(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第4997524号
(P4997524)

(45)発行日　平成24年8月8日(2012.8.8)　(24)登録日　平成24年5月25日(2012.5.25)

(51)Int.Cl.　F I
G06N 3/00 (2006.01)　G06N 3/00 560A
G06N 5/04 (2006.01)　G06N 5/04 550C

請求項の数 15　(全 39 頁)

(21)出願番号　特願2006-34343(P2006-34343)
(22)出願日　平成18年2月10日(2006.2.10)
(65)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)
審査請求日　平成21年1月7日(2009.1.7)

(73)特許権者　506301140
公立大学法人会津大学
福島県会津若松市一箕町大字鶴賀字上居合９０番地
(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

審査官　稲垣　良一

最終頁に続く

(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)【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のラベル情報とを有する複数の分類データからなり、
前記多変数決定木構築システムは、
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の情報を、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、前記訓練用データと求められた前記最近傍の分類データとが同一のグループラベルでない場合には、当該最近傍の分類データの空間座標を前記訓練用データの空間座標から遠ざけるように修正し、さらに、前記訓練用データと同一のグループラベルとなる分類データのうち最近傍となる分類データを求めて当該分類データの空間座標を前記訓練用データの空間座標に近づけるように修正することによって、最近傍の分類データが前記訓練用データと

同一のグループラベルになるまで前記分類データの空間座標の修正を繰り返すことにより前記分類データのデータ情報の修正を行い、最近傍の分類データが前記訓練用データと同一のグループラベルになるまで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、

前記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の情報を、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、前記訓練用データと求められた前記最近傍の分類データとが同一のグループラベルでない場合

には、当該最近傍の分類データの空間座標を前記訓練用データの空間座標から遠ざけるように修正し、さらに、前記訓練用データと同一のグループラベルとなる分類データのうち最近傍となる分類データを求めて当該分類データの空間座標を前記訓練用データの空間座標に近づけるように修正することによって、最近傍の分類データが前記訓練用データと同一のグループラベルになるまで前記分類データの空間座標の修正を繰り返すことにより前記分類データのデータ情報の修正を行い、最近傍の分類データが前記訓練用データと同一のグループラベルになるまで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

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

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、

前記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。

【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の情報を、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、前記訓練用データと求められた前記最近傍の分類データとが同一のグループラベルでない場合には、当該最近傍の分類データの空間座標を前記訓練用データの空間座標から遠ざけるように修正し、さらに、前記訓練用データと同一のグループラベルとなる分類データのうち最近傍となる分類データを求めて当該分類データの空間座標を前記訓練用データの空間座標に近づけるように修正することによって、最近傍の分類データが前記訓練用データと同一のグループラベルになるまで前記分類データの空間座標の修正を繰り返すことにより前記分類データのデータ情報の修正を行い、最近傍の分類データが前記訓練用データと同一のグループラベルになるまで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多変数決定木を構築するためのプログラム。

【請求項１２】

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

ことを特徴とする請求項１１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１３】

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請求項１１または請求項１２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１４】

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、

前記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請求項１１ないし請求項１３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１５】

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１１ないし請求項１４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。

【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Ｃｌａｓｓ０，Ｃｌａｓｓ１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Ｃｌａｓｓ０又はＣｌａｓｓ１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：Ｘ１＜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であってテスト関数：Ｘ１＜０．５を満たす場合に次の判断が求められる非終端節点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の下位の終端節点であって決定結果としてＣｌａｓｓ１を備える終端節点Ｃ２へ処理が移行する。コンピュータは、終端節点ｃ２において決定結果としてＣｌａｓｓ１を取得することにより、入力データ：ＸがＣｌａｓｓ１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、通常の決定木はＡＰＤＴ(Axis-Parallel Decision Tree)とも呼ばれる。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。

【０００９】
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。
【００１０】
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、Ｎｃはクラスの数、｜Ｓ｜はＳのサイズである。
【００１１】
あるテスト関数Ｆを基にＳをＮ個のグループ$Ｓ_1$，$Ｓ_2$，・・・$Ｓ_N$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らないことが知られている。
【００１２】
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、

【数４】

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

・・・・・（５）

ＡＰＤＴ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ＡＰＤＴ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（ＡＤＰＴ）＝Ｏ（Ｎ$_d$×Ｎ$_t$×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】

ＡＰＤＴ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ＡＰＤＴ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ルールは非常な長くなり、理解が困難なものと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（ＭＴＦ：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ＯＤＴ（Oblique Decision Tree）である。ＯＤＴ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、θ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）≧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、ＡＰＤＴよりもＯＤＴの方が効率よくデータを分類することができる。

【００１７】

ＯＤＴ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ＯＣ１である（例えば、非特許文献４参照）。ＯＣ１では、まず最適なＵＴＦを

求め、そこから局所検索を行ってよりよいＭＴＦ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ＯＤＴ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 Kauffman 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.
【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, pp. 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、ＯＤＴのような多変数テスト関数を利用する多変数決定木（ＭＤＴ：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）≧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。

【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ＯＤＴ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ＯＣ１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

$$Cost(ODT)=O[N_d \times N_t^2 \times \log_2(N_t)] \quad \cdots\cdots(8)$$

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、ＯＣ１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、ＯＣ１に使われている方法は、ＯＤＴを求めるのに提案されていたものであり、一般のＭＤＴ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。

【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

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。

【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。

【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。

【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備えることを特徴とする。

【００２７】

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。

【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。

【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。

【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

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。

【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^4-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。

【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。

【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。

【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、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。

【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点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】

【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: Comprehensible 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。

【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、（ｘ１，ｘ２、・・・ｘｎ、クラス）の形で記録される。ここで、ｘ１、ｘ２・・・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＝ｘ１，第２要素＝ｘ２、・・・、第ｎ要素＝ｘｎ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（ｘ１，ｘ２、・・・ｘｎ）と、これらの判断条件（ｘ１，ｘ２、・・・ｘｎ）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。

【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。

【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。

【００４８】
なお、説明の便宜上、訓練用データ記録部３と、ＣＭＤＴ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ＣＭＤＴ構築部２やＣＭＤＴ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。
【００４９】
ＣＭＤＴ構築部２は、計算・処理全般を司る制御部（ＣＵ：Ｃｏｎｔｒｏｌ　Ｕｎｉｔ）、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（ＴＭ：Ｔｅｍｐｏｒａｒｙ　Ｍｅｍｏｒｙ）、ＣＵにおける演算処理をプログラムとして記録するメモリ（ＰＭ：Ｐｒｏｇｒａｍ　Ｍｅｍｏｒｙ）等を備える。なお、これらのＴＭやＰＭ等は、上述した訓練用データ記録部２やＣＭＤＴ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。
【００５０】
ＣＭＤＴ構築部２は、図２に示すように、ＣＭＤＴを構築する機能に応じて、終端節点判断機能１０と、終端節点ラベル決定機能１１と、グループラベル決定機能１２と、ＣＭＴＦ生成機能１３と、早期停止判断機能１４とを有しており、これらの機能を用いることによってＣＭＤＴを構築する。
【００５１】
図３は、ＣＭＤＴ構築部２がＣＭＤＴを構築する過程を示したフローチャートである。ＣＭＤＴ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ＣＭＴＦ（理解可能な多変数テスト関数）を生成し、このＣＭＴＦ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ＣＭＴＦを生成して、最も下位の非終端節点まで、同様のＣＭＴＦの生成を再帰的に行うことによって、ＣＭＤＴを構築する。
【００５２】
図３に示すように、ＣＭＤＴを構築する過程において、ＣＭＤ構築部２は、終端節点判断機能１０によりＣＭＴＦ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Ｓ１）。終端節点であると判断した場合（ステップＳ１でＹｅｓの場合）には、終端節点ラベル決定機能１１により終端節点のラベルを決定し（ステップＳ２）、処理を終了する。
【００５３】
現在の節点が終端節点でないと判断した場合（ステップＳ１でＮｏの場合）、ＣＭＤＴ構築部２は、グループラベル決定機能１２によって、訓練用データのグループラベルを決定する（ステップＳ３）。その後、ＣＭＤＴ構築部２は、ＣＭＴＦ生成機能１３によりその非終端節点におけるＣＭＴＦを生成する（ステップＳ４）。その後、ＣＭＤＴ構築部２は、生成されたＣＭＴＦの分割性能評価を早期停止判断機能１４に基づいて行い（ステップＳ５）、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Ｓ６）、分割性能が評価値T_0を満たしていない場合（ステップＳ６においてＹｅｓの場合）には、現在の節点を終端節点に変更（ステップＳ７）し、処理を終了する。分割性能が評価値満T_0を満たしている場合（ステップＳ６においてＮｏ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、ＣＭＴＦ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Ｓ８）。
【００５４】
図３に示したＣＭＤＴの構築する過程は、単一変数テスト関数（ＵＴＦ）を用いて構築される通常の決定木（ＡＰＤＴ）の構築過程にも似ている。しかしながら、ＡＰＤＴ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ＯＤＴ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

、本発明に基づいてＣＭＴＦ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、ＡＰＤＴやＯＤＴ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、ＣＭＤＴ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、ＣＭＤＴ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。

【００５７】

次に、現在節点が終端節点でないと判断した場合、ＣＭＤＴ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。ＣＭＤＴの各非終端節点にあるＣＭＴＦ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ＣＭＤＴ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、ＣＭＴＦを学習により求めるために利用されることとなる。

【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ＣＭＴＦ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。

【００５９】

まず、ＣＭＤＴ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、ＣＭＤＴ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ＣＭＤＴ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）。

【００６１】

その後、ＣＭＤＴ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ＣＭＴＦ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Ｓ１か集合Ｓ２かのどちらかに振り分ける。

【００６３】
まず、ＣＭＤＴ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、ＣＭＤＴ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$Ｓ_{1}$と集合$Ｓ_{2}$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$Ｓ_{1}$、$Ｓ_{2}$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）、ＣＭＤＴ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$Ｓ_{1}$、$Ｓ_{2}$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）、ＣＭＤＴ構築部２は、集合$Ｓ_{1}$、$Ｓ_{2}$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ＣＭＴＦを生成する処理において説明する。
【００６５】
そして、ＣＭＤＴ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$Ｓ_{1}$と$Ｓ_{2}$と割り振られた場合、集合$Ｓ_{1}$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$Ｓ_{2}$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ＣＭＴＦを生成する処理において、ＣＭＤＴ構築部２は、このグループラベルを教師信号としてＣＭＴＦの生成を行う。
【００６７】
このようにＣＭＤＴ構築部２が、非終端節点毎にＣＭＴＦ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ＣＭＴＦ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、ＣＭＤＴ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ＣＭＤＴの構成を行うことが可能となる。
【００６９】
図５は、ＣＭＤＴ構築部２がＣＭＴＦ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。ＣＭＤＴ構築部２は、ＣＭＴＦ生成機能１３に基づいて、ＣＭＴＦを生成する方法を、ＬＶＱ学習則２６、$Ｒ^{4}$－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ＣＭＴＦ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。ＣＭＤＴ構築部２は、ＣＭＴＦ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ＣＭＴＦをＬＶＱ学習則２６により生成する（ステップＳ２２）。
【００７１】
ＣＭＴＦ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、ＣＭＤＴ構築部２は、その節点におけるＣＭＴＦを$Ｒ^{4}$－Rule学習則により生成する（ステップＳ２４）。

【００７２】
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す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、サポータ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ＣＭＴＦを生成する（ステップＳ２５）。
【００７３】
上述したように、どの学習則を用いてＣＭＴＦ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ＣＭＴＦ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。
【００７４】
次に、上述したＬＶＱ学習則２６、Ｒ^4-Rule学習則２７について説明する。その他の学習則２８は上述したように、ニューラルネットワーク、サポータ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。
【００７５】
［ＬＶＱ学習則を用いたＣＭＴＦの生成］
ＬＶＱ学習則２６およびＲ^4-Rule学習則２７を用いてＣＭＴＦを生成する場合、ＣＭＤＴ構築部２は、最近傍識別器（以下、ＮＮＣという）という多変数テスト関数を生成する。このＮＮＣがＣＭＴＦに該当するものである。
【００７６】
「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ＯＤＴ（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ＮＮＣ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、ＮＮＣ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ＮＮＣ－Ｔｒｅｅという。
【００７７】
まず、ＮＮＣについて説明する。ＮＮＣ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、ＮＮＣ－Ｔｒｅｅ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１２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。
【００７８】
未知のデータＸを分類する場合、ＣＭＤＴ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Ｎｄ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。
【００７９】
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

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Ｐ１～Ｐ４の４つの既知のプロトタイプを用いる。なお、Ｐ１とＰ４とはラベル１、Ｐ２とＰ３とはラベル０を備えるものとする。

【００８０】

まず、ＣＭＤＴ構築部２は、訓練用データＸと全てのプロトタイプＰ１～Ｐ４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Ｐ１であるため、ＣＭＤＴ構築部２は、訓練用データＸをプロトタイプＰ１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、ＮＮＣ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、ＮＮＣ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、ＮＮＣは、多数の単一テスト関数（ＵＴＦ）の集まりに相当するので、非終端節点においてＮＮＣ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、ＣＭＤＴ構築部２において、ＮＮＣ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ＣＭＤＴ構築部２により作成するＮＮＣは、予め作成されるＮＮＣのサイズ（ＮＮＣに含まれるプロトタイプの数）とＮＮＣ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、ＣＭＤＴ構築部２は、この節点において生成するＮＮＣが固定型最近傍識別器の場合に、ＬＶＱ学習則を選択するため、前提としてＮＮＣのサイズとＮＮＣ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。ＮＮＣ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いものよりも速くＮＮＣ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ＮＮＣ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ＬＶＱ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ＮＮＣを修正（更新）することによってＮＮＣの精度を向上させることができる。

【００８５】

ＮＮＣを修正（更新）して精度を向上させるために、ＣＭＤＴ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。ＣＭＤＴ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、ＮＮＣの生成つまりＣＭＴＦの生成を完了する。

【００８６】

また、ＣＭＤＴ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α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αは通常、０＜α＜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ＣＭＤＴ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Ｐ０を求め、求められたプロトタイプ

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Ｐ０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Ｐ０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Ｐ０のクラスと訓練用データＸのラベルとが異なる場合、ＣＭＤＴ構築部２は、最近傍のプロトタイプＰ０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Ｐ１を求める。そして、ＣＭＤＴ構築部２は、プロトタイプＰ０とプロトタイプＰ１とを、

$$P0^{new}=P0^{old}-\alpha(X-P0^{old}) \quad \cdots\cdots(10)$$

$$P1^{new}=P1^{old}+\alpha(X-P1^{old}) \quad \cdots\cdots(11)$$

に修正（更新）する。なお、αは０＜α＜１の値を示している。

【００８８】

また、（１０）式は、プロトタイプＰ０を訓練用データＸの要素データとプロトタイプＰ０のデータ情報との差のα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Ｐ１を訓練用データＸの要素データとプロトタイプＰ１のデータ情報との差のα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Ｐ１が訓練用データＸに近づくようにプロトタイプＰ１の修正を行うと共に、ラベルの異なるプロトタイプＰ０が訓練用データＸから遠ざかるようにプロトタイプＰ０の修正を行うことによって、ＮＮＣ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。

【００９０】

またＣＭＤＴ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ＣＭＤＴ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ＮＮＣ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、

$$p(X)^{new}=\beta\cdot p(X)^{old} \quad \cdots\cdots(12)$$

となるように更新する。ただし、βは０＜β＜１の定数である。

【００９２】

プロトタイプの修正（更新）を行う場合、ＣＭＤＴ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。βは０＜β＜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ＣＭＴＦの生成においてＣＭＤＴ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、ＣＭＤＴ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。

【００９３】

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

【００９４】

まずＣＭＤＴ構築部２は、初期設定を行う（ステップＳ３１）。ＣＭＤＴ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・・・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】

続いてＣＭＤＴ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、ＣＭＤＴ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場合）、ＣＭＤＴ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）、ＣＭＤＴ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_1）とする（ステップＳ３５）。そしてＣＭＤＴ構築部２は、求められたプロトタイプＹ（j_1）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_1）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）、ＣＭＤＴ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βを掛け合わせることによって（ｐ（ｉ）＝β・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_1）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）、ＣＭＤＴ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ＣＭＤＴ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、ＣＭＤＴ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_2）とする（ステップＳ３９）。そして、ＣＭＤＴ構築部２は、プロトタイプＹ（j_1）とプロトタイプ

Ｙ（ｊ$_2$）とを、

$$Y(j_1) = Y(j_1) - \alpha(X(i) - Y(j_1)) \quad \cdots\cdots (13)$$

$$Y(j_2) = Y(j_2) + \alpha(X(i) - Y(j_2)) \quad \cdots\cdots (14)$$

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。

【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。

【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。

【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。

【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

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】

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。

【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、

$$Y(j_1) = Y(j_1) + \alpha(X(i) - Y(j_1)) \quad \cdots\cdots (15)$$

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］

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

)、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、プロトタイプの数を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(Remembrance)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。

【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）。

【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）。

【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がたくさんの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ρ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、

$$\rho(P)^{new}=\rho(P)^{old}+1 \qquad \cdots\cdots(16)$$

とし、ラベルが異なる場合には、

$$\rho(P)^{new}=\rho(P)^{old}-1 \qquad \cdots\cdots(17)$$

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

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、ＣＭＤＴ構築部２は、ステップＳ５２に示したＮＮＣの認識率Ｒと各プロトタイプの重要度を求める処理に処理を移行し、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、ＣＭＤＴ構築部２は、R^4-Rule学習則２７によるＮＮＣ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ＣＭＴＦを求める。つまり、ＣＭＤＴ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ＣＭＴＦとして生成する。

【０１２５】

以上説明したように、ＣＭＤＴ構築部２がＣＭＴＦ生成機能１３によりＣＭＴＦを生成した後（図３に示すステップＳ４の後）、ＣＭＤＴ構築部２は、早期停止判断機能１４により、ＣＭＤＴ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。

【０１２６】

具体的にＣＭＤＴ構築部２は、ＣＭＤＴ構築部２のＣＭＴＦ生成機能１３により生成されたＣＭＴＦ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、ＣＭＤＴ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、ＣＭＴＦ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が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、ＣＭＤＴの構築効率を高めることが可能となる。

【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。

【０１２８】

多変数決定木構築システム１では、ＣＭＴＦ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、ＣＭＴＦ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、ＣＭＤＴ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ＣＭＤＴ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。

【０１２９】

また、ＣＭＴＦ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、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。

【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ＣＭＤＴ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。ＡＰＤＴ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、

$$Cost(ADPT)=O(N_d \times N_t \times m) \quad \cdots\cdots(6)$$

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、

$$Cost(ODT)=O[N_d \times N_t^{\,2} \times \log_2(N_t)] \quad \cdots\cdots(8)$$

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。

【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

$$Cost(NNC-Tree)=O(N_d \times N_t \times N_l \times N_e \times N_p) \quad \cdots\cdots(18)$$

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_l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。

【０１３３】

本実施形態において使用されるデフォルト値として

$$N_l=20,\ N_e=40,\ N_p=10$$

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、

$$Cost(VariableSizeNNC)=C_1 \times O(N_d \times N_t) \quad \cdots\cdots(19)$$

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、

$$Cost(FixedSizeNNC)=C_2 \times O(N_d \times N_t) \quad \cdots\cdots(20)$$

となる。ただし、$C_2=400$である。

【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、ＡＰＤＴ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。

【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、ＡＰＤＴよりも分類精度が高い。

３）本発明で得られるＣＭＤＴは、ＡＰＤＴ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

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、データマインニング、テキストマインニング、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、ＣＭＤＴ構築部２で構成されたＣＭＤＴは、ＣＭＤＴ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ＣＭＤＴを使用（応用）するためには、構築されたＣＭＤＴの性能評価を行うことによってＣＭＤＴの有効性を判断する必要がある。ＣＭＤＴ評価部５は、このＣＭＤＴの性能評価を行う。

【０１３９】

ＣＭＤＴ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、ＣＭＤＴ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、その後に、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ＣＭＤＴを構築し、各ＣＭＤＴ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ＣＭＤＴの評価を行う。このような評価方法をn-fold cross validationと呼ぶ。

【０１４０】

ここで“n-fold cross validation ”の“ｎ”は、繰り返しＣＭＤＴを構築する回数を示しており、通常１０回程度ＣＭＤＴを構築することによってＣＭＤＴの評価を行う。１０回のＣＭＤＴ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ＣＭＤＴ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ＣＭＤＴを構築し、これらのＣＭＤＴの平均性能と信頼区間などで評価を行うことにより、ＣＭＤＴにおける信頼度の評価結果を求める。

【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ＣＭＤＴ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ＣＭＤＴを構築してＣＭＤＴの評価を行う10-fold cross validationを示している。ここで、全データをΩ_1, Ω_2・・・Ω_{10}の１０グループに分割したものとする。

【０１４３】

まず、ＣＭＤＴ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ＣＭＤＴ評価部５は、Ω_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、ＣＭＤＴ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ＣＭＤＴを構築し、構築されたＣＭＤＴをＣＭＤＴ記録部４に記録させる（ステップＳ６３）。

【０１４４】

そして、ＣＭＤＴ評価部５が、ＣＭＤＴ記録部４より構築されたＣＭＤＴを読み出すと

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ＣＭＤＴ構築部２により構築されたＣＭＤＴ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Ｓ６４）。その後、ＣＭＤＴ評価部５は、変数ｉにさらに１を加え（ｉ＝ｉ＋１）（ステップＳ６５）、ｉ＞１０の用件を満たすか否かの判断を行う（ステップＳ６６）。ｉ＞１０の用件を満たす場合、ＣＭＤＴ評価部５は、全てのグループ（Ω_1～Ω_{10}）についてＣＭＤＴを作成して評価を行ったものと判断し、ＣＭＤＴの評価処理を終了する。ｉ＞１０の用件を満たさない場合、ＣＭＤＴ評価部５は、用件を満たすまで繰り返しＳＴＥＰＳ６２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、ＣＭＤＴ構築部２により構築されるＣＭＤＴ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、ＣＭＤＴ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。

【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】

（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１３に示す決定木によって決められている。この決定木はＯＤＴ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】

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、ＮＮＣ－Ｔｒｅｅをモデルとするｃｍｄｔ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１４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）の形で表すことできる。

【０１４９】

実施例１では、10-fold cross validationを使用するため、まず図１４に示す２０００個のデータをランダムに２００個ずつ、Ω_1，Ω_2・・・Ω_{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、ＣＭＤＴ構築部２が１０個のＣＭＤＴを構築し、その後ＣＭＤＴ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ＣＭＤＴの評価結果と、従来から知られているＡＰＤＴを用いた場合の評価結果とを対比して示した表である。

【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

【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）。

【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：

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。

【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のようになる：

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

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のようになる：

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。

【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」と言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。

【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。

【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 ± 0.9	19± 0	3.53±0.33	16.292± 2.508
APDT	9.4 ± 2.1	410.0 ±29.4	---	0.964 ± 0.061
ODT	9.4 ± 2.8	112.8 ± 8.7	2 ± 0	716.866 ± 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。

【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

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ＣＭＤＴ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、ＣＭＤＴ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、データマインニング、テキストマインニング、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。

【０１７０】

さらに、上述したＣＭＤＴ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ＣＭＤＴを構築することが可能となる。

【０１７１】

また、構築されたＣＭＤＴを利用する場合には、図１９に示すように、ＣＭＤＴ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（ソフトウエア、ハードウエア、ファームウエアなど）にＣＭＤＴを変換し、このＣＭＤＴ実装変換部３９で変換されたＣＭＤＴが実装されたＣＭＤＴ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、ＣＭＤＴ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ＣＭＤＴ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、ＣＭＤＴ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ＣＭＤＴ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、ＣＭＤＴ記録部４とＣＭＤＴ実装変換部３９とを加えることにより、ＣＭＤＴ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。

【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。

【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、ＣＭＤＴ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】

【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】ＣＭＤＴ構築部の機能を示したブロック図である。
【図３】ＣＭＤＴ構築部がＣＭＤＴを構築する過程を示したフローチャートである。
【図４】ＣＭＤＴ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】ＣＭＤＴ構築部がＣＭＴＦ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】ＣＭＤＴ構築部がＣＭＴＦ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】ＣＭＤＴ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】ＣＭＤＴ構築部がＮＮＣを生成する過程を示したフローチャートである。
【図９】ＣＭＤＴ構築部がＮＮＣ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とを示したブロック図である。
【図１１】ＣＭＤＴ構築部がR^4-Rule学習則に基づいてＣＭＴＦ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ＣＭＤＴ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２０００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ＣＭＤＴ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ＣＭＤＴ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ＣＭＤＴ記録部とＣＭＤＴ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ｉｆ－ｔｈｅｎ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２３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。

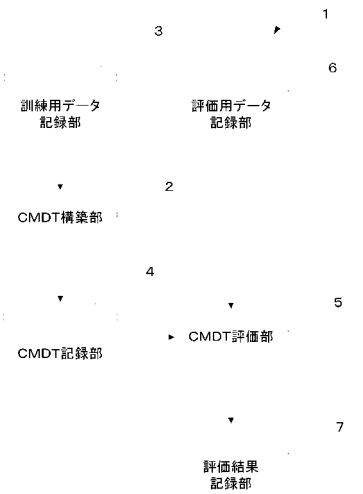
【符号の説明】

【０１７５】

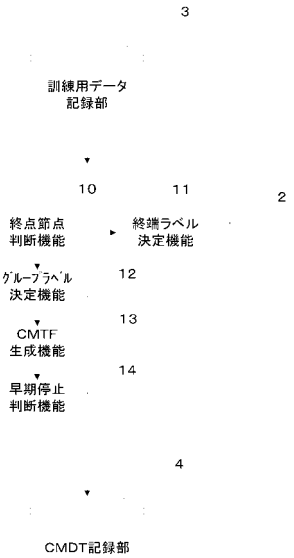
１　…多変数決定木構築システム
２　…ＣＭＤＴ構築部（グループラベル付与手段、多変数テスト関数生成手段、早期停止判断手段、終端節点判別手段、コンピュータ）
３　…訓練用データ記録部
４　…ＣＭＤＴ記録部
５　…ＣＭＤＴ評価部
６　…評価用データ記録部
７　…評価結果記録部
１０　…終端節点判断機能（終端節点判別手段）
１１　…終端節点ラベル決定機能（終端節点判別手段）

１２　…グループラベル決定機能（グループラベル付与手段）
１３　…ＣＭＴＦ生成機能（多変数テスト関数生成手段）
１４　…早期停止判断機能（早期停止判断手段）
２１　…認識機能
２２　…記憶機能
２３　…忘却機能
２４　…復習機能
２６　…ＬＶＱ学習則
２７　…Ｒ４－Rule学習則
２８　…その他の学習則
３０　…データ獲得部
３１　…原始データ記録部
３２　…データ変換部
３３　…ラベル付与部
３４　…データ記録部
３９　…ＣＭＤＴ実装変換部
４０　…ＣＭＤＴ実装部
４１　…処理用データ記録部
４２　…処理結果記録部
４３　…データ処理装置

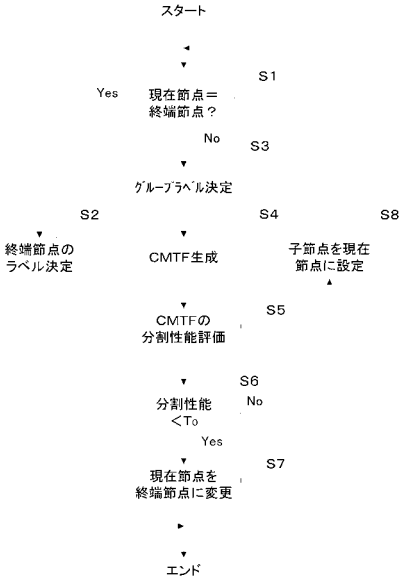
【図１】



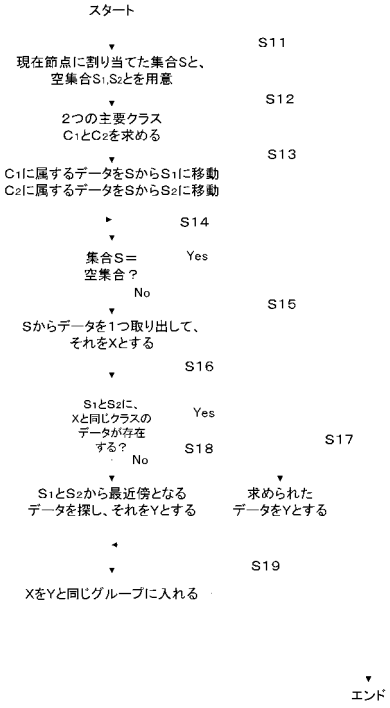
【図２】



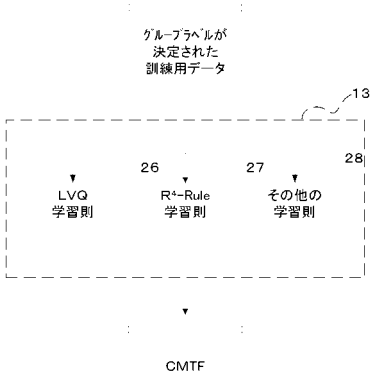
【図３】


スタート
S1
現在節点＝
終端節点？
Yes
No
S3
ｸﾞﾙｰﾌﾟﾗﾍﾞﾙ決定
S2
S4
S8
終端節点の
ラベル決定
CMTF生成
子節点を現在
節点に設定
S5
CMTFの
分割性能評価
S6
分割性能
＜T0
No
Yes
S7
現在節点を
終端節点に変更
エンド

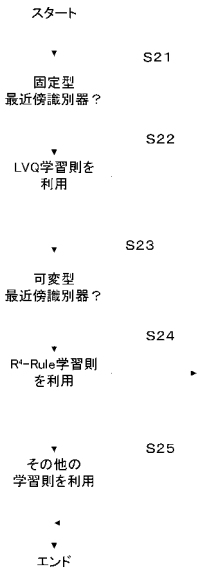
【図４】


スタート
S11
現在節点に割り当てた集合Sと、
空集合S1,S2とを用意
S12
2つの主要クラス
C1とC2を求める
S13
C1に属するデータをSからS1に移動
C2に属するデータをSからS2に移動
S14
集合S＝
空集合？
Yes
No
S15
Sからデータを1つ取り出して、
それをXとする
S16
S1とS2に、
Xと同じクラスの
データが存在
する？
Yes
No
S17
S18
S1とS2から最近傍となる
データを探し、それをYとする
求められた
データをYとする
S19
XをYと同じグループに入れる
エンド

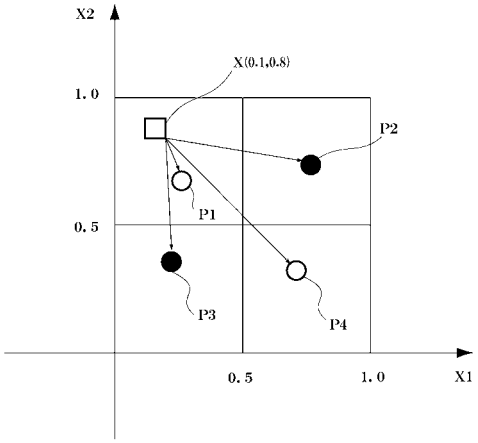
【図５】


ｸﾞﾙｰﾌﾟﾗﾍﾞﾙが
決定された
訓練用データ
13
26
27
28
LVQ
学習則
R4-Rule
学習則
その他の
学習則
CMTF

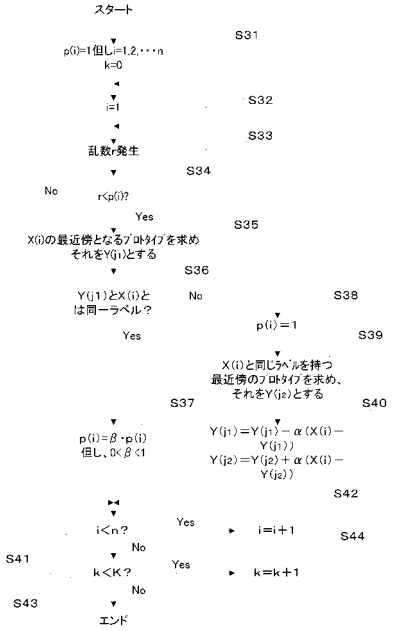
【図６】


スタート
S21
固定型
最近傍識別器？
S22
LVQ学習則を
利用
S23
可変型
最近傍識別器？
S24
R4-Rule学習則
を利用
S25
その他の
学習則を利用
エンド

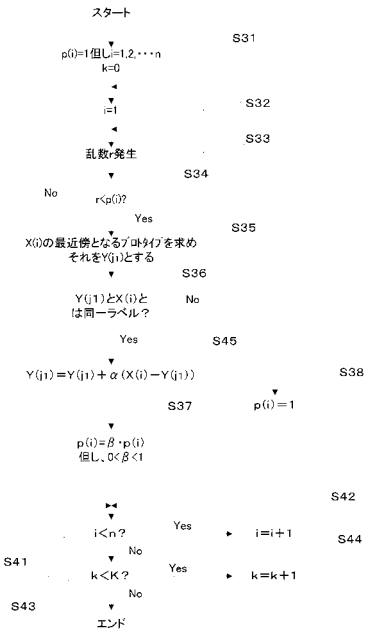
【図７】


X2
1.0
0.5
X(0.1,0.8)
P1
P2
P3
P4
0.5
1.0
X1

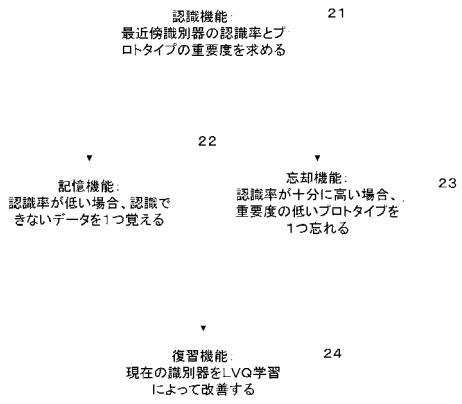
【図８】


スタート
S31
p(i)=1但し,i=1,2,・・・n
k=0
S32
i=1
S33
乱数r発生
S34
No
r<p(i)?
Yes
S35
X(i)の最近傍となるプロトタイプを求め
それをY(j1)とする
S36
Y(j1)とX(i)と
は同一ラベル？
No
S38
p(i)=1
S39
X(i)と同じラベルを持つ
最近傍のプロトタイプを求め、
それをY(j2)とする
S37
S40
p(i)=β・p(i)
但し、0<β<1
Y(j1)=Y(j1)－α(X(i)－
Y(j1))
Y(j2)=Y(j2)＋α(X(i)－
Y(j2))
S42
i<n？
Yes
i=i+1
S44
No
S41
k<K？
Yes
k=k+1
No
S43
エンド

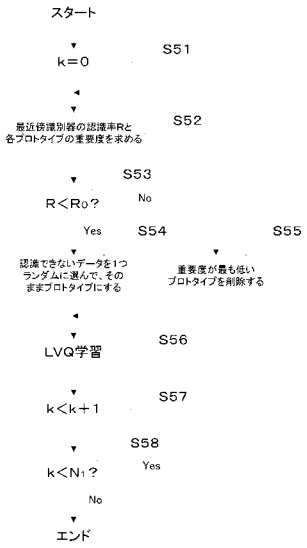
【図９】


スタート
S31
p(i)=1但し,i=1,2,・・・n
k=0
S32
i=1
S33
乱数r発生
S34
No
r<p(i)?
Yes
S35
X(i)の最近傍となるプロトタイプを求め
それをY(j1)とする
S36
Y(j1)とX(i)と
は同一ラベル？
No
Yes
S45
Y(j1)=Y(j1)＋α(X(i)－Y(j1))
S38
S37
p(i)=1
p(i)=β・p(i)
但し、0<β<1
S42
i<n？
Yes
i=i+1
S44
No
S41
k<K？
Yes
k=k+1
No
S43
エンド

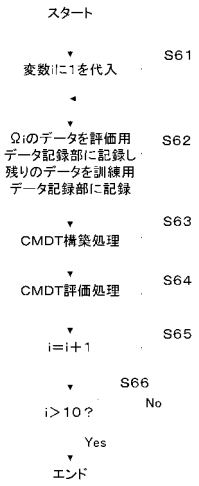
【図１０】


認識機能:
最近傍識別器の認識率とプ
ロトタイプの重要度を求める
21
22
記憶機能:
認識率が低い場合、認識で
きないデータを1つ覚える
忘却機能:
認識率が十分に高い場合、
重要度の低いプロトタイプを
1つ忘れる
23
復習機能:
現在の識別器をLVQ学習
によって改善する
24

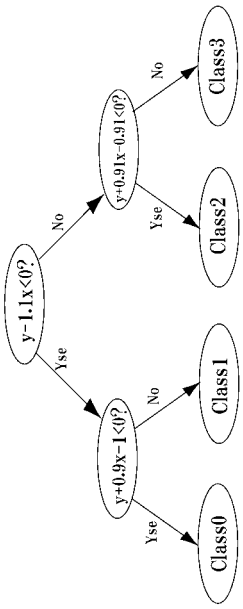
【図１１】



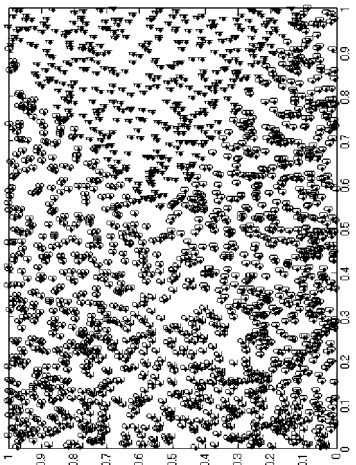
【図１２】



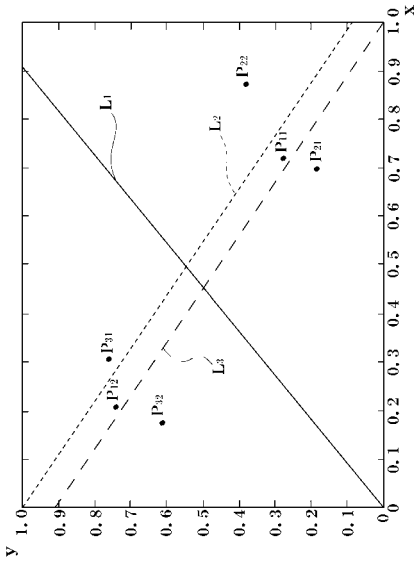
【図１３】



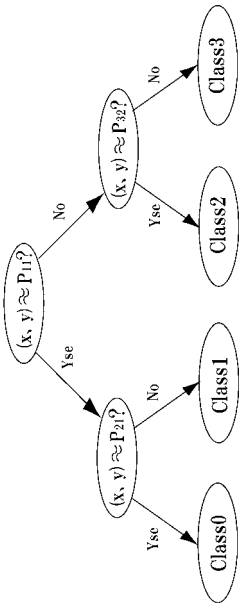
【図１４】



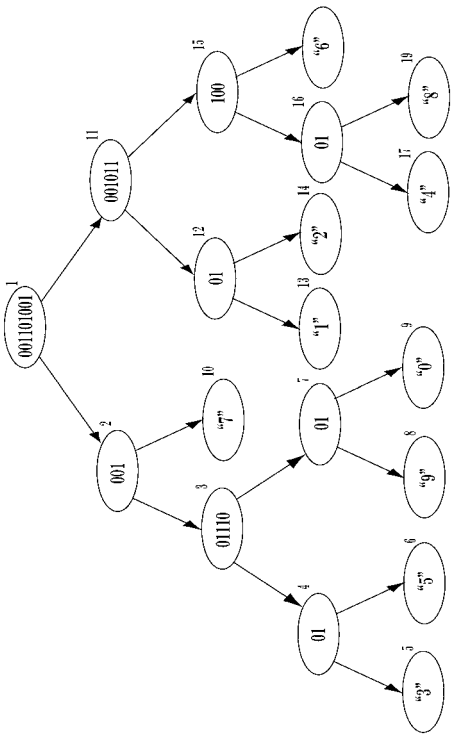
【図１５】



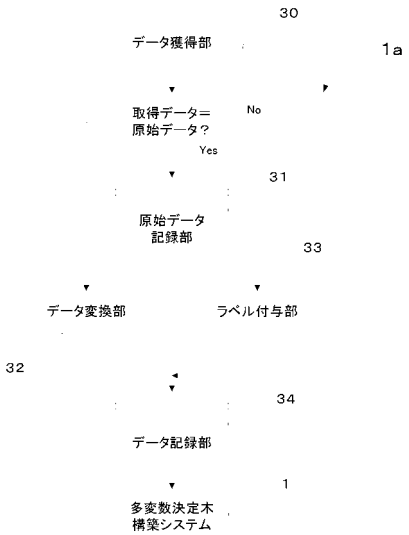
【図１６】



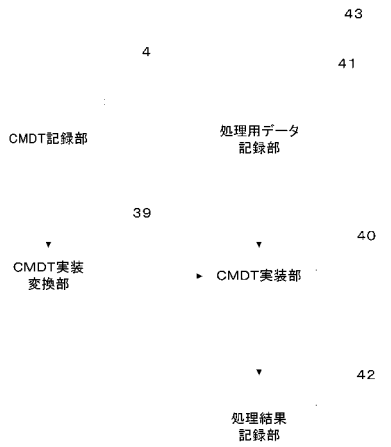
【図１７】



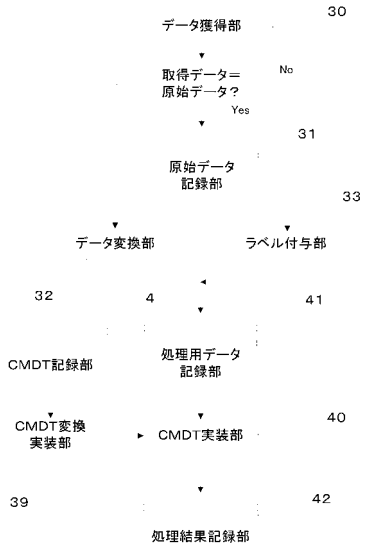
【図１８】



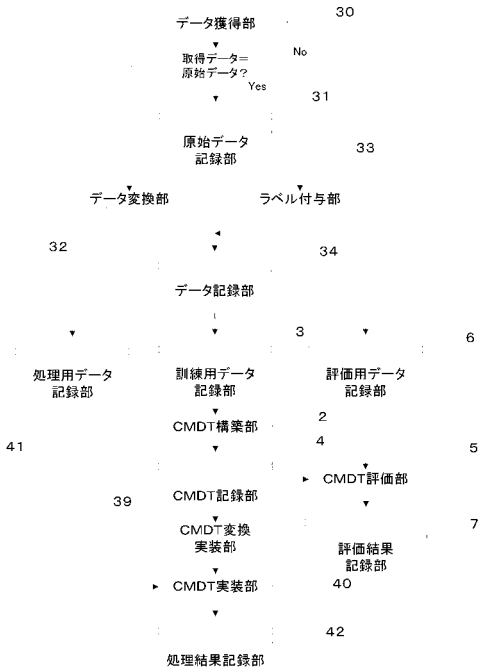
【図１９】



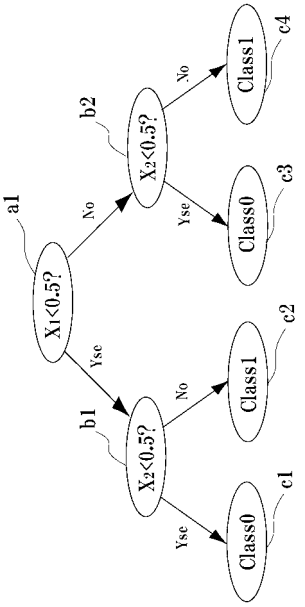
【図２０】



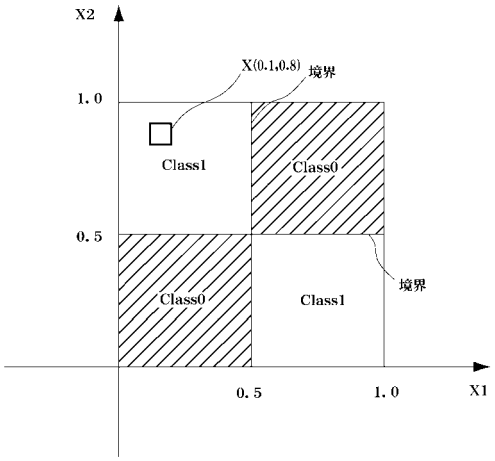
【図２１】



【図２２】



【図２３】



フロントページの続き

(56)参考文献 Qiangfu Zhao et al., Inducing Multivariate Decision Trees Quickly and Effectively, マルチメディア通信と分散処理ワークショップ論文集, 社団法人情報処理学会, ２００５年１１月３０日, Vol.2005, No.19, pp.230-234
川連 太陽 他, ＮＮＣに基づく距離空間での決定木の構築, 電子情報通信学会技術研究報告, 社団法人電子情報通信学会, ２００４年１１月１２日, Vol.104, No.448, pp.53-58
浜本 義彦 他, 遺伝的アルゴリズムを用いた最近傍識別器のための代表サンプル選択, 電子情報通信学会論文誌A, 社団法人電子情報通信学会, １９９７年 2月２５日, Vol.J80-A, No.2, pp.371-378
Sreerama K. Murthy et al., A System for Induction of Oblique Decision Trees, Journal of Artificial Intelligence Research, AI Access Foundation, １９９４年, Vol.2, pp.1-32

(58)調査した分野(Int.Cl., ＤＢ名)
Ｇ０６Ｎ 3／００
Ｇ０６Ｎ 5／０４
Ｇ０６Ｆ １７／３０
ＪＳＴＰｌｕｓ（ＪＤｒｅａｍＩＩ）
ＩＥＥＥ Ｘｐｌｏｒｅ