がそのときの N N C により正しく分類された場合（最近傍のプロトタイプのクラスが訓練用データ X のラベルと等しい場合）に、

$$p(X)^{new} = \alpha \cdot p(X)^{old} \cdot \dots \cdot (12)$$

となるように更新する。ただし、 $\alpha$  は  $0 < \alpha < 1$  の定数である。

30

#### 【0092】

プロトタイプの修正（更新）を行う場合、C M D T 構築部 2 がある訓練用データ X を用いてプロトタイプの修正（更新）を行うか否かは、使用確率  $p(X)$  の値によって決定される。 $\alpha$  は  $0 < \alpha < 1$  の定数であるため、訓練用データ X が何回も正しく認識された場合には、 $p(X)$  が非常に小さくなる。実際に C M T F の生成において C M D T 構築部 2 における処理負担の重い計算は、訓練用データとプロトタイプとの距離を求める計算である。このため、使用確率  $p$  を導入することによって、正しく認識されやすい訓練用データの使用を少なくし、正しく認識されにくい訓練用データだけに着目して距離計算を行うことによって、C M D T 構築部 2 の処理負担を軽減させて処理速度の向上を図ることが可能となる。

40

#### 【0093】

次に、フローチャートを用いて、C M D T 構築部 2 における N N C (C M T F) の生成方法を説明する。図 8 は、C M D T 構築部 2 における N N C の生成過程を示したフローチャートである。

#### 【0094】

まず C M D T 構築部 2 は、初期設定を行う（ステップ S 3 1）。C M D T 構築部 2 は、全て ( $n$  個) の訓練用データの使用率  $p(i)$  (ただし、 $i = 1, 2, 3 \dots n$ ) の初期値に 1 を代入し、さらにエポック数を示す変数  $k$  の初期値に 0 を代入する。

#### 【0095】

50

続いてC M D T構築部2は、訓練用データXの番号を示す変数iに1を代入し(ステップS32)、さらに0から1までの値を示す乱数r発生させる(ステップS33)。そして、C M D T構築部2は、i番目の訓練用データX(i)の使用確率p(i)が乱数rよりも大きいか否かの比較を行う(ステップS34)。

【0096】

乱数rと使用確率p(i)とを比較することにより、乱数rよりも値が小さい使用確率p(i)の訓練用データX(i)、つまり正しく認識されることにより値が減少してしまった使用確率p(i)の訓練用データX(i)を用いて、プロトタイプの修正(更新)を行うことを回避する。

【0097】

ここで、使用確率p(i)との比較を乱数rではなく0から1までの定数により行ってもよいが、数エポック(このフローチャートにおいてはKエポック)回だけ訓練用データX(i)を繰り返し使ってプロトタイプの修正(更新)処理を行うため、エポック毎に異なる基準で使用確率p(i)の選別を行うべく、乱数rを用いることとしている。乱数rを用いることによって、使用確率p(i)の値が小さくなってプロトタイプの修正(更新)に使用されなくなった訓練用データX(i)を、次のエポックの際に再度利用する可能性が生ずるため、プロトタイプの修正(更新)に使用される訓練用データが偏ってしまうことを防止することができる。

【0098】

i番目の訓練用データX(i)の使用確率p(i)が乱数rよりも小さい場合(ステップS34においてNoの場合)、C M D T構築部2は、プロトタイプの更新を行うことなく、変数iが訓練用データ数Nよりも小さいか否かの判断(ステップS41)へ処理を移行する。

【0099】

訓練用データX(i)の使用確率p(i)が乱数rよりも大きい場合(ステップS34においてYesの場合)、C M D T構築部2は、訓練用データX(i)の最近傍となるプロトタイプを求めて、そのプロトタイプをY(j<sub>1</sub>)とする(ステップS35)。そしてC M D T構築部2は、求められたプロトタイプY(j<sub>1</sub>)と訓練用データX(i)とのラベルが同じか否かの判断を行う(ステップS36)。

【0100】

プロトタイプY(j<sub>1</sub>)と訓練用データX(i)とのラベルが同じである場合(ステップS6においてYesの場合)、C M D T構築部2は、訓練用データX(i)の最近傍のプロトタイプにより求められるラベルが訓練用データX(i)のラベルとして最適なラベルであるため、NNCにより適正に訓練用データX(i)が分類されたものと判断し、訓練用データX(i)の使用確率p(i)に対して $\alpha \cdot p(i)$ を掛け合わせることによって(p(i) =  $\alpha \cdot p(i)$ )、使用確率p(i)をより小さい値となるように修正し(ステップS37)、次述するステップ41へ処理を進める。

【0101】

プロトタイプY(j<sub>1</sub>)と訓練用データX(i)とのラベルが異なる場合(ステップS36においてNoの場合)、C M D T構築部2は、訓練用データX(i)の最近傍のプロトタイプにより求められるラベルが訓練用データX(i)のラベルと異なるラベルであるため、NNCにより誤って訓練用データX(i)が分類されたものと判断する。そしてC M D T構築部2は、訓練用データX(i)の使用確率p(i)に1を代入する(ステップS38)。使用確率p(i)に1を代入することにより、次にこの訓練用データX(i)が使用される場合には、ステップS34においてYesと判断され、確実にプロトタイプの修正(更新)に使用されることとなる。

【0102】

その後、C M D T構築部2は、訓練用データX(i)と同じラベルを持つプロトタイプであっての最近傍となるプロトタイプを求め、そのプロトタイプをY(j<sub>2</sub>)とする(ステップS39)。そして、C M D T構築部2は、プロトタイプY(j<sub>1</sub>)とプロトタイプ

10

20

30

40

50

$Y(j_2)$ とを、

$$Y(j_1) = Y(j_1) - (X(i) - Y(j_1)) \cdot \dots \cdot (13)$$

$$Y(j_2) = Y(j_2) + (X(i) - Y(j_2)) \cdot \dots \cdot (14)$$

に修正(更新)し、NNCの判断精度の向上を図る(ステップS40)。

【0103】

そして、CMDT構築部2は、変数*i*が該当する節点において適用される訓練用データの全数*n*よりも小さいか否かの判断を行う(ステップS41)。変数*i*が*n*より小さい場合には、まだプロトタイプの修正(更新)処理に用いられていない訓練用データ*X(i)*が存在することとなるため、変数*i*の値に1を追加して( $i = i + 1$ )(ステップS42)、上述した乱数の発生処理(ステップS33)からの処理を繰り返し実行する。

10

【0104】

変数*i*が*n*より小さくない場合、CMDT構築部2は、全ての訓練用データ*X*が一通りプロトタイプの修正(更新)に使用されたものと判断できるため、1エポック分の処理が完了したものと判断する。

【0105】

そしてCMDT構築部2は、変数*k*が所定の値*K*よりも小さいか否かの判断を行う(ステップS43)。変数*K*は、上述したようにプロトタイプの修正(更新)を行ったエポック数を示すため、ステップS43では、多変数テスト関数の生成に必要とされるエポック数である*K*回だけ、プロトタイプの修正(更新)が行われたか否かの判断を行うこととなる。

20

【0106】

プロトタイプの修正(更新)回数が*K*エポック数よりも少ない場合(ステップS43でYesの場合)、CMDT構築部2は、変数*k*の値に1を追加して( $k = k + 1$ )(ステップS44)、上述した変数*i*に1を代入する処理(ステップS2)から、上述したプロトタイプの修正(更新)処理を繰り返し実行する。

【0107】

プロトタイプの修正(更新)回数が*K*エポック数に達した場合(ステップS43でNoの場合)、CMDT構築部2は、訓練用データを用いたプロトタイプの修正(更新)処理を終了する。CMDT構築部2は、これらの処理により更新が行われたNNCにおけるプロトタイプの座標位置とそのラベルとを基準として最適なラベルを求める多変数テスト関数をCMTFとして生成する。

30

【0108】

このように、LVQ学習則を用いてNNCを修正し、CMTFを生成する場合には、訓練用データにおける訓練用データ*X*の空間位置に対して最も近い位置(最近傍の位置)に存在するプロトタイプのラベル情報が、訓練データのラベルと等しくなるようにプロトタイプが修正(更新)される。このため、訓練用データを用いて繰り返し(本実施例においては*K*エポック回数)プロトタイプを修正(更新)することによって分類精度の高いプロトタイプを生成することができ、このプロトタイプに基づいて訓練用データの分類を行うCMTFを生成することによって分類精度の高い多変数テスト関数を生成することが可能となる。

40

【0109】

また、訓練用データ*X*の空間位置とプロトタイプの空間位置との距離により最適なプロトタイプを求め、そのプロトタイプのラベル情報に基づいて訓練用データ*X*の分類を行うので、多変数テスト関数を用いた判断方法を容易に理解することができ、ODTのように判断方法がブラックボックス化してしまうことを回避することができる。

【0110】

また、多変数決定木構築システム1では、各訓練用データ(訓練用データ)に対して使用確率変数を付与し、最近傍のプロトタイプ検出において検出されたプロトタイプのラベ

50

ルが訓練用データのラベルと同一であると判断された場合、つまり最近傍となるプロトタイプにより正しくグループの分類が行われた場合に、正しく判断された訓練用データの使用確率変数の値を減少させることによって、訓練用データの個別の誤判断率を求めている。このため、使用確率変数が所定値以上の訓練用データ、つまり誤判断率の高い訓練用データをより高い確率で繰り返し用いてプロトタイプのデータ情報を修正（更新）することによって、データ情報の更新に使用する訓練用データ量を減らしつつ、効率よくプロトタイプの修正（更新）を行うことができ、全ての訓練用データを複数回使用してプロトタイプの更新を行う場合に比べて処理量を減少させ、処理スピードを高めることが可能となる。

#### 【0111】

10

以上、LVQ学習則26に基づいてCMDT構築部2がCMTFを生成する方法を説明したが、CMTFを生成する方法は上述した実施形態に記載されるものに限定されるものではない。

#### 【0112】

例えば、上記した実施形態では、プロトタイプの修正（更新）を行う場合、まずCMDT構築部2が訓練用データ $X(i)$ （ $i$ 番目の訓練用データ）の最近傍となるプロトタイプ $Y(j_1)$ のラベルと訓練用データ $X(i)$ のラベルとを比較し、プロトタイプ $Y(j_1)$ のラベルと訓練用データ $X(i)$ のラベルとが異なる場合にのみ新たなプロトタイプ $Y(j_2)$ を求めて(10)式、(11)式に示すようなプロトタイプの修正（更新）を行っているが、プロトタイプの修正（更新）方法はこの方法に限定されない。

20

#### 【0113】

図9は、他のプロトタイプの修正方法を示したフローチャートである。図9に示すプロトタイプの修正方法は、図8のステップS39、ステップS40に示す処理がなくなり、ステップS36とステップS37との間にステップS45に示す処理が追加される点で相違する。

#### 【0114】

図9に示す処理では、訓練用データ $X(i)$ の最近傍となるプロトタイプ $Y(j_1)$ のラベルと訓練用データ $X(i)$ のラベルとを比較し（ステップS36）、プロトタイプ $Y(j_1)$ のラベルと訓練用データ $X(i)$ のラベルとが同じラベルの場合（ステップS36でYesの場合）に、プロトタイプ $Y(j_1)$ のデータ情報を、

30

$$Y(j_1) = Y(j_1) + \frac{(X(i) - Y(j_1))}{\dots(15)}$$

に修正する（ステップS45）。

#### 【0115】

このように、同一ラベルとなるプロトタイプ（ $j_1$ ）が訓練用データ $X(i)$ に近づくようにプロトタイプの修正を行うことによって、上述した実施形態と同様にNNCの認識（分類）精度の向上を図り、各プロトタイプが最適な位置に修正される速度（収束度）を向上させることが可能となる。

#### 【0116】

[ $R^4$ -Rule学習則を用いたCMTFの生成]

40

次に、CMDT構築部2が、 $R^4$ -Rule学習則27を用いてCMTFを生成する場合について説明する。LVQ学習則26によりCMTEを生成する方法は、NNCのサイズ（NNCに含まれるプロトタイプの数）とプロトタイプのラベルとが既知の場合に用いられている。これに対して、 $R^4$ -Rule学習則27によりCMTFを生成する方法は、NNCのサイズとプロトタイプのラベルとがわからない場合に有効なCMTF生成方法である。

#### 【0117】

$R^4$ -Rule学習則27の詳細については、発明者が発表した論文「Q. F. Zhao and T. Higuchi, "Evolutionary learning of nearest neighbor MLP," IEEE Trans. on Neural Networks, Vol. 7, pp. 762-767, 1996」に詳細に書かれている。 $R^4$ -Rule学習則によりCMTFを生成する方法では、認識(Recognition)、記憶(Remembrance)、忘却(Reduction

50

)、復習(Review)という4つの基本操作を繰り返し使用することによって、最小のNNCを自動的に構築する。 $R^4$ -Rule学習則27を用いることによって、プロトタイプの数も動的に決めることができるので、 $R^4$ -Rule学習則27によりCMTFを生成する方法は、NNCの規模に関する事前情報が全くない場合に有効である。

#### 【0118】

図10は、 $R^4$ -Rule学習則27により使用される認識(Recognition)機能21、記憶(Recall)機能22、忘却(Reduction)機能23、復習(Review)機能24という4つの基本機能(基本処理)とその処理手順を模式的に示したブロック図である。認識機能21は、NNCの性能(認識率)とNNCにおける各プロトタイプの重要度を評価するための処理を実行する。記憶機能22は、NNCの認識率が低い場合に、新しいプロトタイプを追加するための処理を実行する。忘却機能23は、NNCの性能が十分よくなった場合に、重要度の低いプロトタイプを削除するための処理を実行する。復習機能24は、NNCを改善するための処理を実行する。なお、この復習機能24には、上述したLVQ学習則26が利用されている。

10

#### 【0119】

図11は、 $R^4$ -Rule学習則27によりCMTFを生成する過程を示したフローチャートである。 $R^4$ -Rule学習則27には、学習周期という概念が用いられている。学習周期は、「認識(記憶 忘却) 復習」と定義される。ここで、ととはそれぞれ、ロジックandとロジックorのことを意味している。

#### 【0120】

$R^4$ -Rule学習則27によりCMTFを生成する場合、CMDT構築部2は、まず、学習周期数(学習周期の数)kをゼロに初期化する(ステップS51)。その後、CMDT構築部2は、認識機能21を利用して、NNCの認識率Rと各プロトタイプの重要度を求める(ステップS52)。

20

#### 【0121】

その後、CMDT構築部2は、認識率Rが予め設定されている期待値 $R_0$ よりも小さいか否かを判断する(ステップS53)。認識率Rが期待値 $R_0$ よりも小さい場合(ステップS53でYesの場合)、CMDT構築部2は、記憶機能22を利用して、認識できないデータをランダムに一つ(複数でも可)選んで、そのままプロトタイプとして用いる(ステップS54)。また、認識率Rが期待値 $R_0$ よりも大きい場合(ステップS53でNoの場合)、CMDT構築部2は、忘却機能23を利用して、重要度が最も低い(あるいは複数の)プロトタイプを削除する(ステップS55)。

30

#### 【0122】

ここで、 $R^4$ -Rule学習則27におけるプロトタイプの重要度とは、基本的にプロトタイプPが訓練用データXの最近傍となる確率を意味している。すなわち、プロトタイプPがたくさんデータの最近傍であれば、重要度が高くなる。重要度を求める方法は複数あるが、一例として、次のような方法を用いることができる。

#### 【0123】

まず、全てのプロトタイプの重要度を0(ゼロ)とする。そして、訓練用データXを一つずつ提供し、各データXに対する最近傍を求める。最近傍がプロトタイプPであり、データXとプロトタイプPとのラベルが同じ場合には、

40

$$(P)^{new} = (P)^{old} + 1 \quad \dots \dots (16)$$

とし、ラベルが異なる場合には、

$$(P)^{new} = (P)^{old} - 1 \quad \dots \dots (17)$$

として重要度を変化させることによって、プロトタイプの重要度を求める。

#### 【0124】

その後CMDT構築部2は、上述のようにして求められたプロトタイプを用い、復習機能24を利用してLVQ学習によりNNCを修正(更新)する(ステップS56)。その後、CMDT構築部2は、学習周期数kを一つ増やし( $k = k + 1$ 、ステップS57)、学習周期数kが予め規定された規定値 $N_1$ よりも小さいか否かの判断を行う(ステップS

50

58)。学習周期数  $k$  が規定値  $N_1$  よりも小さい場合（ステップ S58 で Yes の場合）には、C M D T 構築部 2 は、ステップ S52 に示した N N C の認識率  $R$  と各プロトタイプ的重要度を求める処理に移行し、以下上述した処理を学習周期数  $k$  が規定値  $N_1$  以上になるまで繰り返し実行する。学習周期数  $k$  が規定値  $N_1$  以上の場合、C M D T 構築部 2 は、 $R^4$ -Rule 学習則 27 による N N C の修正（更新）を終了し、求められたプロトタイプの座標位置とそのラベルとを基準として C M T F を求める。つまり、C M D T 構築部 2 は、プロトタイプのデータ情報（＝座標位置を示す情報）とラベル情報とを基準として最適なラベルを求める多変数テスト関数を C M T F として生成する。

#### 【0125】

以上説明したように、C M D T 構築部 2 が C M T F 生成機能 13 により C M T F を生成した後（図 3 に示すステップ S4 の後）、C M D T 構築部 2 は、早期停止判断機能 14 により、C M D T の構築の際に不要な節点が発生することを防止する処理を行う。 10

#### 【0126】

具体的に C M D T 構築部 2 は、C M D T 構築部 2 の C M T F 生成機能 13 により生成された C M T F の分割性能を評価し（図 3 のステップ S5）、評価した分割性能が一定の基準値  $T_0$  よりも小さいか否かの判断を行う（ステップ S6）。分割性能が基準値  $T_0$  以下の場合には、現在の節点をこれ以上分割することは不要であるものと判断して、C M D T 構築部 2 がこの節点を終端節点に変更して（ステップ S7）処理を終了する。分割性能が基準値  $T_0$  以上であった場合には、分割性能が高いため現在節点のテスト関数の性能が十分なものであると判断して、C M T F によって訓練用データを複数のグループに分割し、各グループの訓練用データに基づいて新しい子節点（下位節点）を作成し、この子節点に対して本処理を繰り返し実行する（ステップ S8）。このように、分割性能が低い節点を終端節点とすることによって、後にその節点から子節点を作成されることを防止することができ、不要節点の生成を抑制させて決定木のサイズが肥大化してしまうことを防止することにより、C M D T の構築効率を高めることが可能となる。 20

#### 【0127】

上記分割性能を評価する基準として、本実施形態では [背景技術] において既に説明した情報利得 (IG: Information Gain) を利用する。IG は 0 に近いとき分割性能が悪いと考えられる。例えば、2 分木の場合、128 個のデータがグループ 0 に、1 個のデータだけがグループ 1 に分割されるとする。この分割により得られた IG は 0.05 くらいしかない。このとき、グループ 1 のデータをノイズ（雑音）と判断してその後の節点における分割を停止すれば、より汎化能力の高い決定木を構築することができる。非終端節点を終端節点に変更する場合には、その終端節点のラベルをデータの多い方のラベルに決定（多数決で決定）すればよい。 30

#### 【0128】

多変数決定木構築システム 1 では、C M T F 数の分割性能を情報利得 (IG) に基づいて判断し、分割性能が基準値  $T_0$  未満である場合には、C M T F が生成された非終端節点を終端節点に変更して不要節点の生成を防止するため、C M D T の規模が肥大化することを防止することができる。このため、構築された C M D T の構造が複雑になりにくく、理解しやすい決定木を構築できると共に、決定木構築に要する処理速度の向上および処理負担の軽減を実現することが可能となる。 40

#### 【0129】

また、C M T F の分割性能評価は、各非終端節点において一回のみ行われるので、A P D T や O D T のように大量のテスト関数を生成した後に全てのテスト関数に対して評価を行う場合に比べて、決定木を効率的に構築することが可能となる。

#### 【0130】

本発明に係る多変数決定木構築システム 1 により、上述した方法を用いて C M D T を構築した場合の計算量を説明する。通常、決定木を構築する際に必要とされる計算量は各非終端節点においてテスト関数を求める計算量で計る。A P D T を構築する際に、テスト関数を求めるための計算量は既に説明したように、

$$\text{Cost (ADPT)} = O(N_d \times N_t \times m) \dots (6)$$

である。ただし、 $N_d$  は特徴空間の次元 (特徴の数)、 $N_t$  は現在節点に割り当てられたデータの数、 $m$  は特徴が取り得る値の数である。

【0131】

ODTを構築する際に、テスト関数を求めるための計算量は、

$$\text{Cost (ODT)} = O[N_d \times N_t^2 \times \log_2(N_t)] \dots (8)$$

である。ただし、 $N_d$  は特徴空間の次元、 $N_t^2$  は現在節点に割り当てられたデータ数である。

10

【0132】

本発明に係る多変数決定木構築システム1によりCMDTを構築する際に、多変数テスト関数を求めるための計算量は、全ての学習周期と全てのエポックにおける全ての訓練用データと全てのプロトタイプとの類似度 (ユークリッド距離) を求める計算量であり、合計で

$$\text{Cost (NNC - Tree)} = O(N_d \times N_t \times N_1 \times N_e \times N_p) \dots (18)$$

となる。ただし、 $N_d$  は特徴空間の次元、 $N_t$  は現在節点に割り当てたデータの数、 $N_1$  は、 $R^4$ -Rule学習則の学習周期数 (サイズ固定型NNCをテスト関数とする場合には、この項は不要となる)、 $N_e$  はLVQ学習のエポック数 ( $R^4$ -Rule学習則を使用する場合は、これは復習機能のエポック数)、 $N_p$  はNNCの最大プロトタイプ数である。

20

【0133】

本実施形態において使用されるデフォルト値として

$$N_1 = 20, N_e = 40, N_p = 10$$

を用いる。従って、サイズ可変型NNCを求めるための計算量は、

$$\text{Cost (VariableSizeNNC)} = C_1 \times O(N_d \times N_t) \dots (19)$$

となる。ただし、 $C_1 = 8000$ である。

【0134】

サイズ固定型NNCを求めるための計算量は、

$$\text{Cost (FixedSizeNNC)} = C_2 \times O(N_d \times N_t) \dots (20)$$

となる。ただし、 $C_2 = 400$ である。

30

【0135】

上述した(19)式と(6)式と(8)式とを比較すればわかるように、訓練データ数が大きい場合、本発明に係る方法でCMDTを構築する計算量は、ADPTの構築の計算量よりも低くなる可能性がある。また、上述した計算式は、図8に示すLVQ学習則 (高速LVQ学習則) を用いる場合を考慮しておらず、さらに、早期停止判断機能14により不要な節点の生成を防止する効果をも考慮していないので、本発明に係る方法でCMDTを構築する方法では、さらに計算量が少なくなる可能性が高い。

40

【0136】

実際に、いろいろなデータベース利用して得られた実験結果により、以下のことを確認することができる。

- 1) CMDTの構築はADMTを構築する場合に匹敵する速さで構築を行うことができる。
- 2) データ数が多いときには、本発明で得られるCMDTは、ADPTよりも分類精度が高い。
- 3) 本発明で得られるCMDTは、ADPTよりサイズが遥かに小さく、決定木全体を理解しやすい。
- 4) 本発明に係るCMDTを構築する方法は、既存の多変数決定木の構築方法に比べ、計

50

算量が少なく、実用性が高い。

【0137】

従って本発明に係る多変数決定木構築システム1を多くの分野、例えば、文字認識、音声認識、顔画像認識、データマイニング、テキストマイニング、医療診断、交通状況予測などの広範囲の分野に利用することにより、従来の多変数決定木の構築方法よりも、多変数テスト関数の内容を理解しやすく、さらに多変数決定木のサイズが小さく構築時間が短い多変数決定木を提供することが可能となる。

【0138】

上述したような処理過程により、C M D T構築部2で構成されたC M D Tは、C M D T記録部4に記録される。実際のシステムにおいて構築されたC M D Tを使用(応用)するためには、構築されたC M D Tの性能評価を行うことによってC M D Tの有効性を判断する必要がある。C M D T評価部5は、このC M D Tの性能評価を行う。 10

【0139】

C M D Tの性能評価を行うために、前述した評価用データが用いられる。評価用データは上述したように、訓練用データと同様のデータ形式を備えている。通常、C M D T等の学習装置を構築するためには、訓練用データと評価用データとを構成し得る全データのうち、一部を訓練用データとして用いると共に他を評価用データとして用い、その後、評価用データとして利用されたデータを次に訓練用データとして用いると共に、訓練用データとして使用されたデータを次に評価用データとして用いることによって、複数回C M D Tを構築し、各C M D Tの評価をそれぞれのデータを用いて繰り返し行うことによって全体的なC M D Tの評価を行う。このような評価方法をn-fold cross validationと呼ぶ。 20

【0140】

ここで“n-fold cross validation”の“n”は、繰り返しC M D Tを構築する回数を示しており、通常10回程度C M D Tを構築することによってC M D Tの評価を行う。10回のC M D Tを構築することにより評価を行う方法を10-fold cross validationと呼ぶ。実際の評価結果は評価用データに依存してしまうので、一回だけの評価ではC M D Tの精度がよいか悪いかの判断を行うことが困難であるため、複数回の評価を行う。

【0141】

具体的に10-fold cross validationを用いる場合には、訓練用データと評価用データとを構成し得る全データを、重複のない10個のグループにランダムに分割する(n-fold cross validationを用いる場合には、n個のグループに分割する)。そして分割されたグループのうち、1つのグループのデータを評価用データとして使用し、他のグループのデータ(評価用データ以外のデータ)を訓練用データとして使用する。そして、各グループのデータを順番に訓練用データとして用いたC M D Tを構築し、これらのC M D Tの平均性能と信頼区間などで評価を行うことにより、C M D Tにおける信頼度の評価結果を求める。 30

【0142】

図12は、本発明に係る多変数決定木構築システム1におけるC M D Tの性能評価手順を示したフローチャートである。このフローチャートでは、10個のC M D Tを構築してC M D Tの評価を行う10-fold cross validationを示している。ここで、全データを $D_1, D_2, \dots, D_{10}$ の10グループに分割したものとす。 40

【0143】

まず、C M D T評価部5が、初期値として変数iに1を代入する(ステップS61)。次にC M D T評価部5は、 $D_i$ に該当するデータを評価用データとして評価用データ記録部6に記録させ、残りのデータを訓練用データとして訓練用データ記録部3に記録させる(ステップS62)。その後、C M D T構築部2が、訓練用データ記録部3に記録される訓練用データを読み出してC M D Tを構築し、構築されたC M D TをC M D T記録部4に記録させる(ステップS63)。

【0144】

そして、C M D T評価部5が、C M D T記録部4より構築されたC M D Tを読み出すと 50

共に、評価用データ記録部 6 から評価用データを読み出して、評価用データに基づいて C M D T 構築部 2 により構築された C M D T の評価を行い、評価結果を評価結果記録部 7 に記録する (ステップ S 6 4)。その後、C M D T 評価部 5 は、変数  $i$  にさらに 1 を加え ( $i = i + 1$ ) (ステップ S 6 5)、 $i > 10$  の用件を満たすか否かの判断を行う (ステップ S 6 6)。 $i > 10$  の用件を満たす場合、C M D T 評価部 5 は、全てのグループ ( $1 \sim 10$ ) について C M D T を作成して評価を行ったものと判断し、C M D T の評価処理を終了する。 $i > 10$  の用件を満たさない場合、C M D T 評価部 5 は、用件を満たすまで繰り返し S T E P S 6 2 以降の処理を繰り返し実行する。

#### 【0145】

上記処理が終了した後、評価結果記録部 7 に記録された評価結果を参酌することによって、C M D T 構築部 2 により構築される C M D T が実用性能を満たす分類精度を備えているか否かの判断を行うことが可能となる。評価結果が十分によい結果を得ることができれば、C M D T は現実に使用に耐え得る精度を備えるものと判断することができ、評価結果が悪い場合には、データが足りないのか、パラメータが良くないのか、構築方法自体が良くないのかなどについてさらに調べることにより構築結果の精度向上を図る必要がある。

10

#### 【0146】

次に、本発明に係る多変数決定木構築システム 1 により構築された多変数決定木を用いて行われた評価結果を、具体的な実施例を提示して説明する。

#### 【実施例 1】

#### 【0147】

20

(2次元パターン分類問題)

実施例 1 に示す 2 次元パターン分類問題は、2 次元平面上の四角領域  $[0, 1]^2$  の中にある 2 次元パターンを 4 つのクラスに分類することを目的とする問題である。これらのパターンのクラスラベルはもともと図 1 3 に示す決定木によって決められている。この決定木は O D T であり、

$$L_1 : y = 1.1x$$

$$L_2 : y = -0.91x + 1$$

$$L_3 : y = 0.91x + 0.91$$

の 3 つの式で表される超平面を用いている。

#### 【0148】

30

この問題を解決するために、まず、多変数決定木構築システム 1 を用いて、N N C - T r e e をモデルとする C M D T を構築する。既知データとして上述した同領域にランダムに発生させた 2000 個のパターンデータを用いる。図 1 4 は、発生させたデータのパターンを示している。各データは、数値的に  $(x, y, label)$  の形で表すことができる。

#### 【0149】

実施例 1 では、10-fold cross validation を使用するため、まず図 1 4 に示す 2000 個のデータをランダムに 200 個ずつ、 $1, 2, \dots, 10$  に分割する。そして、上述したフローチャートに基づいて、C M D T 構築部 2 が 10 個の C M D T を構築し、その後 C M D T 評価部 5 が 10 回の評価結果を評価結果記録部 7 に記録する。表 1 は、評価結果記録部 7 に記録される評価結果に基づいて求められる C M D T の評価結果と、従来から知られている A P D T を用いた場合の評価結果とを対比して示した表である。

40

## 【表 1】

[表 1 実施例 1 の評価結果]

	評価用データに対する誤分類率 (%)	決定木の節点数	各 NNC にあるプロトタイプ数	計算時間 (秒)
CMDT	0.398 ± 0.612	7 ± 0	3 ± 0.94	2.55 ± 0.74
APDT	3.7 ± 2.5	101.4 ± 16.6	---	0.05 ± 0

10

## 【0150】

決定木の評価判断を行うための評価内容は、主に4項目で構成される。1つ目は、決定木の規模を示す節点の総数。2つ目は、決定木の汎化能力を示す評価用データに対する誤分類率。3つ目は、各非終端節点にあるNNCの規模を示す平均プロトタイプ数。4つ目は、1つの決定木を構築するため計算時間である。計算時間は使用する計算機によって変化してしまうので、計算時間の絶対値よりも、計算時間の相対的な比較によって判断を行う。表1における各評価結果は、10回の試行の平均値とその95%信頼区間に基づいて示されている。なお、APDTはC4.5で構築されたものである(全てのパラメータはデフォルト値を使用している)。

20

## 【0151】

表1からわかるように、この実施例1の問題に対しては、CMDTに比べてAPDTの規模(決定木の節点数)は14倍くらい大きく、誤差(評価用データに対する誤分類率(%))は10倍くらい大きい。APDTの構築過程では、実際に $L_1$ ,  $L_2$ ,  $L_3$ の垂直、水平線を用いて近似を判断するため、たくさんの線を使用しなければならず、無理にAPDTを構築したとしても、問題の本質を理解することは困難となってしまう。

## 【0152】

CMDTにおける構築過程を理解するために、1つの構築結果を説明する。まず最初に、ルートノードのテスト関数を求める。そのために、上述したグループラベル決定機能12を利用して、全ての訓練用データを2グループに分ける。結果として、クラス0とクラス1のデータをグループ0に分類し、クラス2とクラス3のデータをグループ1に分類する。このグループ分けを実現するNNCを $R^4$ -Ruleで求めたところ、以下のプロトタイプが得られた:

30

$$P_{11} = (0.719, 0.275, 0)$$

$$P_{12} = (0.206, 0.7421, 1)$$

## 【0153】

プロトタイプはデータの形と同じであり、同じ種類のデータの中心であると考えることができる。プロトタイプ $P_{11}$ とプロトタイプ $P_{12}$ との中間線は $L_1$ に非常に近いことが図15から判断できる。

40

## 【0154】

次に、現在節点をルートの左子節点とし、プロトタイプ $P_{11}$ に近いデータをこの子節点に割り当てる。テスト関数を求めるために、まずグループラベル決定機能を利用してデータを2グループに分ける。そして、 $R^4$ -Rule学習則を用いてNNCを求めると、プロトタイプは以下ようになる:

$$P_{21} = (0.700, 0.187, 0)$$

$$P_{22} = (0.874, 0.381, 1)$$

## 【0155】

2つのプロトタイプ $P_{21}$ ,  $P_{22}$ の中間線は、図15に示すように $L_2$ に非常に近くなる。また、クラス0とクラス1とのデータは非常にきれいに分類されているので、現在節点

50

からそれ以上子節点を作る必要はなくなる。

【0156】

次に、現在節点をルートの右節点とし、プロトタイプ  $P_{12}$  に近いデータを利用してテスト関数を求める。左節点の場合と同様に、グループラベル決定機能 12 を利用してデータを 2 グループに分ける。そして、 $R^4$  - Rule 学習則 27 を用いて  $NNC$  を求めると、プロトタイプは以下ようになる：

$$P_{31} = (0.308, 0.759, 0)$$

$$P_{32} = (0.177, 0.614, 1)$$

【0157】

2つのプロトタイプ  $P_{31}$ 、 $P_{32}$  の中間線は、図 15 に示すように  $L_3$  に非常に近くなる。これによりクラス 2 とクラス 3 のデータがきれいに分類される。 10

【0158】

以上のプロセスで構築された  $CMDT$  (構築結果) は図 16 に示すツリー構造となる。図 16 に示す各非終端節点における二重並線記号は「より似ている」という意味を示している。例えば、未知パターン  $X = (x, y)$  がプロトタイプ  $P_{11}$  よりもプロトタイプ  $P_{12}$  に似ている場合には、 $X$  はクラス 2 かクラス 3 に属するものと判断することができる。また、 $X$  がプロトタイプ  $P_{31}$  よりもプロトタイプ  $P_{32}$  に似ている場合には、 $X$  はクラス 2 に属するものと判断することができる。

【実施例 2】

【0159】

(文字認識)

California大学の機械学習データベースに、 $optdigits$  というデータベースがあり、これらのデータベースのデータサンプルを用いて 10 個のアラビア数字を認識する問題を実施例 2 で説明する。このデータベースには、5620 個の手書き文字のデータがあり、各データは 64 個の特徴量と 1 個のクラスラベルと有している。 20

【0160】

実施例 2 についても、10-fold cross validation を用いて本発明に係る多変数決定木構築システム 1 により構築された  $CMDT$  の評価を行う。表 2 は、評価結果を示した表である。比較のため、 $C4.5$  と  $OC1$  による決定木 ( $APDT$  と  $ODT$ ) の評価結果も記載している。 30

【表 2】

[表 2 実施例 2 の評価結果]

	評価用データに対する認識誤差 (%)	決定木の節点数	各 $NNC$ にあ るプロトタイプ数	計算時間 (秒)
$CMDT$	$3.3 \pm 0.9$	$19 \pm 0$	$3.53 \pm 0.33$	$16.292 \pm 2.508$
$APDT$	$9.4 \pm 2.1$	$410.0 \pm 29.4$	---	$0.964 \pm 0.061$
$ODT$	$9.4 \pm 2.8$	$112.8 \pm 8.7$	$2 \pm 0$	$716.866 \pm 366.113$

【0161】

表 2 からわかるように、本発明に係る多変数決定木構築システムで構築した  $CMDT$  は、 $C4.5$  で構築された  $APDT$  と、 $OC1$  で構築された  $ODT$  とに比べて誤差が遥かに小さく、節点数も非常に少ない。計算時間は、同じ計算機で計った結果を示しており、 $C4.5$  が一番速く、 $OC1$  が一番遅いことがわかる。 40

## 【0162】

図17は、本発明に係る多変数決定木構築システムで構築したCMDTの一例を示したものである。図17の中で、終端節点にあるのはクラス情報で、“0”から“9”までの数字である。非終端節点にあるのは、各プロトタイプのグループラベルである。例えば、ルートにあるNNCは9個のプロトタイプがあり、それぞれのグループラベルは001101001である。各節点の右上にある数字はその節点の番号である。この決定木は、全部で19個の節点があるので、10クラス問題に対しては最小の木である。評価用データに対する誤差は2.3%であり、この誤差は平均以下であってAPDTの誤差よりも断然小さいので、判断精度の高い決定木であると考えられる。

## 【実施例3】

## 【0163】

(音声認識)

California大学の機械学習データベースに、isolet (isolated letter speech recognition)という音声認識データベースがある。このデータベースは、26個の英文字に対する150人の発音を、計7797回記録したものであり、各データは617個の特徴と1個のクラスラベルを有している。この音声データを用いて分類を行う問題を実施例3では説明する。

## 【0164】

実施例3においても、10-fold cross validationを用いたCMDTの評価を行う。表3は実施例3における評価結果を示した表である。なお、この問題におけるOC1の計算量が膨大になりすぎてしまったため、評価結果を求め出すことはできなかった。このため表3には、C4.5により構築されたAPDTの評価のみを対比して記載している。

## 【表3】

[表3 実施例3の評価結果]

	評価用データ に対する認識 誤差 (%)	決定木の 節点数	各 NNC にあ るプロトタ イプ数	計算時間 (秒)
CMDT	6.7 ± 0.5	51.6 ± 0.96	2.46 ± 0.18	71.61 ± 18.25
APDT	16.3 ± 3.3	692.2 ± 20.5	---	149.163 ± 5.923

## 【0165】

表3に示すように、本発明に係る多変数決定木構築システム1で得られたCMDTは、認識誤差が小さく、節点数も少ない。また、計算時間においても、多変数決定木構築システムにより構築されたCMDTは、C4.5で構築されるAPDTよりも速いことが示されている。

## 【0166】

以上、本発明に係る多変数決定木構築システムについて図面を用いて説明したが、本発明に係る多変数決定木構築システムは、上述した実施形態に限定されるものではない。当業者であれば、特許請求の範囲に記載された範疇内において、各種の変更例または修正例に想到し得ることは明らかであり、それらについても当然に本発明の技術的範囲に属するものである。

## 【0167】

例えば、上述した実施形態では、CMDT構築部2が、終端節点ラベル決定機能11、グループラベル決定機能12、CMTF生成機能13、早期停止判断機能14等の機能を果たすこととしたが、必ずしも全ての機能を1つのCMDT構築部2だけで行う必要はなく、物理的に異なる複数の演算処理部を用いて処理を行ってもよいし、いくつかの処理を

10

20

30

40

50

1つの演算処理部でまとめることによって2～3個の演算処理部によりC M D T構築部2が構成されるものであってもよい。

【0168】

さらに、本発明は、上述した多変数決定木構築システムに限定されるものではなく、C M D T構築部における処理を実行するためのコンピュータプログラムや、その処理を実現させる多変数決定木構築法も同様に本発明に含まれるものである。

【0169】

本発明は、データの階層的分類と解析を必要とする分野において汎用性高く使用することができるため、訓練用データさえ用意することができれば、利用分野に制限されることなく、多変数決定木を簡易かつ迅速に構築することができる。このため、データ分類と解析が重要とされる、データマイニング、テキストマイニング、医療診断などの分野において効果的に本発明に係る多変数決定木構築システムを利用することができる。

10

20

【0170】

さらに、上述したC M D T構築部2に対して、図18に示すように、データ獲得部(センサー・計測機器によるデータ検出、ネット経由ダウンロードなどによるデータ取得を行う手段)30と、データが原始データ(全く加工されていない生のデータ)である場合に原始データを記録する原始データ記録部31と、原始データを処理しやすい形に変換する(使用目的に応じてデータの特徴抽出・次元圧縮などを行う)データ変換部32と、原始データにラベルを付けるラベル付与部33と、データ変換されたり、ラベルが付与されたりしたデータ(変換データ、ラベル付与データ。なお、データ変換やラベル付与により既知データとして判断することが可能となる)を記録するデータ記録部34などを加えることによって、拡張させた多変数決定木構築システム1aを構成することができる。このように拡張された多変数決定木構築システム1aでは、自ら訓練用データを収集し、取得されたデータ(原始データ等)に基づいてC M D Tを構築することが可能となる。

【0171】

また、構築されたC M D Tを利用する場合には、図19に示すように、C M D T実装変換部39により、利用状況・利用目的に応じた何らかの形式(ソフトウェア、ハードウェア、ファームウェアなど)にC M D Tを変換し、このC M D T実装変換部39で変換されたC M D Tが実装されたC M D T実装部40において、処理用データ記録部41に記録された処理用データを、C M D Tを用いてデータ処理し、処理した結果を処理結果記録部42に記録することによってC M D Tを用いたデータ処理を実現することが可能となる。ここでデータ処理とは、認識、分類、解析などを含むものをいう。このように、C M D T実装部40と、処理用データ記録部41と、処理結果記録部42とを有するにデータ処理装置43を用いることによって、本発明に係る多変数決定木構築システムで構築されたC M D Tを利用することができるが、さらに、データ処理装置43に対して、C M D T記録部4とC M D T実装変換部39とを加えることにより、C M D Tを随時更新しながらデータを処理するシステムを構築することも可能である。

30

【0172】

さらに、このデータ処理装置43に対して、図20に示すように、データ獲得部30と、原始データ記録部31と、データ変換部32と、ラベル付与部33などを加えることによって、実時間でデータを処理することが可能なシステムを提供することも可能となる。なお、このシステムにおいて、データ処理の内容によってラベル付与部33は必要とされない場合もある。

40

【0173】

さらに、図18～20において説明した構成要素と本発明に係る多変数決定木構築システムの構成要素とを加えて、図21に示すようなシステムを構成することによって、C M D Tの構築機能、データ処理機能等の全ての機能を備えるシステムを提供することも可能となるため、より汎用性の高いシステムを実現することが可能となる。

【図面の簡単な説明】

【0174】

50

【図 1】実施形態に示す多変数決定木構築システムの概略構成を示したブロック図である。

【図 2】C M D T 構築部の機能を示したブロック図である。

【図 3】C M D T 構築部が C M D T を構築する過程を示したフローチャートである。

【図 4】C M D T 構築部が訓練用データを 2 つのグループに分類する処理を示したフローチャートである。

【図 5】C M D T 構築部が C M T F を生成する過程において用いる学習則判断を示したブロック図である。

【図 6】C M D T 構築部が C M T F を生成する過程において用いる学習則判断を示したフローチャートである。

【図 7】C M D T 構築部が訓練用データに最適なプロトタイプを求める過程を説明するために用いた図である。

【図 8】C M D T 構築部が N N C を生成する過程を示したフローチャートである。

【図 9】C M D T 構築部が N N C を生成する過程を示した他のフローチャートである。

【図 10】 $R^4$ -Rule 学習則の基本機能とその処理手順を示したブロック図である。

【図 11】C M D T 構築部が  $R^4$ -Rule 学習則に基づいて C M T F を生成する過程を示したフローチャートである。

【図 12】多変数決定木構築システムにおける C M D T の性能評価手順を示したフローチャートである。

【図 13】実施例 1 における決定木の構成を示した図である。

【図 14】実施例 1 において用いられる 2 0 0 0 個のパターンデータを座標位置によって示した図である。

【図 15】実施例 1 における各プロトタイプと境界をなす超平面との関係を示した図である。

【図 16】実施例 1 において構築された C M D T のツリー構造を示す図である。

【図 17】実施例 2 において構築された C M D T のツリー構造を示した図である。

【図 18】本発明に係る多変数決定木構築システムを拡張したシステムの概略構成を示したブロック図である。

【図 19】データ処理装置と C M D T 記録部と C M D T 実装変換部とを示したブロック図である。

【図 20】本発明に係る多変数決定木構築システムを拡張した第 1 のシステムの概略構成を示したブロック図である。

【図 21】本発明に係る多変数決定木構築システムを拡張した第 2 のシステムの概略構成を示したブロック図である。

【図 22】一般的な i f - t h e n ルールに基づいて判断がなされる決定木の構造を示した図である。

【図 23】図 2 3 に示した決定木における決定境界を 2 次元の平面により示した図である。

【符号の説明】

【0 1 7 5】

1 ... 多変数決定木構築システム

2 ... C M D T 構築部 ( グループラベル付与手段、多変数テスト関数生成手段、早期停止判断手段、終端節点判別手段、コンピュータ )

3 ... 訓練用データ記録部

4 ... C M D T 記録部

5 ... C M D T 評価部

6 ... 評価用データ記録部

7 ... 評価結果記録部

1 0 ... 終端節点判断機能 ( 終端節点判別手段 )

1 1 ... 終端節点ラベル決定機能 ( 終端節点判別手段 )

10

20

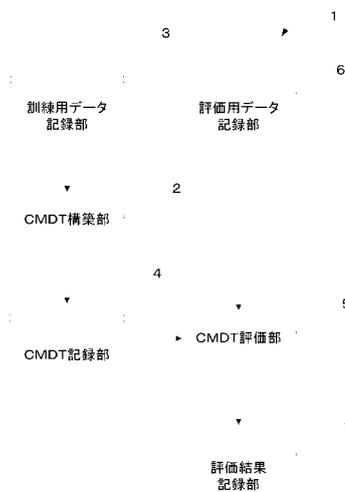
30

40

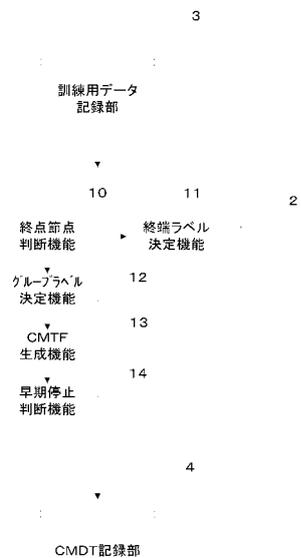
50

1 2	... グループラベル決定機能 ( グループラベル付与手段 )	
1 3	... C M T F 生成機能 ( 多変数テスト関数生成手段 )	
1 4	... 早期停止判断機能 ( 早期停止判断手段 )	
2 1	... 認識機能	
2 2	... 記憶機能	
2 3	... 忘却機能	
2 4	... 復習機能	
2 6	... L V Q 学習則	
2 7	... R <sup>4</sup> - Rule 学習則	
2 8	... その他の学習則	10
3 0	... データ獲得部	
3 1	... 原始データ記録部	
3 2	... データ変換部	
3 3	... ラベル付与部	
3 4	... データ記録部	
3 9	... C M D T 実装変換部	
4 0	... C M D T 実装部	
4 1	... 処理用データ記録部	
4 2	... 処理結果記録部	
4 3	... データ処理装置	20

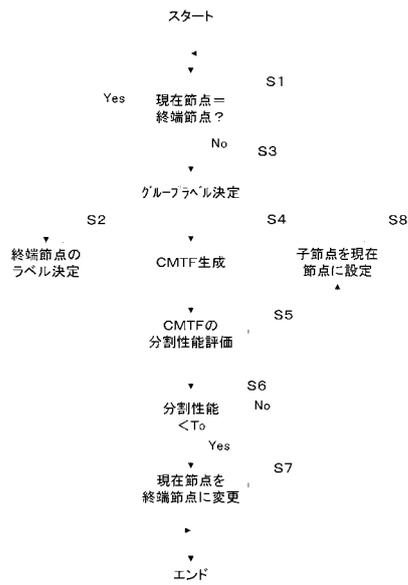
【 図 1 】



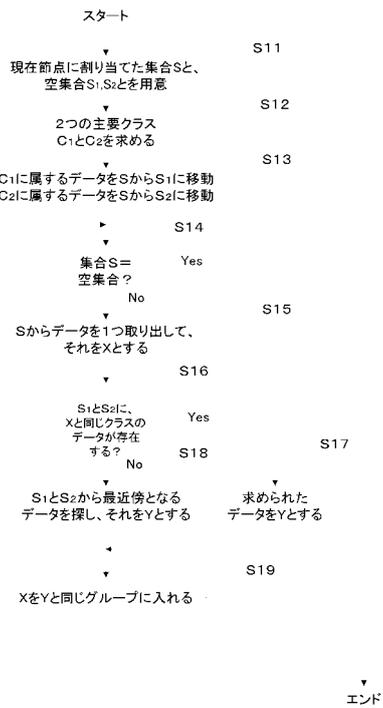
【 図 2 】



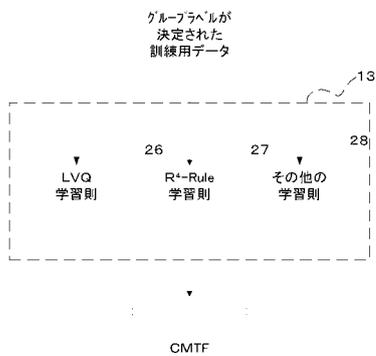
【 図 3 】



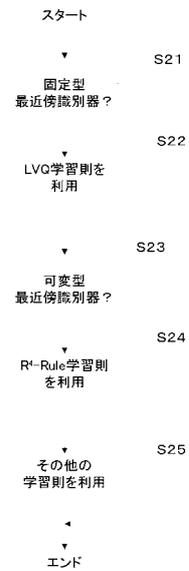
【 図 4 】



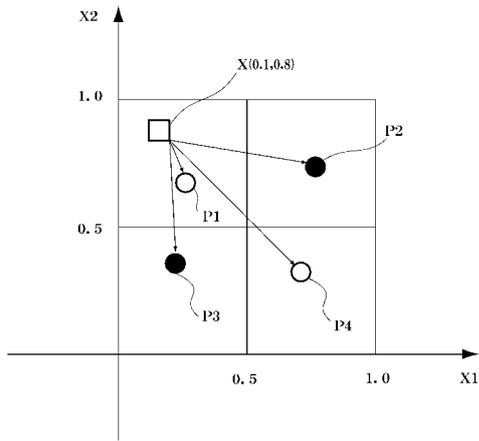
【 図 5 】



【 図 6 】



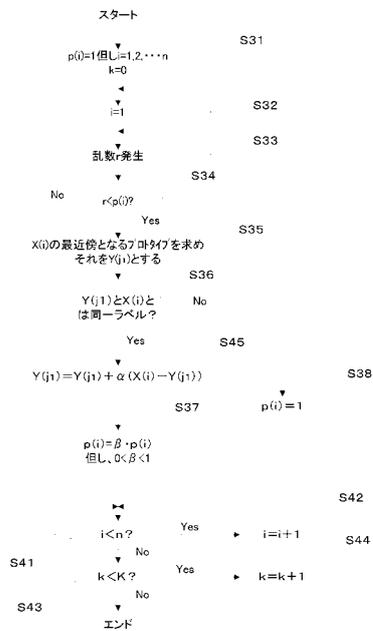
【 図 7 】



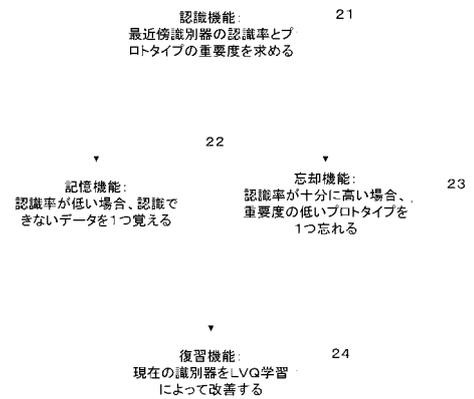
【 図 8 】



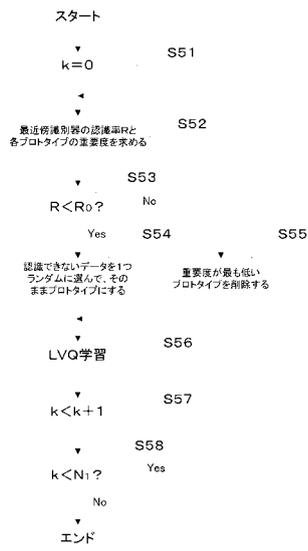
【 図 9 】



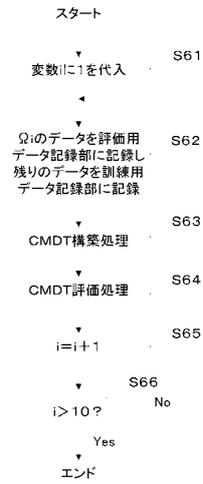
【 図 10 】



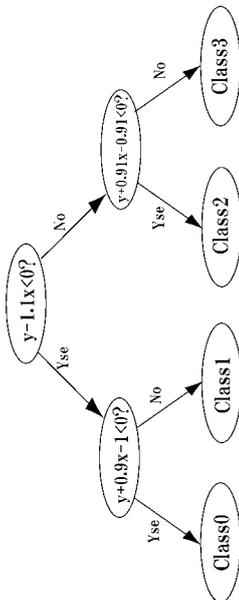
【 図 1 1 】



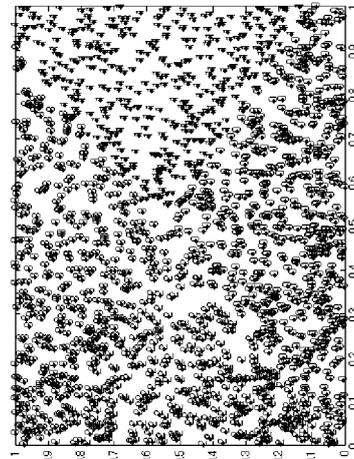
【 図 1 2 】



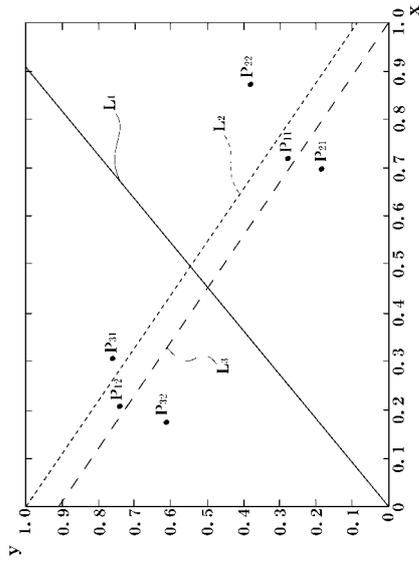
【 図 1 3 】



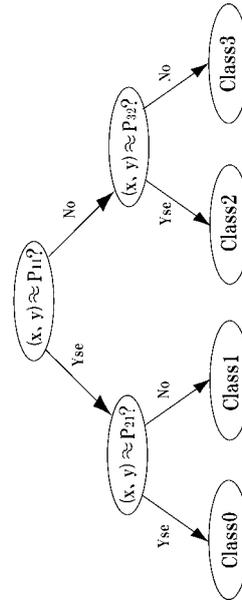
【 図 1 4 】



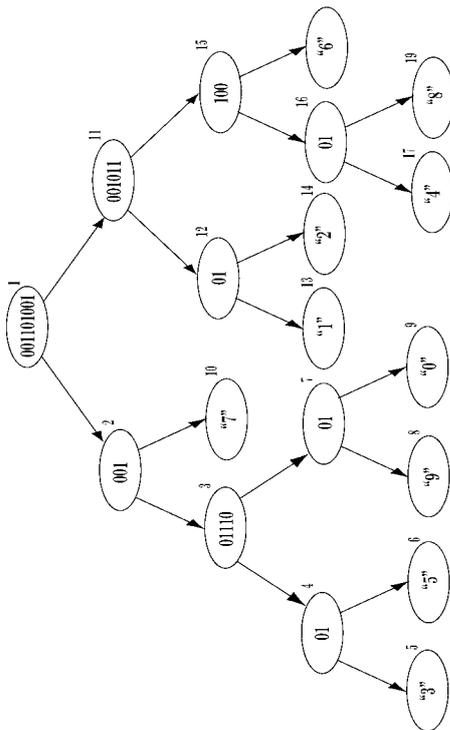
【 図 15 】



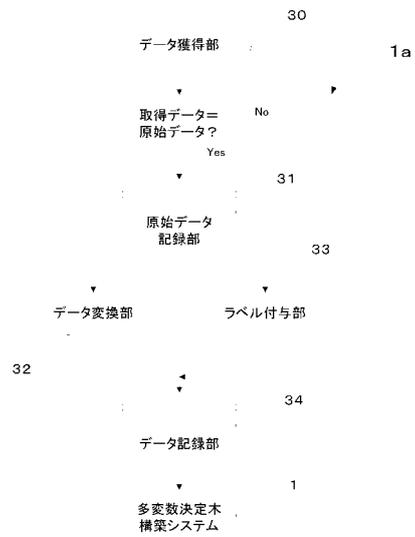
【 図 16 】



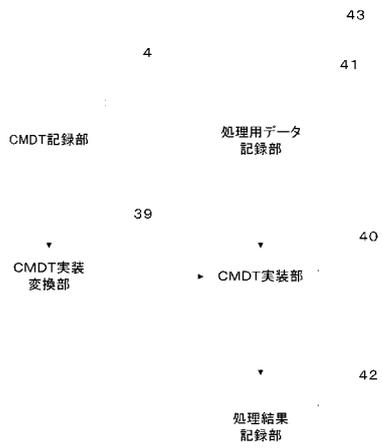
【 図 17 】



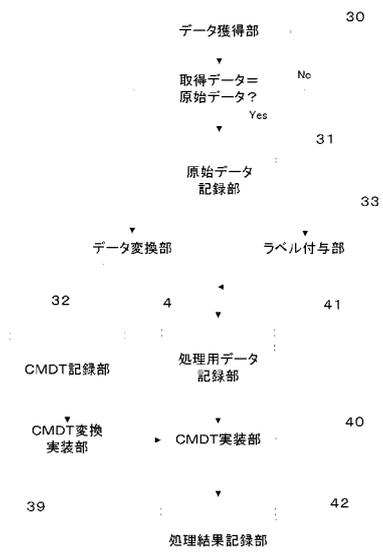
【 図 18 】



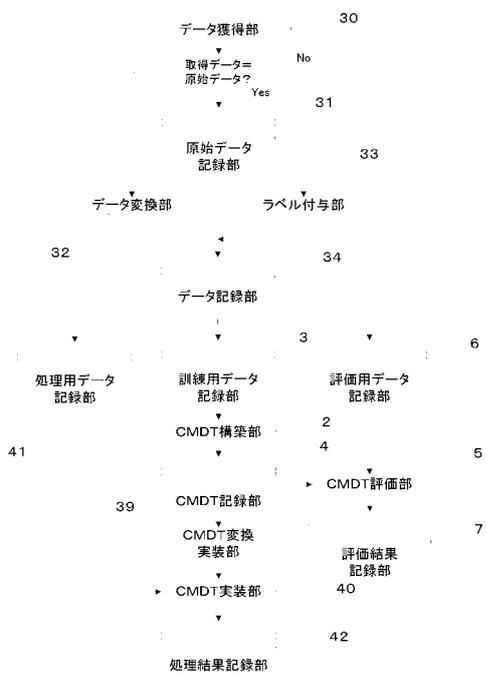
【 図 1 9 】



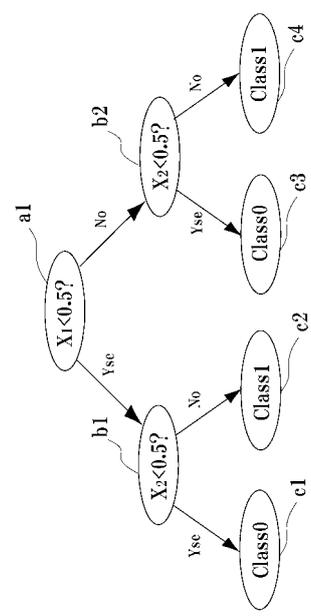
【 図 2 0 】



【 図 2 1 】



【 図 2 2 】



【 图 2 3 】

