

(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2007-213442

(P2007-213442A)

(43) 公開日 平成19年8月23日(2007.8.23)

(51) Int. Cl.

G06N 5/04 (2006.01)

F I

G06N 5/04 550C

テーマコード(参考)

審査請求 未請求 請求項の数 9 O L (全 21 頁)

(21) 出願番号 特願2006-34344 (P2006-34344)  
 (22) 出願日 平成18年2月10日(2006.2.10)

(71) 出願人 391041062  
 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

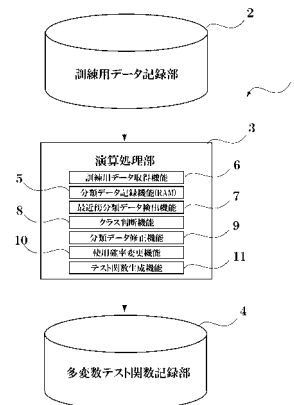
(54) 【発明の名称】 多変数テスト関数生成装置、多変数テスト関数生成システム、多変数テスト関数生成方法および多変数テスト関数を生成するためのプログラム

(57) 【要約】

【課題】 理解可能であって更に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可能な多変数テスト関数を生成すること。

【解決手段】 本発明に係る多変数テスト関数生成システム1は、複数要素の要素データを有する入力データを複数のクラスに分類するための多変数テスト関数を生成するシステムである。多変数テスト関数生成システム1は、訓練用データを記録する訓練用データ記録手段2と、訓練用データに基づいて分類データの修正を行い、修正された分類データに基づいて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3と、生成された多変数テスト関数を記録する多変数テスト関数記録手段4とを備えている。

【選択図】 図1



## 【特許請求の範囲】

## 【請求項1】

複数要素の要素データを有する複数の入力データを複数のクラスに分類するための多変数テスト関数を生成する多変数テスト関数生成装置であって、

予め要素データと分類されるべきクラスとが既知である入力データからなる訓練用データを取得する訓練用データ取得手段と、

前記要素データに対応するデータ情報と前記クラスを示すクラス情報とを有する複数の分類データを記録する分類データ記録手段と、

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、前記分類データ記録手段に記録される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分類データを求める最近傍分類データ検出手段と、

該最近傍分類データ検出手段により求められた分類データのクラス情報と前記訓練用データのクラスとが同一クラスであるか否かを判断するクラス判断手段と、

該クラス判断手段により前記分類データのクラス情報と前記訓練用データのクラスとが異なるクラスであると判断された場合に、前記分類データ記録手段に記録される当該分類データのデータ情報を前記要素データの空間座標から遠ざかるように修正するとともに、前記クラス判断手段により前記訓練用データと同一クラスであると判断される他の分類データを前記最近傍分類データ検出手段により検出させ、前記分類データ記録手段に記録される当該他の分類データのデータ情報を前記要素データの空間座標に近づくように修正する分類データ修正手段と、

前記最近傍分類データ検出手段により検出された最近傍となる前記分類データのクラス情報が前記クラス判断手段により前記訓練用データのクラスと同一であると判断された場合に値が減少する使用確率変数を、前記訓練用データ毎に付与する使用確率変更手段と、

前記分類データ修正手段により修正がなされた分類データのデータ情報とクラス情報とに基づいて前記多変数テスト関数を生成するテスト関数生成手段とを有し、

前記最近傍分類データ検出手段は、前記使用確率変数の値が所定値以上を示す前記訓練用データのみを用いて前記分類データの検出を行うこと

を特徴とする多変数テスト関数生成装置。

## 【請求項2】

複数要素の要素データを有する複数の入力データを複数のクラスに分類するための多変数テスト関数を生成する多変数テスト関数生成装置であって、

予め要素データと分類されるべきクラスとが既知である入力データからなる訓練用データを取得する訓練用データ取得手段と、

前記要素データに対応するデータ情報と前記クラスを示すクラス情報とを有する複数の分類データを記録する分類データ記録手段と、

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、前記分類データ記録手段に記録される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分類データを求める最近傍分類データ検出手段と、

該最近傍分類データ検出手段により求められた分類データのクラス情報と前記訓練用データのクラスとが同一クラスであるか否かを判断するクラス判断手段と、

該クラス判断手段により前記分類データのクラス情報と前記訓練用データのクラスとが同一クラスであると判断された場合に、前記分類データ記録手段に記録される当該分類データのデータ情報を前記要素データの空間座標に近づくように修正する分類データ修正手段と、

前記最近傍分類データ検出手段により検出された最近傍となる前記分類データのクラス

10

20

30

40

50

情報が前記クラス判断手段により前記訓練用データのクラスと同一であると判断された場合に値が減少する使用確率変数を、前記訓練用データ毎に付与する使用確率変更手段と、前記分類データ修正手段により修正がなされた分類データのデータ情報とクラス情報とに基づいて前記多変数テスト関数を生成するテスト関数生成手段と

を有し、

前記最近傍分類データ検出手段は、前記使用確率変数の値が所定値以上を示す前記訓練用データのみを用いて前記分類データの検出を行うこと

を特徴とする多変数テスト関数生成装置。

#### 【請求項 3】

複数要素の要素データを有する入力データを複数のクラスに分類するための多変数テスト関数を生成する多変数テスト関数生成システムであって、

予め要素データと分類されるべきクラスとが既知である入力データを訓練用データとして記録する訓練用データ記録手段と、

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クラスを示すクラス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クラス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クラス情報とに基づいて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、

生成された多変数テスト関数を記録する多変数テスト関数記録手段と

を備え、

前記多変数テスト関数生成手段は、最近傍となる前記分類データのクラス情報が前記訓練用データのクラスと同一であった場合に値が減少する使用確率変数を前記訓練用データ毎に付与し、前記使用確率変数が所定値以上を示す前記訓練用データを用いて前記分類データのデータ情報の修正処理を行う

ことを特徴とする多変数テスト関数生成システム

#### 【請求項 4】

前記多変数テスト関数生成手段は、

前記訓練用データの要素データにおける空間座標から最近傍となる分類データのクラス情報が前記訓練用データのクラスと異なる場合に、当該分類データのデータ情報を前記要素データの空間座標から遠ざかるように修正し、

さらに前記訓練用データと同一クラスのクラス情報を有する分類データのうち、前記要素データの空間座標に最近傍となる他の分類データを求めて、当該他の分類データのデータ情報を前記要素データの空間座標に近づくように修正する

ことを特徴とする請求項 3 に記載の多変数テスト関数生成システム。

#### 【請求項 5】

前記多変数テスト関数生成手段は、

前記訓練用データの要素データにおける空間座標から最近傍となる分類データのクラス情報が前記訓練用データのクラスと同一の場合に、当該分類データのデータ情報を前記要素データの空間座標に近づくように修正する

ことを特徴とする請求項 3 に記載の多変数テスト関数生成システム。

#### 【請求項 6】

複数要素の要素データを有する複数の入力データを複数のクラスに分類するための多変数テスト関数を生成する多変数テスト関数生成方法であって、

訓練用データ取得手段が、予め要素データと分類されるべきクラスとが既知である入力データからなる訓練用データを取得する訓練用データ取得ステップと、

最近傍分類データ検出手段が、前記訓練用データ取得手段により取得された前記要素デ

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クラスを示すクラス情報とを有する複数の分類データを、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分類データを求める最近傍分類データ検出ステップと、

クラス判断手段が、最近傍分類データ検出手段により求められた分類データのクラス情報と前記訓練用データのクラスとが同一クラスであるか否かを判断するクラス判断ステップと、

前記クラス判断手段により前記分類データのクラス情報と前記訓練用データのクラスとが異なるクラスであると判断された場合に、分類データ修正手段が当該分類データのデータ情報を前記要素データの空間座標から遠ざかるように修正するとともに、前記クラス判断手段により前記訓練用データと同一クラスであると判断される他の分類データを前記最近傍分類データ検出手段により検出させ、当該他の分類データのデータ情報を前記要素データの空間座標に近づくように修正する分類データ修正ステップと、

前記最近傍分類データ検出手段により検出された最近傍となる前記分類データのクラス情報が前記クラス判断手段により前記訓練用データのクラスと同一であると判断された場合に値を減少させる使用確率変数を、使用確率変更手段が前記訓練用データ毎に付与する使用確率変更ステップと、

テスト関数生成手段が、前記分類データ修正手段により修正がなされた分類データのデータ情報とクラス情報とに基づいて入力データの分類を行う多変数テスト関数を生成するテスト関数生成ステップと

を有し、

最近傍分類データ検出ステップにおいて、前記使用確率変数の値が所定値以上を示す前記訓練用データのみを用いて前記最近傍分類データ検出手段が前記分類データの検出を行うこと

を特徴とする多変数テスト関数生成方法。

#### 【請求項7】

複数要素の要素データを有する複数の入力データを複数のクラスに分類するための多変数テスト関数を生成する多変数テスト関数生成方法であって、

訓練用データ取得手段が、予め要素データと分類されるべきクラスとが既知である入力データからなる訓練用データを取得する訓練用データ取得ステップと、

最近傍分類データ検出手段が、前記訓練用データ取得手段により取得された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クラスを示すクラス情報とを有する複数の分類データを、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分類データを求める最近傍分類データ検出ステップと、

クラス判断手段が、最近傍分類データ検出手段により求められた分類データのクラス情報と前記訓練用データのクラスとが同一クラスであるか否かを判断するクラス判断ステップと、

該クラス判断手段により前記分類データのクラス情報と前記訓練用データのクラスとが同一クラスであると判断された場合に、分類データ修正手段が当該分類データのデータ情報を前記要素データの空間座標に近づくように修正する分類データ修正ステップと、

前記最近傍分類データ検出手段により検出された最近傍となる前記分類データのクラス情報が前記クラス判断手段により前記訓練用データのクラスと同一であると判断された場合に値を減少させる使用確率変数を、使用確率変更手段が前記訓練用データ毎に付与する使用確率変更ステップと、

テスト関数生成手段が、前記分類データ修正手段により修正がなされた分類データのデ

ータ情報とクラス情報とに基づいて入力データの分類を行う多変数テスト関数を生成する  
テスト関数生成ステップと

を有し、

前記最近傍分類データ検出ステップにおいて、前記使用確率変数の値が所定値以上を示す前記訓練用データのみを用いて前記最近傍分類データ検出手段が前記分類データの検出を行うこと

を特徴とする多変数テスト関数生成方法。

【請求項 8】

複数要素の要素データを有する複数の入力データを複数のクラスに分類するための多変数テスト関数を生成するために、コンピュータに、

10

予め要素データと分類されるべきクラスとが既知である入力データからなる訓練用データを、訓練用データ取得手段により取得させる訓練用データ取得ステップと、

該訓練用データ取得ステップにおいて取得された前記要素データの要素数に基づいて、最近傍分類データ検出手段により当該要素数に対応する複数次元の特徴空間を構成させ、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させるとともに、前記要素データに対応するデータ情報と前記クラスを示すクラス情報とを有する複数の分類データを、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させ、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分類データを求めさせる最近傍分類データ検出ステップと、

該最近傍分類データ検出ステップにおいて求められた分類データのクラス情報と前記訓練用データのクラスとが同一クラスであるか否かを、クラス判断手段により判断させるクラス判断ステップと、

20

該クラス判断ステップにおいて前記分類データのクラス情報と前記訓練用データのクラスとが異なるクラスであると判断された場合に、分類データ修正手段により当該分類データのデータ情報を前記要素データの空間座標から遠ざかるように修正させるとともに、前記クラス判断ステップにおいて前記訓練用データと同一クラスであると判断される他の分類データを前記最近傍分類データ検出ステップにより検出させ、当該他の分類データのデータ情報を前記要素データの空間座標に近づくように修正させる分類データ修正ステップと、

前記最近傍分類データ検出ステップにおいて検出された最近傍となる前記分類データのクラス情報が前記クラス判断ステップにおいて前記訓練用データのクラスと同一であると判断された場合に値を減少させる使用確率変数を、使用確率変更手段により前記訓練用データ毎に付与させる使用確率変更ステップと、

30

前記分類データ修正ステップにおいて修正がなされた分類データのデータ情報とクラス情報とに基づいて入力データの分類を行う多変数テスト関数を、テスト関数生成手段により生成させるテスト関数生成ステップと

を実行させ、

さらに、前記最近傍分類データ検出ステップにおいて、前記使用確率変数の値が所定値以上を示す前記訓練用データのみを用いて前記最近傍分類データ検出手段に前記分類データの検出を行わせる

40

ことを特徴とする多変数テスト関数を生成するためのプログラム。

【請求項 9】

複数要素の要素データを有する複数の入力データを複数のクラスに分類するための多変数テスト関数を生成するために、コンピュータに、

予め要素データと分類されるべきクラスとが既知である入力データからなる訓練用データを、訓練用データ取得手段により取得させる訓練用データ取得ステップと、

該訓練用データ取得ステップにおいて取得された前記要素データの要素数に基づいて、最近傍分類データ検出手段により当該要素数に対応する複数次元の特徴空間を構成させ、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させるとともに、前記要素データに対応するデータ情報と前記クラスを示すクラス情報とを有する複数の

50

分類データを、前記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させ、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分類データを求める最近傍分類データ検出ステップと、

該最近傍分類データ検出ステップにおいて求められた分類データのクラス情報と前記訓練用データのクラスとが同一クラスであるか否かを、クラス判断手段により判断させるクラス判断ステップと、

該クラス判断ステップにおいて前記分類データのクラス情報と前記訓練用データのクラスとが同一クラスであると判断された場合に、分類データ修正手段により当該分類データのデータ情報を前記要素データの空間座標に近づくように修正させる分類データ修正ステップと、

10

前記最近傍分類データ検出ステップにおいて検出された最近傍となる前記分類データのクラス情報が前記クラス判断ステップにおいて前記訓練用データのクラスと同一であると判断された場合に値を減少させる使用確率変数を、使用確率変更手段により前記訓練用データ毎に付与させる使用確率変更ステップと、

前記分類データ修正ステップにおいて修正がなされた分類データのデータ情報とクラス情報とに基づいて入力データの分類を行う多変数テスト関数を、テスト関数生成手段により生成させるテスト関数生成ステップと

を実行させ、

さらに、前記最近傍分類データ検出ステップにおいて、前記使用確率変数の値が所定値以上を示す前記訓練用データのみを用いて前記最近傍分類データ検出手段に前記分類データの検出を行わせる

20

ことを特徴とする多変数テスト関数を生成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、複数要素の要素データを有する複数の入力データを複数のクラスに分類する多変数テスト関数を生成するための多変数テスト関数生成装置、多変数テスト関数生成システム、多変数テスト関数生成方法および多変数テスト関数を生成するためのプログラムに関する。

【背景技術】

30

【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。

【0003】

複数の *if - then* ルールを効率よく、わかりやすくまとめる方法としては、決定木（ツリー構造）がある。図5は、決定木の一例を示している。図5に示す決定木は決定結果（クラス）*Class 0*、*Class 1*を持つ終端節点（*c 1 ~ c 4*）と単一変数テスト関数（*UTF*：Univariate Test Function）を使って局所的な判断を行う非終端節点（*a 1*、*b 1*、*b 2*）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 *a 1*（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）へと判断分類を行い、最終的に終端節点における決定結果に基づいて判断を行う。

40

【0004】

例えば、入力データ： $X = (0.1, 0.8)$ として、図5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 *a 1*（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であってルート内のテスト関数を満たす場合に次の判断が求められる非終端節点 *b 1* へと処理が移行する。

50

## 【0005】

そしてコンピュータは、非終端節点 b 1 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 b 1 の下位の終端節点であって決定結果として *Class 1* を備える終端節点 c 2 へと処理が移行する。コンピュータは終端節点 c 2 において決定結果として *Class 1* の決定結果を取得して、入力データ：  $X$  が *Class 1* であると判断する。

## 【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判断を行わせることによって、判断内容を *if-then* ルールで示すことができるので、理解しやすいという特徴がある。

## 【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 6 参照）ので、通常の決定木は *APDT* (*Axis-Parallel Decision Tree*) と呼ばれる。*ADTP* 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】

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性能（認識率など）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（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、対応する *if-then* ルールが非常に長くなり、理解が困難なものになってしまうという問題があった。

## 【0009】

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*）を用いることにより節点数を減らす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* では次式に示す式 1 のようなテスト関数が用いられている。

## 【数 1】

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

・・・・・・・・ (1)

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、*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。

【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

20

30

40

50

## 【発明の開示】

## 【発明が解決しようとする課題】

## 【0010】

しかしながら、ODTのような多変数テスト関数を利用する決定木（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、式1に示す多変数テスト関数では、データXが超平面の下側（ $F(X) < 0$ ）ならばクラス0と判断され、超平面の上（ $F(X) \geq 0$ ）ならばクラス1と分類される場合、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判断することは容易ではない。

## 【0011】

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されるという問題があった。例えば、最も簡単多変数決定木であるODTを構築する場合でさえ、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。

## 【0012】

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「S. K. Murthy, S. Kasif and S. Salzberg, "A system for induction of oblique decision trees," Journal of Artificial Intelligence Research, No. 2, pp. 112, 1994」）。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。しかしながら、OC1の中に確率的方法が含まれるので、実際の計算量が非常に大きくなる場合がある。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。

## 【0013】

本発明は、上記問題に鑑みてなされたものであり、理解可能であって更に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可能な多変数テスト関数を生成するための多変数テスト関数生成装置、多変数テスト関数生成システム、多変数テスト関数生成方法および多変数テスト関数を生成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】

## 【0014】

上記課題を解決するために、本発明に係る多変数テスト関数生成装置は、複数要素の要素データを有する複数の入力データを複数のクラスに分類するための多変数テスト関数を生成する多変数テスト関数生成装置であって、予め要素データと分類されるべきクラスとが既知である入力データからなる訓練用データを取得する訓練用データ取得手段と、前記要素データに対応するデータ情報と前記クラスを示すクラス情報とを有する複数の分類データを記録する分類データ記録手段と、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、前記分類データ記録手段に記録される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分類データを求める最近傍分類データ検出手段と、該最近傍分類データ検出手段により求められた分類データのクラス情報と前記訓練用データのクラスとが同一クラスであるか否かを判断するクラス判断手段と、該クラス判断手段により前記分類データのクラス情報と前記訓練用データのクラスとが異なるクラスであると判断された場合に、前記分類データ記録手段に記録される当該分類データのデータ情報を前記要素データの空間座標から遠ざかるように修正するとともに、前記クラス判断手段により前記訓練用データと同一クラスであると判断される他の分類データを前記最近傍分類データ検出手段により検出させ、前記分類データ記録手段に記録される当該他の分類データのデータ情報を前記要素データの空間座標に近づくように修正する分類データ修正手段と、前記最近傍分類データ検出手段により検出された最近傍となる前

10

20

30

40

50



記分類データのクラス情報が前記クラス判断手段により前記訓練用データのクラスと同一であると判断された場合に値が減少する使用確率変数を、前記訓練用データ毎に付与する使用確率変更手段と、前記分類データ修正手段により修正がなされた分類データのデータ情報とクラス情報とに基づいて前記多変数テスト関数を生成するテスト関数生成手段とを有し、前記最近傍分類データ検出手段は、前記使用確率変数の値が所定値以上を示す前記訓練用データのみを用いて前記分類データの検出を行うことを特徴とする。

**【0015】**

一方で、本発明に係る多変数テスト関数生成装置は、複数要素の要素データを有する複数の入力データを複数のクラスに分類するための多変数テスト関数を生成する多変数テスト関数生成装置であって、予め要素データと分類されるべきクラスとが既知である入力データからなる訓練用データを取得する訓練用データ取得手段と、前記要素データに対応するデータ情報と前記クラスを示すクラス情報とを有する複数の分類データを記録する分類データ記録手段と、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、前記分類データ記録手段に記録される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分類データを求める最近傍分類データ検出手段と、該最近傍分類データ検出手段により求められた分類データのクラス情報と前記訓練用データのクラスとが同一クラスであるか否かを判断するクラス判断手段と、該クラス判断手段により前記分類データのクラス情報と前記訓練用データのクラスとが同一クラスであると判断された場合に、前記分類データ記録手段に記録される当該分類データのデータ情報を前記要素データの空間座標に近づくように修正する分類データ修正手段と、前記最近傍分類データ検出手段により検出された最近傍となる前記分類データのクラス情報が前記クラス判断手段により前記訓練用データのクラスと同一であると判断された場合に値が減少する使用確率変数を、前記訓練用データ毎に付与する使用確率変更手段と、前記分類データ修正手段により修正がなされた分類データのデータ情報とクラス情報とに基づいて前記多変数テスト関数を生成するテスト関数生成手段とを有し、前記最近傍分類データ検出手段は、前記使用確率変数の値が所定値以上を示す前記訓練用データのみを用いて前記分類データの検出を行うことを特徴とする。

**【0016】**

本発明に係る多変数テスト関数生成システムは、複数要素の要素データを有する入力データを複数のクラスに分類するための多変数テスト関数を生成する多変数テスト関数生成システムであって、予め要素データと分類されるべきクラスとが既知である入力データを訓練用データとして記録する訓練用データ記録手段と、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クラスを示すクラス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クラス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クラス情報とに基づいて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、生成された多変数テスト関数を記録する多変数テスト関数記録手段とを備え、前記多変数テスト関数生成手段は、最近傍となる前記分類データのクラス情報が前記訓練用データのクラスと同一であった場合に値が減少する使用確率変数を前記訓練用データ毎に付与し、前記使用確率変数が所定値以上を示す前記訓練用データを用いて前記分類データのデータ情報の修正処理を行うことを特徴とする。

**【0017】**

また、多変数テスト関数生成システムは、前記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における空間座標から最近傍となる分類データのクラス情報が前記訓練用データのクラスと異なる場合に、当該分類データのデータ情報を前記要素データの空間座標から遠ざかるように修正し、さらに前記訓練用データと同一クラスのクラス情報を有する分類データのうち、前記要素データの空間座標に最近傍となる他の分類データを求めて、当該他の分類データのデータ情報を前記要素データの空間座標に近づくように修正することを特徴とするものであってもよい。

【0018】

さらに、多変数テスト関数生成システムは、前記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における空間座標から最近傍となる分類データのクラス情報が前記訓練用データのクラスと同一の場合に、当該分類データのデータ情報を前記要素データの空間座標に近づくように修正することを特徴とするものであってもよい。

10

【0019】

本発明に係る多変数テスト関数生成方法は、複数要素の要素データを有する複数の入力データを複数のクラスに分類するための多変数テスト関数を生成する多変数テスト関数生成方法であって、訓練用データ取得手段が、予め要素データと分類されるべきクラスとが既知である入力データからなる訓練用データを取得する訓練用データ取得ステップと、最近傍分類データ検出手段が、前記訓練用データ取得手段により取得された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クラスを示すクラス情報とを有する複数の分類データを、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分類データを求める最近傍分類データ検出ステップと、クラス判断手段が、最近傍分類データ検出手段により求められた分類データのクラス情報と前記訓練用データのクラスとが同一クラスであるか否かを判断するクラス判断ステップと、前記クラス判断手段により前記分類データのクラス情報と前記訓練用データのクラスとが異なるクラスであると判断された場合に、分類データ修正手段が当該分類データのデータ情報を前記要素データの空間座標から遠ざかるように修正するとともに、前記クラス判断手段により前記訓練用データと同一クラスであると判断される他の分類データを前記最近傍分類データ検出手段により検出させ、当該他の分類データのデータ情報を前記要素データの空間座標に近づくように修正する分類データ修正ステップと、前記最近傍分類データ検出手段により検出された最近傍となる前記分類データのクラス情報が前記クラス判断手段により前記訓練用データのクラスと同一であると判断された場合に値を減少させる使用確率変数を、使用確率変更手段が前記訓練用データ毎に付与する使用確率変更ステップと、テスト関数生成手段が、前記分類データ修正手段により修正がなされた分類データのデータ情報とクラス情報とに基づいて入力データの分類を行う多変数テスト関数を生成するテスト関数生成ステップとを有し、最近傍分類データ検出ステップにおいて、前記使用確率変数の値が所定値以上を示す前記訓練用データのみを用いて前記最近傍分類データ検出手段が前記分類データの検出を行うことを特徴とする。

20

30

【0020】

また、本発明に係る多変数テスト関数生成方法は、複数要素の要素データを有する複数の入力データを複数のクラスに分類するための多変数テスト関数を生成する多変数テスト関数生成方法であって、訓練用データ取得手段が、予め要素データと分類されるべきクラスとが既知である入力データからなる訓練用データを取得する訓練用データ取得ステップと、最近傍分類データ検出手段が、前記訓練用データ取得手段により取得された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クラスを示すクラス情報とを有する複数の分類データを、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分類デ

40

50

ータを求める最近傍分類データ検出ステップと、クラス判断手段が、最近傍分類データ検出手段により求められた分類データのクラス情報と前記訓練用データのクラスとが同一クラスであるか否かを判断するクラス判断ステップと、該クラス判断手段により前記分類データのクラス情報と前記訓練用データのクラスとが同一クラスであると判断された場合に、分類データ修正手段が当該分類データのデータ情報を前記要素データの空間座標に近づくように修正する分類データ修正ステップと、前記最近傍分類データ検出手段により検出された最近傍となる前記分類データのクラス情報が前記クラス判断手段により前記訓練用データのクラスと同一であると判断された場合に値を減少させる使用確率変数を、使用確率変更手段が前記訓練用データ毎に付与する使用確率変更ステップと、テスト関数生成手段が、前記分類データ修正手段により修正がなされた分類データのデータ情報とクラス情報とに基づいて入力データの分類を行う多変数テスト関数を生成するテスト関数生成ステップとを有し、前記最近傍分類データ検出ステップにおいて、前記使用確率変数の値が所定値以上を示す前記訓練用データのみを用いて前記最近傍分類データ検出手段が前記分類データの検出を行うことを特徴とする。

10

**【0021】**

本発明に係る多変数テスト関数を生成するためのプログラムは、複数要素の要素データを有する複数の入力データを複数のクラスに分類するための多変数テスト関数を生成するために、コンピュータに、予め要素データと分類されるべきクラスとが既知である入力データからなる訓練用データを、訓練用データ取得手段により取得させる訓練用データ取得ステップと、該訓練用データ取得ステップにおいて取得された前記要素データの要素数に基づいて、最近傍分類データ検出手段により当該要素数に対応する複数次元の特徴空間を構成させ、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させるとともに、前記要素データに対応するデータ情報と前記クラスを示すクラス情報とを有する複数の分類データを、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させ、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分類データを求めさせる最近傍分類データ検出ステップと、該最近傍分類データ検出ステップにおいて求められた分類データのクラス情報と前記訓練用データのクラスとが同一クラスであるか否かを、クラス判断手段により判断させるクラス判断ステップと、該クラス判断ステップにおいて前記分類データのクラス情報と前記訓練用データのクラスとが異なるクラスであると判断された場合に、分類データ修正手段により当該分類データのデータ情報を前記要素データの空間座標から遠ざかるように修正させるとともに、前記クラス判断ステップにおいて前記訓練用データと同一クラスであると判断される他の分類データを前記最近傍分類データ検出ステップにより検出させ、当該他の分類データのデータ情報を前記要素データの空間座標に近づくように修正させる分類データ修正ステップと、前記最近傍分類データ検出ステップにおいて検出された最近傍となる前記分類データのクラス情報が前記クラス判断ステップにおいて前記訓練用データのクラスと同一であると判断された場合に値を減少させる使用確率変数を、使用確率変更手段により前記訓練用データ毎に付与させる使用確率変更ステップと、前記分類データ修正ステップにおいて修正がなされた分類データのデータ情報とクラス情報とに基づいて入力データの分類を行う多変数テスト関数を、テスト関数生成手段により生成させるテスト関数生成ステップとを実行させ、さらに、前記最近傍分類データ検出ステップにおいて、前記使用確率変数の値が所定値以上を示す前記訓練用データのみを用いて前記最近傍分類データ検出手段に前記分類データの検出を行わせることを特徴とする。

20

30

40

**【0022】**

また、本発明に係る多変数テスト関数を生成するためのプログラムは、複数要素の要素データを有する複数の入力データを複数のクラスに分類するための多変数テスト関数を生成するために、コンピュータに、予め要素データと分類されるべきクラスとが既知である入力データからなる訓練用データを、訓練用データ取得手段により取得させる訓練用データ取得ステップと、該訓練用データ取得ステップにおいて取得された前記要素データの要素数に基づいて、最近傍分類データ検出手段により当該要素数に対応する複数次元の特徴

50

空間を構成させ、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させるとともに、前記要素データに対応するデータ情報と前記クラスを示すクラス情報とを有する複数の分類データを、前記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させ、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分類データを求める最近傍分類データ検出ステップと、該最近傍分類データ検出ステップにおいて求められた分類データのクラス情報と前記訓練用データのクラスとが同一クラスであるか否かを、クラス判断手段により判断させるクラス判断ステップと、該クラス判断ステップにおいて前記分類データのクラス情報と前記訓練用データのクラスとが同一クラスであると判断された場合に、分類データ修正手段により当該分類データのデータ情報を前記要素データの空間座標に近づくように修正させる分類データ修正ステップと、前記最近傍分類データ検出ステップにおいて検出された最近傍となる前記分類データのクラス情報が前記クラス判断ステップにおいて前記訓練用データのクラスと同一であると判断された場合に値を減少させる使用確率変数を、使用確率変更手段により前記訓練用データ毎に付与させる使用確率変更ステップと、前記分類データ修正ステップにおいて修正がなされた分類データのデータ情報とクラス情報とに基づいて入力データの分類を行う多変数テスト関数を、テスト関数生成手段により生成させるテスト関数生成ステップとを実行させ、さらに、前記最近傍分類データ検出ステップにおいて、前記使用確率変数の値が所定値以上を示す前記訓練用データのみを用いて前記最近傍分類データ検出手段に前記分類データの検出を行わせることを特徴とする。

10

#### 【発明の効果】

20

#### 【0023】

本発明に係る多変数テスト関数生成装置、多変数テスト関数生成システム、多変数テスト関数生成方法および多変数テスト関数を生成するためのプログラムを用いることによって、各訓練用データに対して使用確率変数を付与し、最近傍の分類データ検出において検出された前記分類データのクラス情報が訓練用データのクラスと同一であると判断された場合、つまり最近傍となる分類データにより正しくクラスの分類が行われた場合に、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のみ、つまり誤判断率の高い訓練用データのみを繰り返し用いて分類データ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、効率よく分類データの修正（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分類データ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。

30

#### 【0024】

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、その分類データのクラス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。

#### 【0025】

40

さらに、本発明に係る多変数テスト関数生成装置等を用いることによって、多変数決定木の構築を高速に行うことが可能となる。

#### 【発明を実施するための最良の形態】

#### 【0026】

以下、本発明に係る多変数テスト関数生成システムを、図面を用いて説明する。図1は、本発明に係る多変数テスト関数生成システム1の概略構成を示したブロック図である。

#### 【0027】

多変数テスト関数生成システム1は、多変数テスト関数を生成するため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（訓練用データ記録手段）2と、多変数テスト関数を生成するための演算処理部3と、演算処理部3により作成された多変数テスト

50

関数を記録する多変数テスト関数記録部（多変数テスト関数記録手段）4とを有している。

【0028】

訓練用データ記録部2は、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訓練用データを読み出すことが可能な構成となっている。

【0029】

訓練用データは、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を示している。演算処理部3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、決定結果としてクラスCが求められるような多変数テスト関数を生成する。つまり、演算処理部3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算出する多変数テスト関数を生成する。

【0030】

多変数テスト関数記録部4も、訓練用データ記録部2と同様に、メモリ、ハードディスク、フレキシブルディスク、光学記録装置等のデータを記録・読み出し可能な記録装置で構成され、演算処理部3によって作成された多変数テスト関数を格納する構成となっている。

【0031】

なお、説明の便宜上、訓練用データ記録部2と多変数テスト関数記録部4とを別々の記録装置として図示したが、両方を同一の記録装置によって構成しても良い。さらに、訓練用データ記録部2や多変数テスト関数記録部4は、必ずしも物理的に演算処理部3に繋がっているものである必要はなく、訓練用データ記録部2は、演算処理部3に対して訓練用データを出力することができ、多変数テスト関数記録部4は、演算処理部3により作成された多変数テスト関数を受信して記録することができれば良いので、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良い。

【0032】

演算処理部3は、計算・制御処理全般を司るCPU、演算処理において必要なデータを一時的に記録するRAM（図1に示す分類データ記録機能5としての役割）、CPUにおける演算処理をプログラムとして記録するROM等を備える。なお、これらのRAMやROM等は、上述した訓練用データ記録部2や多変数テスト関数記録部4等に用いられる記録装置と兼用するものであっても良い。

【0033】

演算処理部3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。「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（1）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。本発明に係る多変数テスト関数生成システムで生成するNNCは、人間らしい判断が可能な多変数テスト関数である。

【0034】

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、分類を行うための訓練用データ（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、特徴空間における空間座標として表すことができる。また、各プロトタイプはクラス（NNCを用いて非終端節点において分類（分割）されるべきグループの情報、分類結果）を備えており、こ

10

20

30

40

50

の点で、プロトタイプは既知のデータであるともいえる。

【0035】

未知のデータXを分類する場合、演算処理部3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と同じクラスに分類する。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（通常、ユークリッド距離を使うが、他の距離を利用してかまわない）Dによって求める。特徴空間の次元はNdとして、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2】

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

10

・・・・・・(2)

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。

【0036】

図2は、式2により入力データ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示した図であり、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。入力データX=(0.1, 0.8)とし、プロトタイプYとしてP1~P4の4つの既知のタイプを用いる。なお、P1とP4とはクラス1、P2とP3とはクラス0のクラスを備えるものとする。

20

【0037】

まず、演算処理部3は、入力データXと全てのプロトタイプP1~P4との距離を求める。図2から明らかなように、入力データ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、入力データXはプロトタイプP1と同じクラス1に属することとなり、演算処理部3は、入力データXはクラス1という分類結果(決定結果)を判断する。

【0038】

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、入力データとプロトタイプとの2点間距離(類似度)に基づいてクラスを判断(分類)することができる。即ち、未知の入力データXが前例(プロトタイプY)に似ていれば、入力データXはその前例(プロトタイプY)と同じクラス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。

30

【0039】

次に、演算処理部3において、NNCを生成する方法をより詳細に説明する。

40

【0040】

まず、本実施形態において演算処理部3により作成するNNCは、予め作成されるNNCのサイズ(プロトタイプの数)とプロトタイプのクラス(ここでのクラスとはNNCにより分類されるグループのラベルを示す)が既知のものとする。NNCのサイズとプロトタイプのクラスとが既知のものであれば、サイズとクラスが決まっているものよりも速くNNCを構築することができる。ただし、サイズとクラスが既知のものでない場合には、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クラスを決めるか又は各クラスに同じ数のプロトタイプを割り振る方法が用いられる。このようにしてサイズを仮定し、クラス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。

50

## 【 0 0 4 1 】

NNCを改善して精度を向上させるために、演算処理部3は複数エポック（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）訓練用データを訓練用データ記録部2より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。演算処理部3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の生成を完了する。

## 【 0 0 4 2 】

また、演算処理部3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、 $0 < \alpha < 1$ の初期値を取り、更新により徐々に減少する値である。

10

## 【 0 0 4 3 】

演算処理部3が、プロトタイプの修正（更新）を行う場合、まず演算処理部3は、入力データ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クラスと入力データXのクラスとを比較する。プロトタイプP0のクラスと入力データXのクラス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、次の入力データを読み取り同様の処理を続ける。プロトタイプP0のクラスと入力データXのクラスとが異なる場合、演算処理部3は、最近傍のプロトタイプP0以外のプロトタイプとして、入力データXのクラスと同じクラスを持つプロトタイプの中から入力データXに最も近いプロトタイプP1を求める。そして、演算処理部3は、プロトタイプP0とプロトタイプP1とを、

20

$$P0^{new} = P0^{old} - \alpha (X - P0^{old}) \quad \dots \dots (3)$$

$$P1^{new} = P1^{old} + \alpha (X - P1^{old}) \quad \dots \dots (4)$$

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

## 【 0 0 4 4 】

(3)式は、プロトタイプP0を入力データ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 $\alpha$ 倍だけ入力データXの空間位置より遠ざける計算式を示し、(4)式は、プロトタイプP1を入力データ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 $\alpha$ 倍だけ入力データXの空間位置に近づける計算式を示している。

## 【 0 0 4 5 】

このように、1つの入力データXを用いて、クラスの等しいプロトタイプP1が入力データXに近づくようにプロトタイプP1の修正を行うと共に、クラスの異なるプロトタイプP0が入力データ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の認識（分類）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。

30

## 【 0 0 4 6 】

また演算処理部3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。

## 【 0 0 4 7 】

具体的に演算処理部3は、入力データXの使用確率p(X)の初期値をp(X) = 1とし、入力データ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入力データXのクラスと等しい場合）に、

40

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

となるように使用確率p(X)を更新する。ただし、 $\alpha$ は $0 < \alpha < 1$ の定数である。

## 【 0 0 4 8 】

演算処理部3がプロトタイプの修正（更新）を行う場合、演算処理部3がある入力データ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、入力データ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NNCの生成において演算処理部3における処理負担の重い計算は、入力データとプロトタイプとの距離を求める計算である。

50

このため、使用確率を導入することによって、正しく認識されやすい入力データの使用を少なくし、正しく認識されにくい入力データだけに着目して距離計算を行うことによって、演算処理部3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。

【0049】

次に、フローチャートを用いて、演算処理部3におけるNNCの生成方法を説明する。図3は、演算処理部3におけるNNCの生成過程を示したフローチャートである。

【0050】

まず演算処理部3は、初期設定を行う(ステップS1)。演算処理部3は、全て( $n$ 個)の入力データの使用率 $p(i)$ (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。

10

【0051】

続いて演算処理部3は、入力データ $X$ の番号を示す変数 $i$ に1を代入し(ステップS2)、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S3)。そして、演算処理部3は、 $i$ 番目の入力データ $X(i)$ 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S4)。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 $p(i)$ の入力データ $X(i)$ 、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 $p(i)$ の入力データ $X(i)$ 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。

【0052】

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良いが、数エポック(このフローチャートにおいては $K$ エポック)回だけ入力データ $X(i)$ 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入力データ $X(i)$ 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)に使用される入力データが偏ってしまうことを防止することができる。

20

【0053】

$i$ 番目の入力データ $X(i)$ の使用確率 $p(i)$ が乱数 $r$ よりも小さい場合(ステップS4においてNoの場合)、演算処理部3は、プロトタイプの更新を行うことなく、次述する変数 $i$ が入力データ数 $n$ よりも小さいか否かの判断を行う処理(ステップS11)へ移行する。

30

【0054】

入力データ $X(i)$ の使用確率 $p(i)$ が乱数 $r$ よりも大きい場合(ステップS4においてYesの場合)、演算処理部3は、入力データ $X(i)$ 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S5)。そして演算処理部3は、求められたプロトタイプ $Y(j_1)$ と入力データ $X(i)$ とのクラスが同じか否かの判断を行う(ステップS6)。

【0055】

プロトタイプ $Y(j_1)$ と入力データ $X(i)$ とのクラスが同じである場合(ステップS6においてYesの場合)、演算処理部3は、入力データ $X(i)$ の最近傍のプロトタイプにより求められるクラスが入力データ $X(i)$ のクラスとして最適なクラスであるため、NNCにより適正に入力データ $X(i)$ が分類されたものと判断する。そして演算処理部3は、入力データ $X(i)$ の使用確率 $p(i)$ に対して  $\alpha$  を掛け合わせることによって( $p(i) = \alpha \cdot p(i)$ )、使用確率 $p(i)$ がより小さい値となるように修正を行い(ステップS7)、次述するステップ11へ処理を進める。

40

【0056】

プロトタイプ $Y(j_1)$ と入力データ $X(i)$ とのクラスが異なる場合(ステップS6においてNoの場合)、演算処理部3は、入力データ $X(i)$ の最近傍のプロトタイプにより求められるクラスが入力データ $X(i)$ のクラスと異なるクラスであるため、NNC

50



により誤って入力データ  $X(i)$  が分類されたものと判断する。そして演算処理部 3 は、入力データ  $X(i)$  の使用確率  $p(i)$  に 1 を代入する (ステップ S 8)。使用確率  $p(i)$  に 1 を代入することにより、次にこの入力データ  $X(i)$  が使用される場合には、ステップ S 4 において Yes と判断され、確実にプロトタイプの更新 (変更) に使用されることとなる。

【0057】

その後、演算処理部 3 は、入力データ  $X(i)$  と同じクラスを持つプロトタイプである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 S 9)。そして、演算処理部 3 は、プロトタイプ  $Y(j_1)$  とプロトタイプ  $Y(j_2)$  とを、

$$Y(j_1) = Y(j_1) - (X(i) - Y(j_1))$$

$$Y(j_2) = Y(j_2) + (X(i) - Y(j_2))$$

に修正 (更新) し、NNC の判断精度の向上を図る (ステップ S 10)。

【0058】

そして、演算処理部 3 は、変数  $i$  が入力データの全数  $n$  よりも小さいか否かの判断を行う (ステップ S 11)。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入力データ  $X(i)$  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 S 12)、上述した乱数の発生処理 (ステップ S 3) からの処理を繰り返し実行する。

【0059】

変数  $i$  が  $n$  より小さくない場合には、全ての入力データ  $X$  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。

【0060】

そして演算処理部 3 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 S 13)。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 S 13 では、多変数テスト関数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。

【0061】

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 S 13 で Yes の場合)、演算処理部 3 は、変数  $k$  の値に 1 を追加した後 ( $k = k + 1$ ) (ステップ S 14)、ステップ S 2 に示す変数  $i$  に 1 を代入する処理以降の処理を繰り返し実行する。

【0062】

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 S 13 で No の場合)、演算処理部 3 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。演算処理部 3 は、これらの処理により求められたプロトタイプの座標位置とそのクラスとを基準として入力データに最適な分類結果 (クラス) を求める。つまり、演算処理部 3 は、プロトタイプのデータ情報 (= 座標位置を示す情報) とクラス情報とを基準として最適なクラスを求める演算処理関数を多変数テスト関数として生成する。このようにして生成された多変数テスト関数は、多変数テスト関数記録部 4 に格納される。生成された多変数テスト関数を利用する場合には、利用しようとするシステム、例えば、文字認識、音声認識、交通状況予測等に適した形式にデータ形式を変更して、各システムの判断に用いられる。

【0063】

以上説明したように、本発明に係る多変数テスト関数生成システムを用いることによって、訓練用データにおける入力データ  $X$  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クラス情報が、訓練データのクラス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

10

20

30

40

50

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入力データの分類を行うNNC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。

【0064】

また、入力データ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クラス情報に基づいて入力データ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。

【0065】

また、多変数テスト関数生成システムでは、各訓練用データ（入力データ）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クラス情報が訓練用データのクラス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クラス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のみを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。

10

【0066】

以上、本発明に係る多変数テスト関数生成システム1について、図面を用いて詳細に説明を行ったが、本発明に係る多変数テスト関数生成手段は、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。いわゆる当業者であれば、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を想定し得ることは明らかであり、それらについても当発明の技術的範囲に属するものと了解される。

20

【0067】

例えば、上記実施形態では、演算処理部3がプロトタイプの修正（更新）を行う場合、まず演算処理部3は、入力データ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<sub>1</sub>)のクラスと入力データX(i)のクラスとを比較し、プロトタイプY(j<sub>1</sub>)のクラスと入力データX(i)のクラスとが異なる場合にのみ新たなプロトタイプY(j<sub>2</sub>)を求めて(3)式、(4)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るものではない。

30

【0068】

図4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4に示すプロトタイプの修正方法は、図3のステップS9、ステップS10に示す処理がなくなり、ステップS6とステップS7との間にステップS15に示す処理が追加される点で相違する。

【0069】

図4に示す処理では、入力データX(i)の最近傍となるプロトタイプY(j<sub>1</sub>)のクラスと入力データX(i)のクラスとを比較し、プロトタイプY(j<sub>1</sub>)のクラスと入力データX(i)のクラスとが同じクラスの場合（ステップS6でYesの場合）に、プロトタイプY(j<sub>1</sub>)のデータ情報を、

40

$$Y(j_1) = Y(j_1) + (X(i) - Y(j_1))$$

に修正する（ステップS15）。このように、同一クラスとなるプロトタイプ(j<sub>1</sub>)が入力データ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（収束速度）を向上させることが可能となる。

【0070】

また、上述したように、同一クラスとなる最近傍のプロトタイプを入力データXに近づくように修正するだけでなく、異なるクラスであって最近傍となるプロトタイプを求めて

50

、求められたプロトタイプを入力データXより遠ざかるように修正してもよい。このように修正を行うことによって、上述と同様に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(収束速度)を向上させることが可能となる。

【0071】

さらに、本発明に係る発明は、上述した多変数テスト関数生成システム1に限定されるものではなく、上述した多変数テスト関数生成システム1を構成する演算処理部3部分を多変数テスト関数生成装置として抽出したものであっても、本発明に係る効果を奏することができ、さらに、上述した多変数テスト関数を生成するために演算処理部が上記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テスト関数生成方法も同様に本発明に含まれるものである。

10

【0072】

また、上述した実施形態では、演算処理部3が、図1に示すように、訓練用データを取得するための訓練用データ取得手段としての機能(訓練用データ取得機能6)や、最近傍となる分類データ(プロトタイプ)を求める最近傍分類データ検出手段としての機能(最近傍分類データ検出機能7)や、取得した分類データ(プロトタイプ)と訓練用データ(入力データ)とのクラスの同一性を判断するクラス判断手段としての機能(クラス判断機能8)や、クラスが同一か否かによって分類データ(プロトタイプ)の修正を行う分類データ修正手段としての機能(分類データ修正機能9)や、使用確率変数を各訓練用データに付与する使用確率変更手段としての機能(使用確率変更機能10)や、修正が行われた分類データ(プロトタイプ)を用いて多変数テスト関数を生成するテスト関数生成手段としての機能(テスト関数生成機能11)を果たすこととしたが、必ずしも全ての機能を1つの演算処理部が処理する必要はなく、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良いし、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上述した処理を行うような構成としても良い。

20

【0073】

さらに、本発明は、多変数テスト関数生成装置、多変数テスト関数生成システム、多変数テスト関数生成方法および多変数テスト関数を生成するためのプログラムに関するものであるが、本発明に係るこれらの装置などは、多変数決定木の構築のみならず、通常の最近傍識別器(NNC)の設計及びNNCに基づく各種ニューラルネットワークの学習に利用することも可能である。

30

【図面の簡単な説明】

【0074】

【図1】本発明に係る多変数テスト関数生成システムを示したブロック図である。

【図2】演算制御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3】演算制御部がNNCを生成する過程を示したフローチャートである。

【図4】演算制御部が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5】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6】図5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。

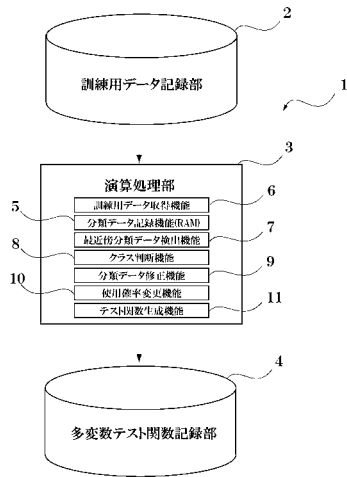
40

【符号の説明】

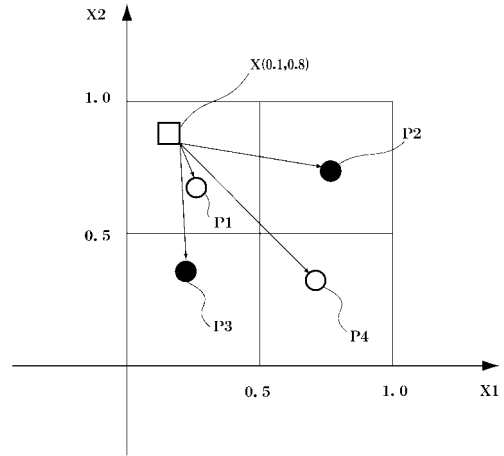
【0075】

- 1 ... 多変数テスト関数生成システム
- 2 ... 訓練用データ記録部
- 3 ... 演算処理部(多変数テスト関数生成装置、訓練用データ取得手段、分類データ記録手段、最近傍分類データ検出手段、クラス判断手段、分類データ修正手段、使用確率変更手段、テスト関数生成手段)
- 4 ... 多変数テスト関数記録部

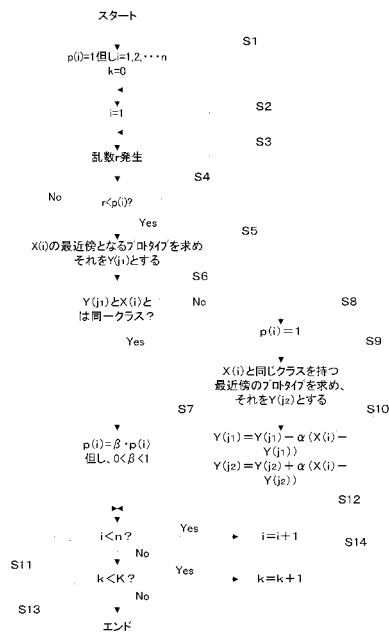
【 図 1 】



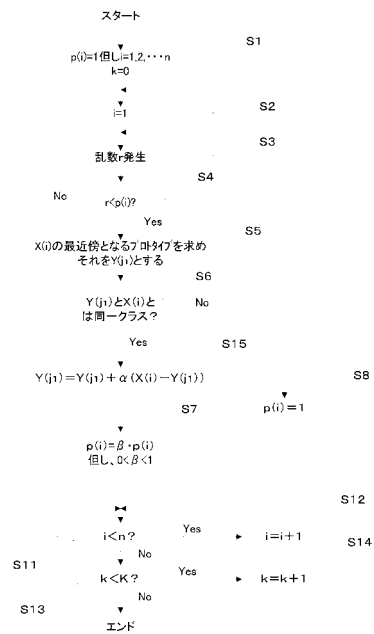
【 図 2 】



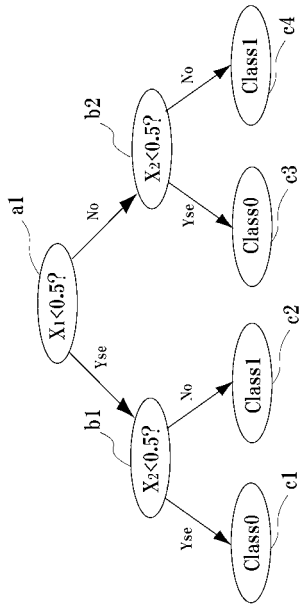
【 図 3 】



【 図 4 】



【 図 5 】



【 図 6 】

