

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5284530号
(P5284530)

(45) 発行日 平成25年9月11日(2013.9.11)

(24) 登録日 平成25年6月7日(2013.6.7)

(51) Int.Cl. F I
GO6N 3/00 (2006.01) GO6N 3/00 560A
GO6T 7/00 (2006.01) GO6T 7/00 350C

請求項の数 25 (全 32 頁)

(21) 出願番号	特願2012-256706 (P2012-256706)	(73) 特許権者	000001007
(22) 出願日	平成24年11月22日(2012.11.22)		キヤノン株式会社
(62) 分割の表示	特願2007-130898 (P2007-130898) の分割		東京都大田区下丸子3丁目30番2号
原出願日	平成19年5月16日(2007.5.16)	(74) 代理人	100076428
(65) 公開番号	特開2013-65336 (P2013-65336A)		弁理士 大塚 康德
(43) 公開日	平成25年4月11日(2013.4.11)	(74) 代理人	100112508
審査請求日	平成24年11月22日(2012.11.22)		弁理士 高柳 司郎
		(74) 代理人	100115071
			弁理士 大塚 康弘
		(74) 代理人	100116894
			弁理士 木村 秀二
		(74) 代理人	100130409
			弁理士 下山 治
		(74) 代理人	100134175
			弁理士 永川 行光

最終頁に続く

(54) 【発明の名称】 情報処理方法、情報処理装置

(57) 【特許請求の範囲】

【請求項1】

情報処理装置が行う情報処理方法であって、
入力手段が、それぞれ異なるクラスに属する処理データを入力する入力工程と、
第1の計算手段が、前記入力工程で入力したそれぞれの処理データ間の測地線距離関係を求める第1の計算工程と、
設定手段が、前記クラス間の距離を求め、前記クラス間の距離が所定の値よりも小さい場合に、前記クラス間の距離が大きくなるように、クラス間分離度を設定する設定工程と、
更新手段が、前記処理データのそれぞれが属するクラスの前記設定されたクラス間分離度に基づいて、同じクラスに属する処理データ間の測地線距離が、他クラスに属する処理データとの測地線距離よりも小さくなるように、前記処理データ間の測地線距離関係を更新する更新工程と、
第2の計算手段が、前記更新工程で更新された測地線距離関係を用いて、ユークリッド距離関係を近似するデータ写像関係を示す情報を求める第2の計算工程と
を備えることを特徴とする情報処理方法。

【請求項2】

2つの処理データ x_1 、 x_2 間のグラフ距離 $d_G(x_1, x_2)$ が、 x_1 、 x_2 が近傍で無い場合は であるとした時、

2つの処理データ 、 間の測地線距離 $d_M(、)$ は、 $d_G(、)$ か、当該処

理データとは異なる処理データ a を経由する $dG(\quad, a) + dG(a, \quad)$ の、何れか小さい方であることを特徴とする請求項 1 に記載の情報処理方法。

【請求項 3】

前記 2 つの処理データ x_1 、 x_2 間のグラフ距離 $dG(x_1, x_2)$ が、 x_1 、 x_2 が近傍である場合、ユークリッド距離であることを特徴とする請求項 2 に記載の情報処理方法。

【請求項 4】

前記 2 つの処理データ x_1 、 x_2 間のグラフ距離 $dG(x_1, x_2)$ が、 x_1 、 x_2 が近傍である場合、ミンコフスキー距離またはマハラノビス距離であることを特徴とする請求項 2 に記載の情報処理方法。

【請求項 5】

前記設定工程では、予め定義されたクラス間の分離度に応じて前記クラス間分離度を設定することを特徴とする請求項 1 乃至 4 の何れか 1 項に記載の情報処理方法。

【請求項 6】

前記設定工程では、最短距離法、最長距離法、群平均法、重心法、メジアン法、ワード法、可変法のいずれかを用いて前記クラス間の距離を求めることを特徴とする請求項 1 乃至 5 の何れか 1 項に記載の情報処理方法。

【請求項 7】

前記 2 つの処理データ x_1 、 x_2 が近傍であるか否かは、処理データ x_1 から距離の近いものから順番に予め設定された個数までの処理データに処理データ x_2 が存在する場合に、前記 2 つのデータ x_1 、 x_2 が近傍であると判定することを特徴とする請求項 2 乃至 4 の何れか 1 項に記載の情報処理方法。

【請求項 8】

前記 2 つの処理データ x_1 、 x_2 が近傍であるか否かは、前記 2 つの処理データ x_1 、 x_2 間の距離が予め設定された距離以内である場合に、前記 2 つの処理データ x_1 、 x_2 が近傍であると判定することを特徴とする請求項 2 乃至 4 の何れか 1 項に記載の情報処理方法。

【請求項 9】

前記更新工程では、同クラスに属する処理データとの距離に 1 より小さい正の数乗することを特徴とする請求項 1 乃至 8 の何れか 1 項に記載の情報処理方法。

【請求項 10】

前記更新工程では、2 つの処理データのそれぞれが属するクラスの前記クラス間分離度に比例して処理データ間の距離を更新するとともに、同クラスに属する処理データとの距離を、他クラスに属する処理データとの距離よりも小さくなるような正の数とすることを特徴とする請求項 1 乃至 8 の何れか 1 項に記載の情報処理方法。

【請求項 11】

前記第 2 の計算工程では、写像後の空間における距離関係と更新後の距離関係の誤差を最小にする写像を求めることを特徴とする請求項 1 乃至 10 の何れか 1 項に記載の情報処理方法。

【請求項 12】

前記第 2 の計算工程では、多次元尺度法を用いて処理データの写像後の対応関係を求めることを特徴とする請求項 1 乃至 3 の何れか 1 項に記載の情報処理方法。

【請求項 13】

前記第 2 の計算工程では、求めた前記対応関係を教師データとしてトレーニングしたニューラルネットワークを構築することを特徴とする請求項 12 に記載の情報処理方法。

【請求項 14】

前記ニューラルネットワークは、多層フィードフォワード型ニューラルネットワークであることを特徴とする請求項 13 に記載の情報処理方法。

【請求項 15】

前記第 2 の計算工程では、写像後の空間における距離関係と更新後の距離関係の誤差を

10

20

30

40

50

最小にする線形写像を求めることを特徴とする請求項 1 に記載の情報処理方法。

【請求項 16】

前記第 2 の計算工程では、 i 番目、 j 番目の処理データを x_i 、 x_j とし、 i 番目、及び j 番目の処理データ間の更新後の距離関係を $d_d(i, j)$ とした時、

$$J(A) = \frac{1}{2} \cdot \frac{1}{\sum_{i,j} d_d(i, j)} \sum_{i,j} \frac{\left(\left| A^T x_i - A^T x_j \right| - d_d(i, j) \right)^2}{d_d(i, j)}$$

なる誤差関数 $J(A)$ を最小化する線形写像行列 A を求めることを特徴とする請求項 15 に記載の情報処理方法。

10

【請求項 17】

前記第 2 の計算工程では、写像後の空間における距離関係と更新後の距離関係の誤差を最小にする非線形写像 (x) を求め、当該非線形写像 (x) は、半正定値性を満たす実対称関数であるカーネル関数 $K(\cdot, \cdot)$ と、処理データ x_i ($i = 1, 2, \dots$) 用いて、 $(x) = \sum_i x_i \cdot K(x, x_i)$ と表されることを特徴とする請求項 1 に記載の情報処理方法。

【請求項 18】

前記第 2 の計算工程では、ベクトル κ_i を、処理データ x_i に対する処理データ x_j との前記カーネル関数値 $K(x_i, x_j)$ を j 番目の要素とするベクトルとし、処理データ x_i と処理データ y_i と間の更新後の距離関係を $d_d(i, j)$ とした時、

20

$$J(\Gamma) = \frac{1}{2} \cdot \frac{1}{\sum_{i,j} d_d(i, j)} \sum_{i,j} \frac{\left(\left| \Gamma^T \kappa_i - \Gamma^T \kappa_j \right| - d_d(i, j) \right)^2}{d_d(i, j)}$$

なる誤差関数 $J(\Gamma)$ を最小化する行列 Γ を求め、求めた行列 Γ の i 行目の行ベクトルを、前記 κ_i とすることを特徴とする請求項 17 に記載の情報処理方法。

【請求項 19】

前記第 2 の計算工程では、ベクトル κ_i を、処理データ x_i に対する処理データ x_j との前記カーネル関数値 $K(x_i, x_j)$ を j 番目の要素とするベクトルとし、処理データ x_i と処理データ y_i と間の更新後の距離関係を $d_d(i, j)$ 、 λ を正の定数、 $\|\cdot\|_{L1}$ を行列 Γ の k 番目の列ベクトルの $L1$ ノルムとした時、

30

$$J(\Gamma) = \frac{1}{2} \cdot \frac{1}{\sum_{i,j} d_d(i, j)} \sum_{i,j} \frac{\left(\left| \Gamma^T \kappa_i - \Gamma^T \kappa_j \right| - d_d(i, j) \right)^2}{d_d(i, j)} + \lambda \cdot \sum_k \|\kappa_k\|_{L1}$$

なる誤差関数 $J(\Gamma)$ を最小化する行列 Γ を求め、求めた行列 Γ の i 行目の行ベクトルを、前記 κ_i とすることを特徴とする請求項 17 に記載の情報処理方法。

【請求項 20】

前記第 2 の計算工程では少なくとも、写像後の距離関係の順序が更新後の距離関係の順序を満たすことを特徴とする請求項 11 乃至 19 の何れか 1 項に記載の情報処理方法。

40

【請求項 21】

情報処理装置が行う情報処理方法であって、

入力手段が、それぞれ異なるクラスに属する処理データを入力する入力工程と、

第 1 の計算手段が、前記入力工程で入力したそれぞれの処理データ間の測地線距離関係を求める第 1 の計算工程と、

設定手段が、前記クラス間の距離を求め、前記クラス間の距離が所定の値よりも小さい場合に、前記クラス間の距離が大きくなるように、クラス間分離度を設定する設定工程と、

更新手段が、前記処理データのそれぞれが属するクラスの前記設定されたクラス間分離度に基づいて、同じクラスに属する処理データ間の測地線距離が、他クラスに属する処理

50

データとの測地線距離よりも小さくなるように、前記処理データ間の測地線距離関係を更新する更新工程と、

第2の計算手段が、前記更新工程で更新された測地線距離関係を用いて、ユークリッド距離関係を近似するデータ写像関係を示す情報を求める第2の計算工程と、

生成手段が、前記データ写像関係により写像される空間において定義可能な前記それぞれ異なるクラスを識別する識別規則を示す情報を生成する生成工程と、

第2の入力手段が、識別対象データを入力する第2の入力工程と、

写像手段が、前記識別対象データを、前記第2の計算工程で求めた情報が示すデータ写像関係を用いて写像する写像工程と、

識別手段が、前記写像工程で写像されたデータと、前記生成工程で生成した情報が示す前記識別規則を用いて、前記識別対象データのラベルを識別する識別工程と

を備えることを特徴とする情報処理方法。

【請求項22】

コンピュータに請求項1乃至21の何れか1項に記載の情報処理方法を実行させるためのコンピュータプログラム。

【請求項23】

請求項22に記載のコンピュータプログラムを格納した、コンピュータ読み取り可能な記憶媒体。

【請求項24】

情報処理装置であって、

それぞれ異なるクラスに属する処理データを入力する入力手段と、

前記入力手段が入力したそれぞれの処理データ間の測地線距離関係を求める第1の計算手段と、

前記クラス間の距離を求め、前記クラス間の距離が所定の値よりも小さい場合に、前記クラス間の距離が大きくなるように、クラス間分離度を設定する設定手段と、

前記処理データのそれぞれが属するクラスの前記設定されたクラス間分離度に基づいて、同じクラスに属する処理データ間の測地線距離が、他クラスに属する処理データとの測地線距離よりも小さくなるように、前記処理データ間の測地線距離関係を更新する更新手段と、

前記更新手段によって更新された測地線距離関係を用いて、ユークリッド距離関係を近似するデータ写像関係を示す情報を求める第2の計算手段と

を備えることを特徴とする情報処理装置。

【請求項25】

情報処理装置であって、

それぞれ異なるクラスに属する処理データを入力する入力手段と、

前記入力手段が入力したそれぞれの処理データ間の測地線距離関係を求める第1の計算手段と、

前記クラス間の距離を求め、前記クラス間の距離が所定の値よりも小さい場合に、前記クラス間の距離が大きくなるように、クラス間分離度を設定する設定手段と、

前記処理データのそれぞれが属するクラスの前記設定されたクラス間分離度に基づいて、同じクラスに属する処理データ間の測地線距離が、他クラスに属する処理データとの測地線距離よりも小さくなるように、前記処理データ間の測地線距離関係を更新する更新手段と、

前記更新手段によって更新された測地線距離関係を用いて、ユークリッド距離関係を近似するデータ写像関係を示す情報を求める第2の計算手段と、

前記データ写像関係により写像される空間において定義可能な前記それぞれ異なるクラスを識別する識別規則を示す情報を生成する生成手段と、

識別対象データを入力する第2の入力手段と、

前記識別対象データを、前記第2の計算手段で求めた情報が示すデータ写像関係を用いて写像する写像手段と、

10

20

30

40

50

前記写像手段によって写像されたデータと、前記生成手段が生成した情報が示す識別規則を用いて、前記識別対象データのラベルを識別する識別手段と

を備えることを特徴とする情報処理装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、冗長性を削減した高効率なデータの表現技術、このデータを用いた処理技術に関するものである。

【背景技術】

【0002】

近年、非特許文献1のIsomapや、非特許文献2のLocally Linear Embedding (LLE) に代表される、非線形の次元圧縮手法が提案されている。これらは、高次元空間内で、より低次元の超曲面 (Manifold) 上にあると考えられるデータを、Manifold固有の表面形状が許容できる程度に保存された、新たな低次元の空間に写像する手法を提供する。

【0003】

係る手法は、より低次元の空間でデータを表現できるという意味で、高効率なパターン表現には成功している。しかし、データが何れのクラスに属するかという情報は用いておらず、データの分類を効率的に表すという点では、最適であるとは言えない。

【0004】

これに対し、特許文献1に開示されている手法では、カーネルフィッシャー線形識別関数、またはフィッシャー線形判別関数を用いて、従来のIsomap法を拡張することにより、パターン分類のための画像を表す手法を提供している。

【0005】

また、非特許文献3においては、従来のIsomap法の改良として、他クラスに属するデータ間の測地線距離を強制的に増加させることにより、クラス間の分離度を高める写像を構築する手法が提案されている。

【0006】

このように、Manifold固有の表面形状を許容できる程度に保存し、且つパターン分類のためのデータを表現できる方法が望まれている。

【特許文献1】特表2005-535017号広報

【非特許文献1】Joshua B. Tenenbaum, Vin de Silva, John C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction", Science, Vol. 290, pp. 2319-2323, 2000

【非特許文献2】Sam T. Roweis, Lawrence K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding", Science, Vol. 290, pp. 2323-2326, 2000

【非特許文献3】Bisser Raytchev, Ikushi Yoda, Katsuhiko Sakaue, "Multi-View Face Recognition By Nonlinear Dimensionality Reduction And Generalized Linear Models", Proceedings of the 7th International Conference on Automatic Face Gesture Recognition, pp. 625-630, 2006

【発明の概要】

【発明が解決しようとする課題】

【0007】

本発明は以上の問題に鑑みてなされたものであり、Manifold固有の表面形状を、許容できる程度に保存し、且つパターン分類に適した、データ表現方法、及びその表現方法を利用した、パターン識別方法に係る技術を提供することを目的とする。

【0008】

より具体的には、あるクラスにラベル付けされた複数のデータにおいて、同一のクラスのデータ集合が1つのクラスタとして表現され、且つ各クラスタ間の距離を所望の距離に設定可能なデータ表現方法、及びその表現方法を利用したパターン識別方法である。

10

20

30

40

50

【課題を解決するための手段】

【0009】

本発明の目的を達成するために、例えば、本発明の情報処理方法は以下の構成を備える。

【0010】

即ち、情報処理装置が行う情報処理方法であって、

入力手段が、それぞれ異なるクラスに属する処理データを入力する入力工程と、

第1の計算手段が、前記入力工程で入力したそれぞれの処理データ間の測地線距離関係を求める第1の計算工程と、

設定手段が、前記クラス間の距離を求め、前記クラス間の距離が所定の値よりも小さい場合に、前記クラス間の距離が大きくなるように、クラス間分離度を設定する設定工程と、

更新手段が、前記処理データのそれぞれが属するクラスの前記設定されたクラス間分離度に基づいて、同じクラスに属する処理データ間の測地線距離が、他クラスに属する処理データとの測地線距離よりも小さくなるように、前記処理データ間の測地線距離関係を更新する更新工程と、

第2の計算手段が、前記更新工程で更新された測地線距離関係を用いて、ユークリッド距離関係を近似するデータ写像関係を示す情報を求める第2の計算工程と

を備えることを特徴とする。

【発明の効果】

【0011】

本発明の構成によれば、Manifold固有の表面形状を、許容できる程度に保存し、且つパターン分類に適した、データ表現方法、及びその表現方法を利用した、パターン識別方法に係る技術を提供することができる。

【図面の簡単な説明】

【0012】

【図1】人物の顔の領域についてラベル付け（ラベリング）がなされたパターン画像を用いたデータ表現処理を行う情報処理装置の機能構成例を示すブロック図である。

【図2】パターン画像を用いて行うデータ表現処理のフローチャートである。

【図3】近似的な測地線距離算出の処理を示すフローチャートである。

【図4】本発明の第2の実施形態に係る情報処理装置を構成する各部のうち、学習モード時において動作する各部についてのみ、その機能構成を示したブロック図である。

【図5】本発明の第2の実施形態に係る情報処理装置が学習モード時に行う処理のフローチャートである。

【図6】ステップS56において近似システム構築処理部46が行う処理のフローチャートである。

【図7】本発明の第2の実施形態に係る情報処理装置を構成する各部のうち、識別モード時において動作する各部についてのみ、その機能構成を示したブロック図である。

【図8】本発明の第2の実施形態に係る情報処理装置が識別モード時に行う処理のフローチャートである。

【図9】本発明の第3の実施形態に係る情報処理装置を構成する各部のうち、学習モード時において動作する各部についてのみ、その機能構成を示したブロック図である。

【図10】本発明の第3の実施形態に係る情報処理装置が学習モード時に行う処理のフローチャートである。

【図11】図1, 4, 7, 9に示した各部をコンピュータプログラムでもって実装した場合に、このコンピュータプログラムを実行するコンピュータのハードウェア構成例を示すブロック図である。

【発明を実施するための形態】

【0013】

先ず、本実施形態の概要について説明する。

【 0 0 1 4 】

本実施形態は、それぞれ異なるクラスに属する処理データと共に、処理データが属するクラスを示すラベルデータを入力すると、入力したそれぞれの処理データ間の距離関係を求める（第1の計算）。そして、クラス間のクラス間分離度を設定し、距離関係を示す情報を、ラベルデータと、クラス間分離度を示す情報と、に基づいて更新すると、更新された情報が示す距離関係を近似するデータ写像関係を示す情報を求める（第2の計算）。

【 0 0 1 5 】

ここで、求める距離関係とは測地線距離関係であり、データ写像関係を示す情報を用いて近似される距離関係はユークリッド距離関係である。

【 0 0 1 6 】

2つの処理データ x_1 、 x_2 間のグラフ距離 $dG(x_1, x_2)$ が、 x_1 、 x_2 が近傍で無い場合はであるとする。この時、2つの処理データ、間の測地線距離 $dM(、)$ は、 $dG(、)$ か、処理データとは異なる処理データ a を経由する $dG(、a) + dG(a、)$ の、何れか小さい方である。

【 0 0 1 7 】

ここで、グラフ距離とは、2つの処理データ x_1 、 x_2 間のグラフ距離 $dG(x_1, x_2)$ が、 x_1 、 x_2 が近傍である場合、ユークリッド距離や、マンハッタン距離といった、いわゆるミンコフスキー距離である。または、マハラノビス距離といった統計的な距離である。

【 0 0 1 8 】

クラス間分離度は、予め定義されたクラス間の分離度に応じて設定すればよい。このクラス間分離度は、分離度が大きい場合、つまり、ある2つのクラスの差異を強調したいなど、クラスの間隔を大きく分離して表現したい場合には、その2つのクラス間におけるクラス間分離度を大きくする。逆に、分離度が小さい場合、例えば、ある2つのクラスを明確に区別しなくてもよいような、クラスの間隔を大きく分離して表現しなくてもよい場合には、その2つのクラス間におけるクラス間分離度を小さくする。

【 0 0 1 9 】

また、このクラス間分離度は、クラス間の距離を求め、このクラス間の距離に基づいて、クラス間分離度を設定するようにしてもよい。この場合は、クラス間の距離が小さい時に、クラス間分離度を大きく設定するのが好適である。このようにすることにより、分離表現が比較的困難な2つのクラスを、効率よく分離表現することが可能になる。クラス間の距離を求める手法としては、クラスタ分析において一般的な、最短距離法や、最長距離法、群平均法、重心法、メジアン法、ワード法、可変法等を用いて、クラス間の距離を求めるようにすればよい。

【 0 0 2 0 】

また、2つの処理データ x_1 、 x_2 が近傍であるか否かは、処理データ x_1 から距離の近いものから順番に予め設定された個数までの処理データに処理データ x_2 が存在する場合に、2つのデータ x_1 、 x_2 が近傍であると判定することを特徴とする。または、2つの処理データ x_1 、 x_2 間の距離が予め設定された距離以内である場合に、2つの処理データ x_1 、 x_2 が近傍であると判定するようにしてもよい。

【 0 0 2 1 】

また、距離関係の更新では、2つの処理データのそれぞれが属するクラスのクラス間分離度に比例して処理データ間の距離を更新するとともに、同クラスに属する処理データとの距離が、他クラスに属する処理データとの距離よりも小さくなるように更新する。

【 0 0 2 2 】

同クラスに属する処理データとの距離が、他クラスに属する処理データとの距離よりも小さくなるように更新する方法としては、同クラスに属する処理データとの距離に1より小さい正の数に乗じる手法が挙げられる。また、同クラスに属する処理データとの距離を、他クラスに属する処理データとの距離よりも小さくなるような正の数とするようにしてもよい。

10

20

30

40

50

【 0 0 2 3 】

データ写像関係を求める手法は、写像後の空間における距離関係と更新後の距離関係の誤差を最小にする写像を求める手法である。このような手法として、多次元尺度法を用いて処理データの写像後の対応関係を求める手法が挙げられる。また、多次元尺度法を用いて処理データの写像後の対応関係を求め、その後、求めた対応関係を教師データとしてトレーニングしたニューラルネットワークを構築するような手法でもよい。このニューラルネットワークとしては、多層フィードフォワード型ニューラルネットワークを用いることができる。

【 0 0 2 4 】

また、データ写像関係を求める手法としては、写像後の空間における距離関係と更新後の距離関係の誤差を最小にする線形写像を求めるような手法でも構わない。このような手法としては次のようなものが挙げられる。即ち、 i 番目、 j 番目の処理データを x_i 、 x_j とし、 i 番目、及び j 番目の処理データ間の更新後の距離関係を $d_d(i, j)$ とした時、

$$J(A) = \frac{1}{2} \cdot \frac{1}{\sum_{i < j} d_d(i, j)} \sum_{i < j} \frac{(|A^T x_i - A^T x_j| - d_d(i, j))^2}{d_d(i, j)}$$

なる誤差関数 $J(A)$ を最小化する線形写像行列 A を求める手法が挙げられる。

【 0 0 2 5 】

また、データ写像関係を求める手法としては、写像後の空間における距離関係と更新後の距離関係の誤差を最小にする非線形写像 (x) を求める手法が挙げられる。ここで非線形写像 (x) は、半正定値性を満たす実対称関数であるカーネル関数 $K(\quad, \quad)$ と、処理データ x_i ($i = 1, 2, \dots$) 用いて、 $(x) = \sum_i x_i \cdot K(x, x_i)$ と表される。

【 0 0 2 6 】

この非線形写像 (x) は、ベクトル x_i を、処理データ x_i に対する処理データ x_j とのカーネル関数値 $K(x_i, x_j)$ を j 番目の要素とするベクトルとし、処理データ x_i と処理データ y_i と間の更新後の距離関係を $d_d(i, j)$ とした時、

$$J(\Gamma) = \frac{1}{2} \cdot \frac{1}{\sum_{i < j} d_d(i, j)} \sum_{i < j} \frac{(|\Gamma^T \kappa_i - \Gamma^T \kappa_j| - d_d(i, j))^2}{d_d(i, j)}$$

なる誤差関数 $J(\Gamma)$ を最小化する行列 Γ を求め、求めた行列 Γ の i 行目の行ベクトルを、 x_i とすることにより求められる。または、更に、 λ を正の定数、 $\|\kappa_k\|_{L1}$ を行列の k 番目の列ベクトルの $L1$ ノルムとした時、

$$J(\Gamma) = \frac{1}{2} \cdot \frac{1}{\sum_{i < j} d_d(i, j)} \sum_{i < j} \frac{(|\Gamma^T \kappa_i - \Gamma^T \kappa_j| - d_d(i, j))^2}{d_d(i, j)} + \lambda \cdot \sum_k \|\kappa_k\|_{L1}$$

なる誤差関数 $J(\Gamma)$ を最小化する行列 Γ を求め、求めた行列 Γ の i 行目の行ベクトルを、 x_i とするようにして求めてもよい。これらの写像を求める際に、少なくとも、写像後の距離関係の順序が、更新後の距離関係の順序を満たすようにしてもよい。

【 0 0 2 7 】

更に、パターン識別に適用するには、データ写像関係により写像される空間において定義可能なそれぞれ異なるクラスを識別する識別規則を示す情報を生成し、識別対象データを入力（第2の入力）して、生成した情報が示すデータ写像規則を用いて写像する。そして、写像されたデータと、生成した情報が示すデータ写像規則を用いて、識別対象データのラベルを識別する。

【 0 0 2 8 】

以下では、このような本発明の幾つかの実施形態について、添付図面を参照しながら詳細に説明する。なお、以下に説明する技術事項のうち幾つかを適宜選択し、適宜組み合わせて用いても良い。

【0029】

[第1の実施形態]

本実施形態で取り扱う画像は、人物の顔を含む原画像からこの顔の領域を切り出すことで得られる、縦20画素、横20画素のサイズの抽出画像（パターン画像）であって、グレースケール画像であるとする。もちろん、複数の人物の顔が原画像中に含まれている場合には、パターン画像（データ）は複数存在する。また、それぞれのパターン画像中の顔領域（パターン）には、誰の顔であるのかを示すラベルが付けられている（ラベリング処理済み）。即ち、本実施形態で取り扱うこのパターン画像には、このラベルのデータも含まれているものとして説明する。

10

【0030】

本実施形態では、係るパターンを新たな空間上にマッピングするデータ表現方法（技術）について説明する。

【0031】

縦横20画素のサイズの抽出画像は、各画素値をラスタスキャン的に要素として並べた $20 \times 20 = 400$ 次元のベクトルと見なせる。この場合、1つのパターンは、400次元空間内の1つの点となる。一般に、“人物の顔”といった特定のカテゴリであるパターンの集合は、400次元の空間に比べて、より低次元の超曲面（Manifold）を形成する。つまり、“人物の顔”を表現するには、400次元は冗長であり、より低い次元の空間で表現可能である。

20

【0032】

この冗長性を削減するための、最も一般的な手法として、主成分分析（PCA）を用いた手法がある。しかし、“人物の顔”のように、例えば顔の向きの変動等、本質的に非線形な変動を含むパターンの集合に対して、パターン分布が正規分布であることを仮定しているPCAでは、十分な冗長性削減を期待できない。

【0033】

そこで、非特許文献1で提案されているIsomapでは、先ず、上記非線形な超曲面上にある任意の2点のデータについて、超曲面に沿った、この2点のデータの最短距離である測地線距離を近似的に推定する。そして、多次元尺度法（Multidimensional Scaling: MDS）を用いて、すべてのデータの組み合わせについて推定した測地線距離関係を、ユークリッド距離として近似する写像を求める。これにより、本質的に非線形な分布の集合を、冗長性を削減した、よりコンパクトな空間で表現することが可能になる。

30

【0034】

しかし、PCAも同様であるが、Isomapでは、“人物の顔”というカテゴリをコンパクトな空間で表現することを目的としているため、必ずしも、例えば、その画像が誰であるかといった分類を表現するのに適した空間になるとは限らない。そこで本実施形態では、1つのカテゴリ内を細分化する、例えば人物の種別のような分類に適した空間での表現を、データに予め付加されたラベルを用いて行う。

40

【0035】

図1は、人物の顔の領域についてラベル付け（ラベリング）がなされたパターン画像を用いたデータ表現処理を行う情報処理装置の機能構成例を示すブロック図である。また、係る情報処理装置に係るパターン画像を用いて行うデータ表現処理のフローチャートを図2に示す。以下では、図1、2を用いて、係るデータ表現処理について説明する。なお、以下の説明は、パターン画像のサイズが他のサイズであっても、ラベリング対象が人物の顔以外であっても、実質的には同じである。

【0036】

ステップS20では、データ入力部10は、以上説明したような1以上のパターン画像

50

5を入力し、後段の画像正規化部11に転送する。データ入力部10によるパターン画像の入力は、インターネットなどのネットワークを介して外部から送信されたパターン画像を受信することで行うようにしても良い。また、ハードディスクドライブ装置などの大容量情報記憶装置に保存されているパターン画像を読み出すことで行うようにしても良い。何れにせよ、本情報処理装置にパターン画像を入力する形態については特に限定するものではない。

【0037】

ここで、 i 番目にデータ入力部10に入力されたパターン画像を構成する各画素をラスタスキャン的に並べた400次元のベクトルを $\sim x_i$ とし、そのラベルを y_i とする。従って、「N個のパターン画像をデータ入力部10に入力する」とは、 $\sim x_i$ と y_i のセットをNセットデータ入力部10に入力することに等価である($i = 1, 2, \dots, N$)。

10

【0038】

ラベルの値は、同一人物で同じ値になるように設定するのであれば、如何なる値であっても良い。ここでは説明を簡単にするために、 m 人($m > 1$)の種別が存在するデータセットを用いる場合、 $y_i \in \{1, 2, \dots, m\}$ と、 m 個のクラスラベルを用いて表現するようにすればよい。

【0039】

ここで、データ入力部10に入力されたデータの内、このラベルが同一であるデータの集合を、1つのクラスと定義する。また以下では、ラベルが c であるデータの集合をクラス c とする。

20

【0040】

次にステップS21では、画像正規化部11は、データ入力部10から転送されたデータのうち、パターン画像に対応する $\sim x_i$ を正規化した $x_i = (\sim x_i - u_i \cdot 1) / \sigma_i$ を、全てのパターン画像(i)について求める。そして、求めたそれぞれの x_i を保持しておく。

【0041】

ここで、 u_i は、 $\sim x_i$ ベクトルの、各要素の平均値である。また、 1 は、全ての要素が1である、 $\sim x_i$ と同次元、つまり400次元のベクトルである。また σ_i は、 $\sim x_i$ ベクトルの、各要素の標準偏差である。ここでの正規化処理は必須では無いが、一般に、本実施形態のように、データ入力部10に入力するデータが画像等の場合は、全体的な信号(ここでは各画素値)の強弱の変動による要因を排除する必要があるため、上記のような正規化を行うと良い。

30

【0042】

ここまでの処理で、N組の、正規化後のパターン画像 x_i と、そのラベル y_i の組が、画像正規化部11等が有するメモリに格納される。

【0043】

次にステップS22では、測地線距離関係行列生成部12は先ず、画像正規化部11が正規化したN個のパターン画像から2つのパターン画像を取り出す。N個のパターン画像から2つのパターン画像を選択する組み合わせ数は $N! / 2! (N-2)! (= M)$ 通りあるので、2つのパターン画像によるセットがMセット得られる。そして、Mセットのそれぞれのセットについて、前述のIsoMapと同様に、2つのパターン画像間の測地線距離 $d_M(i, j)$ (ここで、 $d_M(i, j)$ は、 i 番目のパターン画像 x_i と、 j 番目のパターン画像 x_j との間の測地線距離)を算出する。そして、M個の測地線距離 $d_M(i, j)$ が得られると、測地線距離関係行列DMを求める。

40

【0044】

測地線距離関係行列DMは、 i 行 j 列の成分が、 $d_M(i, j)$ の行列であり、入力されたパターン画像数がN個であるため、N次正方行列となる。また、成分である測地線距離 $d_M(i, j)$ は、 $d_M(i, j) = d_M(j, i)$ なので、測地線距離関係行列DMは対称行列となり、且つ $d_M(i, i) = 0$ なので、対角成分は全て0となる。

50

【 0 0 4 5 】

測地線距離は、前述のように、入力された多数のデータが構成する $Manifold$ の表面に沿った、データ間の最短距離である。ここで、ステップ $S22$ において行われる、本実施形態における測地線距離の近似的な算出方法を、図 3 に示したフローチャートを用いて説明する。図 3 は、近似的な測地線距離算出の処理を示すフローチャートである。なお、図 3 のフローチャートに従った処理は、測地線距離関係行列生成部 12 が行うものとする。

【 0 0 4 6 】

まず、ステップ $S320$ では、 N 個のパターン画像のうち、任意の 2 点間のユークリッド距離 $d_x(i, j)$ を、全ての組み合わせについて算出し、ユークリッド距離関係行列 D_x を求める。ここで、 $d_x(i, j)$ は、 i 番目のパターン画像 x_i と、 j 番目のパターン画像 x_j との間のユークリッド距離である。

10

【 0 0 4 7 】

ユークリッド距離関係行列 D_x は、 i 行 j 列の成分が、 $d_x(i, j)$ の行列であり、測地線距離関係行列 D_M と同様に、 N 次正方の対角成分が 0 である対称行列である。本実施形態では、ステップ $S320$ においてユークリッド距離を用いた。しかし、これに限るものではなく、例えばマンハッタン距離等のミンコフスキー距離や、マハラノビス距離といった統計的な距離等、対称性や、非負性といった、一般的な距離の公理を満たすものであれば、その他の指標を用いても構わない。

【 0 0 4 8 】

20

続いて、ステップ $S321$ において、今度は、入力された N 個のパターン画像のうち、任意の 2 点間のグラフ距離 $d_G(i, j)$ を、全ての組み合わせについて算出し、グラフ距離関係行列 D_G を求める。ここで、 $d_G(i, j)$ は、 i 番目のパターン画像 x_i と、 j 番目のパターン画像 x_j との間のグラフ距離である。

【 0 0 4 9 】

ここで、グラフ距離とは、例えば、 i 番目のパターン画像 x_i と、 j 番目のパターン画像 x_j と 2 点が近傍である場合は、 $d_G(i, j) = d_x(i, j)$ であり、この 2 点が近傍でない場合は、 $d_G(i, j) =$ となる距離である。現実的な演算においては、という数値は利用できないので、の代わりに、任意の 2 つのパターン画像間のユークリッド距離に比べ、十分に大きい定数を利用すればよい。グラフ距離関係行列 D_G は、 i 行 j 列の成分が $d_G(i, j)$ の行列であり、やはり同様に、 N 次正方の対角成分が 0 である対称行列である。

30

【 0 0 5 0 】

2 点が近傍であるか否かは、本実施形態では、それぞれのパターン画像自身から、ステップ $S320$ において求めた距離が近いものから順に、自身を除いた k 個 ($k-1$) のパターン画像 (自身を含めると、 $(k+1)$ 個のデータ) を近傍であると判定する。そして、ある 2 点のパターン画像で、どちらの点においても近傍であると判定されなかった場合、その 2 点は近傍ではないと判定する。このように本実施形態では、自身以外で、距離の近い順に k 個のパターン画像を近傍としているが、例えば、距離が正の値 以内である関係のパターン画像を近傍とするようにしても良い。この場合、は、全てのパターン画像のそれぞれにおいて、少なくとも、自身を除く 1 つのパターン画像が近傍とみなされる程度に大きい値にする必要がある。しかし が大きすぎると、本来近傍とみなすべきでないパターン画像までが、近傍とされてしまうため、あまり大きな値にすることは好ましくない。入力されたパターン画像の数等にも依存するが、通常この は、数個程度のパターン画像が近傍とみなされる程度の大きさにしておくが良い。

40

【 0 0 5 1 】

ステップ $S322$ では、ステップ $S321$ で求めたグラフ距離関係行列 D_G に対して $Floyd-Warshall$ 法を用い、入力された N 個のパターン画像のうち任意の 2 点間の前述の測地線距離 $d_M(i, j)$ を、全ての組み合わせについて算出する。これにより、測地線距離関係行列 D_M を求める。 $Floyd-Warshall$ 法により i 番目の

50

パターン画像 x_i と、 j 番目のパターン画像 x_j の 2 点間の測地線距離 $d_M(i, j)$ は、 $d_M(i, j) = \min \{ d_G(i, j), d_G(i, k) + d_G(k, j) \}$ 、 k i, j のように計算される。

【0052】

以上のステップ S 3 2 0 ~ ステップ S 3 2 2 までの処理により、測地線距離関係行列 D_M を求めることができる。

【0053】

次に、ステップ S 2 3 では、クラス間分離度設定部 1 3 がクラス間分離度設定処理を行う。以下に、クラス間分離度設定部 1 3 が行う処理について詳細に説明する。

【0054】

クラス間分離度設定部 1 3 は、クラス間分離度 (c_p, c_q) を設定する。ここで、 c_p 、及び c_q は、クラスラベルであり、本実施形態では、 m 人の種別が存在するパターン画像セットを用いるので、 $c_p, c_q \in \{1, 2, \dots, m\}$ となり、クラス間分離度 (c_p, c_q) には、 $m \times m$ の組み合わせが存在する。このクラス分離度 (c_p, c_q) は、クラス c_p と、クラス c_q 間の分離表現度合いを表しており、そのクラス間の所望の分離度合いに応じて設定する。具体的には、分離度合いを高めたい場合は、このクラス分離度を 1 より大きく設定し、分離度合いを下げたい場合は 1 より小さく設定する。また、特に分離度合いを変更する必要がない場合は、このクラス分離度は 1 に設定する。ここでの分離度合いとは、前述のように、クラス間の差異の強調度合いであり、例えばクラス c_p と、クラス c_q の差異を強調して表現したい場合には、この 2 つのクラス間のクラス間分離度 (c_p, c_q) を 1 より大きく設定するようにすればよい。

【0055】

この定義のように、クラス間分離度は、可換な 2 クラス間の関係により設定するものであるため、対称性 $(c_p, c_q) = (c_q, c_p)$ を満たす。また、同クラス間のクラス間分離度 (c_p, c_p) は、実際には用いないため任意であり、単純に 1 に設定しておけばよい。

【0056】

次に、本実施形態におけるクラス分離度の設定方法について説明する。本実施形態では、各クラスの分離表現を促進するために、以下の手法により、類似したクラス間のクラス間分離度を 1 より大きく設定し、類似していないクラス間のクラス間分離度は 1 に設定する。

【0057】

まず、全クラスから 2 クラスを選択する全ての組み合わせにおいて、その 2 クラスが類似しているか否かの判定を行う。2 クラスが類似しているか否かの判定方法としては様々な方法が考えられるが、本実施形態では、クラスタ分析で一般的な最短距離法を用いてクラス間の距離を算出し、そのクラス間の距離が閾値より小さい場合に、その 2 クラスが類似していると判定する。そして、類似していると判定された 2 クラス間のクラス間分離度を 1 より大きく設定する。本実施形態では、具体的には、上記閾値を、求めたクラス間の距離で除算した値を、クラス間分離度として設定する。類似していると判定されるクラス間の距離は、上記閾値より小さいので、この値は必ず 1 より大きい値となる。また、類似していると判定されなかった 2 クラス間のクラス間分離度は 1 に設定する。

【0058】

上記閾値は、例えば兄弟等の類似している人物の顔が類似していると判定されるような値を実験的に予め求めておき、それを用いるようにすればよい。また、本実施形態では、最短距離法を用いてクラス間の距離を算出するが、最長距離法や、群平均法、重心法、メジアン法、ウォード法、可変法等、その他のクラス間の距離を算出する手法を用いても構わない。更に、類似したクラス間のクラス間分離度として、上記閾値を、求めたクラス間の距離で除算した値を設定するが、例えば 1 以上の値を設定する等、1 より大きくなるような設定方法であれば、その他の方法であっても構わない。

【0059】

10

20

30

40

50

以上のように、本実施形態では、クラス間の距離に基づいてクラス間分離度を設定するが、これに限るものではなく、予めクラスラベルに与えられた情報に基づいて、クラス間分離度を設定するようにしても構わない。例えば、予めA氏とB氏が双子であるという情報が与えられていれば、A氏とB氏という2つのクラス間のクラス間分離度を1より大きい適当な値に設定するようにしてもよい。

【0060】

また、クラス間分離度の設定は、類似するクラス間のクラス間分離度を高めるという設定方法に限るものではなく、その他の設定方法を用いることも可能である。例えば、所定のクラス c_p を、その他のクラス c_q ($q \neq p$) と明確に分離表現したいような場合は、

(c_p, c_q) のみを1より大きい値に設定し、その他を全て1に設定するというような設定方法が挙げられる。また、クラス c_p と c_q を、明確に区別する必要がないような場合は、 (c_p, c_q) のみを1より小さい値に設定し、その他は全て1に設定するというような設定方法でも構わない。このように、本実施形態で説明するデータ表現方法では、各クラス間の所望の分離度合いに応じて、任意にクラス間分離度を設定することが可能である。

【0061】

次にステップS24では、測地線距離関係行列更新部14は、歪曲測地線距離 $d_d(i, j)$ を算出し、歪曲測地線距離関係行列 D_d を算出する。ここで、 $d_d(i, j)$ は、測地線距離関係行列生成部12において求めた i 番目のパターン画像 x_i と、 j 番目のパターン画像 x_j との間の歪曲測地線距離を示す。また、歪曲測地線距離 $d_d(i, j)$ は、 i 番目のパターン画像 x_i と、 j 番目のパターン画像 x_j との間の測地線距離 $d_M(i, j)$ を、ラベル y_i 及び y_j と、クラス間分離度 (y_i, y_j) に基づき更新したものである。

【0062】

本実施形態では、この歪曲測地線距離 $d_d(i, j)$ を、 $d_d(i, j) = d_M(i, j) \cdot [(y_i, y_j) - \{(y_i, y_j) - 1\} \cdot \alpha_{y_i, y_j}]$ のように求める。ここで、 α_{y_i, y_j} は、クロネッカーの記号で、 $i = j$ の時、 $\alpha_{y_i, y_j} = 1$ 、 $i \neq j$ の時、 $\alpha_{y_i, y_j} = 0$ である。また、係数 α_{y_i, y_j} は、1より小さい正の数である。即ち、歪曲測地線距離 $d_d(i, j)$ は、 i 番目のデータ x_i と、 j 番目のデータ x_j が、同一のラベル、つまり、同一のクラスに属する場合は、 $d_d(i, j) = d_M(i, j)$ となり、測地線距離 $d_M(i, j)$ より小さくなる。

【0063】

一方、同一のラベルでない、つまり異なるクラスに属する場合は、 $d_d(i, j) = d_M(i, j) \cdot (y_i, y_j)$ となる。従って、歪曲測地線距離 $d_d(i, j)$ は、測地線距離 $d_M(i, j)$ に、クラス間分離度 (y_i, y_j) を乗じたものになる。

【0064】

また、2つのデータが同一クラスである場合に乘じる係数 α_{y_i, y_j} は、以下の条件を満たす値を求め、それを設定する。この条件は、同一クラスである任意の2つのパターン画像 x_i 、 x_j と、この2つのパターン画像とは異なるクラスの任意のパターン画像 x_c において、 $d_d(i, j) = d_M(i, j) < d_d(i, c) = d_M(i, c) \cdot (y_i, y_c)$ を満たすことである。つまり、同クラスである2つのパターン画像間の歪曲測地線距離は、異なるクラスの任意のパターン画像との距離より小さくなるようにすれば良い。究極的には、 $\alpha_{y_i, y_j} = 0$ とすることで、必ず上記関係を満たすことができるが、後述のデータ写像関係生成部15における処理において不都合を生じさせる可能性が高くなるため、 $\alpha_{y_i, y_j} > 0$ である値を設定するようにする。

【0065】

本実施形態に係るデータ表現方法では、上記のように、同一のクラスであるパターン画像間の距離が、異なるクラスであるパターン画像との距離よりも小さくなるように、パターン画像間の距離関係を更新する。また、その上記条件において、異なるクラスのパターン画像間の距離は、設定したクラス間分離度に基づいて更新される。これにより、パター

ン画像のクラス内での連続的な変動、例えばパターン画像が“人物の顔”の画像であれば、正面向きから横向きに変化するような、原特徴空間における非線形で連続的な変動を変動の軸方向に凝集させた距離関係を構築することになる。また、各クラス間の関係については、所望のクラスの分離度合いに基づいた距離関係を構築することになる。本実施形態では、上記同一のクラスであるパターン画像間における距離関係の更新を、同一クラスであるパターン画像間距離に、係数 α を乗じて更新することによって行う。

【0066】

しかし、パターン画像間における距離関係の更新は、これに限るものではなく、同一のクラスであるパターン画像間の距離が、異なるクラスであるパターン画像との距離より小さくなるような更新手法であれば、その他の方法を用いても構わない。例えば、同一クラスである全てのパターン画像間の距離を、異なるクラスのパターン画像との距離より小さい、定数 β に更新するようにしても構わない。

【0067】

次にステップS25では、データ写像関係生成部15は、測地線距離関係行列更新部14が算出した歪曲測地線距離関係行列 D_d を用い、全てのパターン画像 x_i それぞれに対応する写像先である、出力ベクトル z_i を求める。

【0068】

以下では、測地線距離関係行列更新部14が算出した i 番目のパターン画像 x_i に対応する出力ベクトル z_i を、 i 番目の出力ベクトル z_i と表記する。ここで出力ベクトルは、 i 番目の出力ベクトル z_i と、 j 番目の出力ベクトル z_j のユークリッド距離 $d_z(i, j)$ が、 i 番目のパターン画像 x_i と、 j 番目のパターン画像 x_j の歪曲測地線距離 $d_d(i, j)$ の近似となるように求められる。本実施形態では、前述の $Isomap$ と同様に、多次元尺度法(MDS)を用いて、このような出力ベクトルを求める(MDSを用いた出力ベクトルの算出法については、非特許文献1を参照)。

【0069】

通常、可視化を目的としてMDSを用いる場合、出力ベクトルを3次元以下とするが、本実施形態に係るデータ表現方法は、特に可視化を目的としているわけではないため、出力ベクトルは、4次元以上であっても構わない。この出力ベクトルの次元を非常に高くすれば、上記距離の近似を比較的厳密に行うことができる。しかし、データの冗長性を削減するという目的では、入力データ(パターン画像)の次元(本実施形態では、400次元)より小さくなるように、可能な限り出力ベクトルの次元を小さくする方が良い。

【0070】

そこで本実施形態では、入力された全パターン画像のペアの、それぞれの距離誤差率のうち、最大の距離誤差率が所定値以下である最小の次元で出力ベクトルを求めるようにする。ここで、 i 番目のデータと、 j 番目のデータの距離誤差率は、 $|d_d(i, j) - d_z(i, j)| / d_d(i, j)$ である。ここで用いる最大の距離誤差率の許容範囲は、入力されたパターン画像のカテゴリ等に依存するが、例えば10%以下というようにすれば良い。また、本実施形態では、最大の距離誤差率が、予め設定された値以下になるような最小の次元を出力ベクトルの次元として用いた。しかしながら、これに限るものではなく、例えば距離誤差率の総和が予め設定された値以下になるような最小の次元を、出力ベクトルの次元としても良い。

【0071】

以上説明したデータ入力部10からデータ写像関係生成部15までの各部における処理により、ラベル付きの各パターン画像について、 $x_i \rightarrow z_i$ という対応関係を得ることができる。これにより、全パターン画像について、冗長性を削減した出力ベクトル z_i の次元の空間において、出力ベクトル z_i が、同クラスであるパターン画像間で近づく。つまり、1つのクラスタとして表現され、尚且つ、各クラスに対応するクラスタを、所望の分離度合いで表現することが可能になる。

【0072】

そしてデータ写像関係生成部15はその後、ラベル付きの各パターン画像について求め

た $\sim x_i$ 、 x_i 、 z_i という対応関係を示す情報をメモリ 16 に格納する。係る情報の表現方法については特に限定するものではないが、例えば、 $\sim x_i$ のファイル名、 x_i のファイル名、 z_i のファイル名をこの順番で関連付けたファイルを作成しても良い。

【0073】

本実施形態では、パターン画像として、人物の顔を切り出したグレースケール画像と、それが何れの人物であるのかのラベルを用いたが、これに限るものではない。例えば、パターン画像の代わりに、予め設定された単語を発話した音声データと、その単語をラベルとするデータを用いても良い。更には、HTMLで記述されたWebページと、そのコンテンツカテゴリをラベルとしたものであっても良い。即ち、ラベル付けされた元のデータ間で、類似度を反映する何らかの距離を定義できるものであれば、どのようなデータであっても、本実施形態は適用可能である。

10

【0074】

[第2の実施形態]

本実施形態では、第1の実施形態で示したデータ表現方法を応用する。本実施形態では、人物の顔を含む原画像からこの顔の領域を切り出すことで得られる縦20画素、横20画素のサイズの抽出画像(パターン画像)であって、グレースケール画像であるパターン画像を入力し、それが何れの人物を含むものであるのかを識別する。もちろん、この入力するパターン画像には、第1の実施形態で説明したようなラベリング処理は行っていない。

【0075】

20

本実施形態は、学習モードと識別モードの2つモードから構成される。

【0076】

学習モードでは、第1の実施形態で取り扱ったパターン画像、即ち、ラベリング処理済みのパターン画像を用いてデータ写像関係を近似的に構築し、その写像先の空間で、各ラベルに対応するクラスのモデルを生成する。

【0077】

識別モードではまず、学習モードで構築したデータ写像関係を用いて、ラベリング処理がなされていないパターン画像を写像する。そして写像先の空間で、学習モードで生成した、各ラベルに対応するクラスのモデルを用い、その画像が、何れの人物の顔画像であるかを識別する。

30

【0078】

<学習モード>

図4は、本実施形態に係る情報処理装置を構成する各部のうち、学習モード時において動作する各部についてのみ、その機能構成を示したブロック図である。なお、図4において、図1に示した構成と共通の部分については、同じ番号を付けており、その説明は省略する。

【0079】

図5は、本実施形態に係る情報処理装置が学習モード時に行う処理のフローチャートである。なお、図5において、図2に示した構成と共通の部分については、同じ番号を付けており、その説明は省略する。

40

【0080】

以下、図4、5を用いて、学習モード時における処理について説明する。

【0081】

本実施形態では、N個のパターン画像(第1の実施形態で取り扱ったパターン画像と同じで、ラベリング処理済みのパターン画像)を入力し、入力したそれぞれのパターン画像について第1の実施形態と同様の処理を行う。これにより、各パターン画像に対する出力ベクトル(本実施形態ではこの出力ベクトルの次元数がhであったとする)を求める。ここまでの処理については、第1の実施形態での処理と同様であるので、説明を省略する。

【0082】

データ写像関係生成部15までの処理により、 $i = 1, 2, \dots, N$ までの任意のi

50

におけるラベル y_i 、正規化後ベクトル x_i 、出力ベクトル z_i の、3つから構成されるデータセット（トレーニングデータセット）が得られる。このトレーニングデータセットを用いて、近似システム構築処理部46は、 x_i z_i という写像を近似するシステムを構築する。本実施形態では、この写像を近似するシステムとして、3層のフィードフォワードニューラルネットワーク（3層NN）を用いる。

【0083】

この3層NNは、入力層、中間層、出力層の3層構造になっており、それぞれの層が複数のニューロンを有する。

【0084】

入力層のニューロン数は、正規化後ベクトルの次元数400次元と同一の400個であり、これらの入力層のニューロンは、3層NNへの入力データである400次元のベクトルの各要素値を、それぞれの状態値とする。

【0085】

中間層のニューロン数は、データ写像関係生成部15での処理において決まり、これらの中間層のニューロンは、入力層の全ニューロンと結合する。

【0086】

ここで結合とは、ある重みを持っており、結合先ニューロン（中間層のニューロンであれば、入力層のニューロン）の状態値に、その重みを乗じたものを入力として受け取るものである。中間層のニューロンは、受け取った全ての入力の総和を取り、そこから予め設定された値（閾値）を差し引いた値に、予め設定された非線形変換を行った値を状態値とする。これらの重みの値や閾値は、初期の状態ではランダムな値を設定しておき、後述する近似システム構築処理部46での処理において値が確定する。また、上記非線形変換は、いわゆるシグモイド関数を用いた変換が一般的であり、本実施形態では、この非線形変換として、双曲線正接関数（ $f(u) = \tanh(u)$ ）を用いる。

【0087】

出力層のニューロン数は、出力ベクトルの次元数と同一の個数であり、これらの出力層のニューロンは、中間層の全ニューロンと結合する。出力ベクトルの次元数は前述のように、データ写像関係生成部15における処理により決まり、ここでは上述のように、出力ベクトルの次元数が h であったとしたので、出力層のニューロン数は h 個となる。

【0088】

出力層のニューロンは、中間層のニューロンと同様に、結合先のニューロン（ここでは中間層のニューロン）の状態値に、結合ごとの重みを乗じた値を入力として受け取る。そして、受け取った全ての入力の総和を取り、そこから閾値を差し引いた値を状態値、つまり出力値とする。このように、出力層のニューロンは、中間層のニューロンとは異なり、非線形変換を行わない。

【0089】

中間層のニューロンからの結合の重みの値や閾値に関しても、中間層のニューロンと同様に、初期の状態ではランダムな値を設定しておき、この近似システム構築処理部46での処理において値が確定する。

【0090】

上記3層構造の演算システムを用い、400次元のベクトルの各要素値を、入力層の400個のニューロンのそれぞれの状態値として設定する（以下では単に、3層NNへ入力すると記す）ことで、出力層の、 h 個のニューロンの状態値が得られる。この h 個のニューロンの状態値を、 h 次元のベクトルの各要素値と考えると、この3層NNを用いた演算により、400次元ベクトルから、 h 次元ベクトルへの写像が行われるとみなせる。

【0091】

ここで、上述したように、結合の重みの値や閾値を様々に変更することで、中間層のニューロンの数にも依存するが、任意の写像を、任意の精度で近似できることが知られている。そこで本実施形態では、この3層NNを用い、結合の重みの値や閾値を調整することで、上記トレーニングデータセットの、任意の i における x_i z_i という写像、つまり

10

20

30

40

50

、正規化後ベクトルから出力ベクトルへの写像を近似する。

【0092】

図5において、ステップS56では、近似システム構築処理部46が写像を近似するシステムを構築する処理を行う。

【0093】

以下では、ステップS56において近似システム構築処理部46が行う処理、即ち、3層NNの結合の重みの値や閾値の調整、及び中間層のニューロン数の決定について、同処理のフローチャートを示す図6を用いて説明する。

【0094】

先ず、ステップS660では、データ写像関係生成部15までの処理により得られた前述のトレーニングデータセット（ラベル、正規化後ベクトル、出力ベクトルの3つを1組とするN組のデータセット）を、2つのセットに分割する。この2つのセットのうち、1つは、結合の重みの値や閾値の調整に用い、以下では、このセットを、調整用セットとする。もう1つのセットは、後述のステップS664における検定に用い、以下では、このセットを検定用セットとする。

【0095】

分割の際には、N組のトレーニングデータから、M組（ $0 < M < N$ ）をランダムに選択し、それらを検定用セットとする。そして、残りの（ $N - M$ ）組が、調整用セットとなる。Mの値は、任意であるが、本実施形態では、Nの30%とする。ここで、この2つのセットの両方において、全てのラベルそれぞれのデータが、少なくとも1つ存在することが好ましい。そこで本実施形態では、もし、どちらかのセットにおいて、あるラベルのデータが1つも存在しない場合は、もう一度ランダムに選択をやり直すようにする。

【0096】

続いてステップS661では、中間層のニューロンの数を設定する。最初は中間層のニューロンの数を予め定めた数に設定する。そして、後述するステップS666における判定により、再度ステップS661に戻ってきた場合に、中間層のニューロンの数を1つ増加させる。最初に設定する中間層のニューロンの数は任意であるが、初めはできるだけ少ない数にしておくことが好ましい。そこで、本実施形態では、初期の中間層のニューロンの数を2とする。

【0097】

次に、ステップS662では、全ての結合の重みの値と閾値をランダムな値に初期化する。ここでは、後のステップS663での処理における初期値依存性を低減するために、全ての値をほぼ0とみなせる程度に小さな値にすることが好ましい。

【0098】

ステップS663では、ステップS662で初期化した結合の重みの値と閾値からスタートし、調整用セットを用いた結合の重みの値と閾値の調整を行い、調整用セット内の、正規化後ベクトルから出力ベクトルという写像を近似した3層NNを構築する。ここでは、調整用セットの中の全ての正規化後ベクトルそれぞれを3層NNに入力した時に、出力層のh個のニューロンのそれぞれの状態値が、それぞれに対応する出力ベクトルの各要素値に近づくように、結合の重みの値と閾値の調整を行う。この調整では、多層フィードフォワードニューラルネットワークの学習手法として一般的な、誤差逆伝播法を用いる。誤差逆伝播法を用いた、結合の重みの値と閾値の調整手法の詳細は、S. Haykin, "Neural Networks A Comprehensive Foundation 2nd Edition", Prentice Hall, pp. 156-255, July 1998を参照されたい。

【0099】

そしてこの調整を、調整用セットの中の全ての正規化後ベクトルそれぞれを3層NNに入力した時に得られる出力層のh個のニューロンの状態値と、それぞれに対応する出力ベクトルの各要素値との誤差が収束した段階で終了する。本実施形態では、この誤差として、出力層のh個のニューロンの状態値と、対応する出力ベクトルの各要素値の2乗誤差総和を用いる。また、誤差が収束したか否かの判定は、誤差逆伝播法を用いた、予め設定さ

10

20

30

40

50

れたステップ数の調整において、ほぼ誤差が減少しなくなった時（例えば、誤差の減少率が1%以下であった時など）に、誤差が収束したと判定する。判定に用いる「予め設定されたステップ数」は任意に設定すればよいが、これが小さすぎると、本来は収束していないのに、誤って収束したと判定してしまう可能性が高くなる。しかし、これが大きすぎると、収束判定がなされない可能性が高くなるので、確実に収束判定がなされるような値を実験的に決めて設定するようにすれば良い。

【0100】

以上説明したステップS663における処理により、中間層のニューロン数が、ステップS661で設定した数の場合の3層NNを用いた、調整用セット内の正規化後ベクトルから出力ベクトルという写像を近似する3層NNの構築が完了する。

10

【0101】

ステップS664では、検定用セットを用いて、ステップS663において構築した3層NNの写像性能を評価する。ここではまず、ステップS663において構築した3層NNに、検定用セット内の全ての正規化後ベクトルをそれぞれ入力し、それぞれの入力に対応する出力層のh個のニューロンの状態値を得る。そして、得られた各h個の状態値と、入力した正規化後ベクトルに対応する、検定用セット内の出力ベクトルの各要素値との2乗誤差総和を求める。この2乗誤差が小さい程、未知の正規化後データに対して所望の写像を行う性能が高いと言える。そこで、この2乗誤差総和を、ステップS663において構築した3層NNの写像性能の評価値として用いる。つまり、この評価値が小さいほど、写像性能が高いと判断できる。

20

【0102】

ステップS665では、ステップS663にて構築した3層NNの中間層のニューロン数、全ての結合の重みの値、閾値、及びステップS664で求めた写像性能の評価値を、それぞれデータとして近似システム構築処理部46が有するメモリに記録する。

【0103】

ステップS666では、これまでにメモリ内に記録した評価値を用いて、3層NNの写像性能が悪化しているか否かを判定する。係る判定の結果、悪化したと判定した場合は、処理をステップS667に進める。一方、悪化していないと判定した場合は、処理をステップS661に戻し、上述のように、中間層のニューロンの数を1つ増加させ、ステップS662以降の処理を行う。

30

【0104】

ここで、ステップS666では、予め設定されたステップ数分前にステップS665で記録された3層NNの評価値に比べ、それ以降の3層NNの評価値が全て悪化、つまり2乗誤差総和が大きい場合に、写像性能が悪化していると判定する。「予め設定されたステップ数」は小さすぎると、悪化したか否かの判定を誤る可能性が高くなるので、現実的な程度に大きい値を設定しておけば良い。初期の段階では、「予め設定されたステップ数」分前の評価値が無いため、悪化の判断はできない。その場合は、必ずステップS661に戻るようにする。

【0105】

ステップS667では、これまでにステップS665で記録された写像性能の評価値のうち最も評価値が小さいものを選択する。そして選択した評価値と共にステップS665で記録された中間層のニューロン数、全ての結合の重みの値、閾値を、後述するクラスモデル生成部47に出力する。

40

【0106】

以上説明したステップS660からステップS667までの処理により、汎化性能の高い、正規化後ベクトルから出力ベクトルへの写像を近似するシステムとして、適切な中間層のニューロン数を有する3層NNが構築される。

【0107】

本実施形態では、上記手法を用いて、正規化後ベクトルから出力ベクトルへの写像を近似する3層NNを構築した。しかし、本実施形態はこれに限るものではなく、その他、一

50

般的な3層NNの構築方法を用いても構わない。更に、本実施形態では、正規化後ベクトルから出力ベクトルへの写像を近似するシステムとして、3層NNを用いたが、これに限るものではない。例えば、3層以上のフィードフォワードニューラルネットワークや、カーネル法を用いたサポートベクター回帰法等の非線形な写像システムにより、上記非線形な写像を近似するようにしても構わない。

【0108】

次にステップS57では、クラスモデル生成部47が、近似システム構築処理部46での処理において構築した3層NNと、トレーニングデータセットを用いて、各ラベルに対応するクラスモデルを生成する。ここで生成するクラスモデルとは、後述の識別モードにおいて用いるものであり、3層NNにより写像される先の空間において、クラスの識別を行うための識別器のパラメータ生成に対応する。例えば、識別モードにおいて、線形識別関数を用いたクラス識別を行う場合、線形識別関数の係数ベクトル w_c と、バイアス値 b_c (c はクラスラベル)を生成すればよい。

10

【0109】

本実施形態では、識別モードでのクラス識別において最近傍法を用いるため、各クラスについて少なくとも1つのプロトタイプデータを生成する。具体的には先ず、構築した3層NNにトレーニングデータセット内の正規化後ベクトルを入力し、その時の3層NNの出力層の h 個のニューロンの状態値を算出する。そして、その状態値を各要素値とする h 次元の写像後ベクトルと、対応するラベルのセット、つまり3層NNにより写像された後のデータとそのラベルを、プロトタイプデータとして、メモリ48に記録する処理を行う。このプロトタイプデータの記録は、トレーニングデータセット内からランダムに選択したデータに対して行ってもよいが、本実施形態では、全てのデータに対して行うようにする。

20

【0110】

以上の処理により、近似システム構築処理部46での処理において構築した3層NNと、クラスモデル生成部47において記録した複数のプロトタイプデータが得られ、これをもって、学習モードでの処理が終了する。つまり学習モードでは、入力されたデータが冗長性が削減された同一クラスのデータであれば近づき、クラス間の分離度合いが所望の分離度合いとなるような空間に写像され、その写像を近似するシステムを構築する。そして、その写像先で、各クラスのモデルを生成する。

30

【0111】

< 識別モード >

図7は、本実施形態に係る情報処理装置を構成する各部のうち、識別モード時において動作する各部についてのみ、その機能構成を示したブロック図である。

【0112】

図8は、本実施形態に係る情報処理装置が識別モード時に行う処理のフローチャートである。

【0113】

以下、図7、8を用いて、識別モード時における処理について説明する。

【0114】

先ずステップS80では、データ入力部70によって、誰の画像であるのかを識別する対象であるパターン画像75 (ラベリング処理がなされていないパターン画像)を入力する。

40

【0115】

次にステップS81では、画像正規化部71がパターン画像75に対して、第1の実施形態において説明した画像正規化部11と同様の画像正規化処理を行い、正規化後の画像の各画素値をラスタスキャン的に並べた400次元のベクトルを生成する。ここで得られたこのベクトルを、正規化後ベクトル x とする。

【0116】

ステップS87では、データ写像処理部77が、学習モードにおいて得られた3層NN

50

を用いて、写像後ベクトル z を算出する。写像後ベクトル z は、学習モードにおいて得られた 3 層 NN に、正規化後ベクトル x を入力し、その時の 3 層 NN の出力層の h 個のニューロンの状態値を各要素値とした h 次元のベクトルである。

【0117】

次にステップ S88 では、識別処理部 78 が、最近傍法を用いたクラス識別を行うため、まず、データ写像処理部 77 が算出した写像後ベクトル z と、学習モードで記録した全プロトタイプデータの写像後ベクトルとのユークリッド距離を算出する。そして、算出した距離が、最も小さかった写像後ベクトルを選択し、それに対応して記録されているラベルを求める。

【0118】

次にステップ S89 では、識別結果出力部 79 は、識別処理部 78 が求めたラベルに対応する人物の種別を示す情報を出力する。ラベルに対応する人物の種別は、予めテーブルとして識別結果出力部 79 が有するメモリ内に保持しておけばよい。係る情報については特に限定するものではないが、例えば、人物名など、人物に関する情報を記したテキスト文章データでも良い。また、本情報処理装置に人物名を発声させるための音声データであっても良い。この場合、情報処理装置には、この音声データに基づいて音声信号を生成する為の構成と、この音声信号に従った音声を出力する音声スピーカとを設ける必要がある。

【0119】

以上説明した処理により、識別対象のパターン画像から、それが誰の顔画像であるのかを識別する処理が可能になる。本実施形態では、入力されるパターン画像（顔の画像）は、予め学習モードで用いたデータ内に含まれる人物である場合を想定しているため、識別結果は、必ず学習モードで用いたデータ内に含まれる人物の何れかとなる。もし、学習モードで用いたデータ内に含まれない人物の画像が入力されるような場合は、識別処理部 78 において求めた最も小さい距離が予め設定された値以上であった場合、それは不明な人物の画像であるという識別結果にすればよい。ここで用いる「予め設定された値」は、データ内に含まれない人物の画像を入力し、それが不明な人物の画像であると判定されるように、実験的に求めてやればよい。

【0120】

以上説明した、学習モード、及び識別モードの処理により、ラベルの付与されていないパターン画像を入力し、それが何れの人物であるかを識別する処理が可能になる。本実施形態では、識別モードにおける識別処理部 78 による識別処理に最近傍法を用いたが、これに限るものではなく、 k -最近傍法や、サポートベクターマシン等、その他の一般的な識別方法を用いても実現できることは明らかである。その場合、学習モードのクラスモデル生成部 47 における、各クラスのモデル化の処理も、用いる識別処理において必要な、いわゆるパラメータの学習や、データの収集といった処理に変更すればよい。

【0121】

このように、本実施形態に係るパターン識別方法ではまず、同クラスのデータ同士ならば、そのデータ間の距離が他のクラスのデータとの距離よりも近づけ、クラス間の関係が所望の分離度合いで表現されるようなデータ写像システムが構築される。そして、この構築されたデータ写像システムにより写像される写像先の空間において、クラス識別を行う識別器を生成しておく。このデータ写像システムを用いて新たに入力されたデータを写像し、写像後のデータを、生成した識別器を用いて識別することで、所望のクラス分離度合いに応じた識別特性を有し、データの本質的に非線形な変動に対して頑健なパターン識別を行うことができる。

【0122】

第 1 の実施形態と同様の手法で、クラス間分離度を設定した場合、類似したクラス間の分離度合いが高まるようになるため、類似したクラスの分離度が高まるような識別特性を得ることができる。これは、類似したクラス間の差異が強調されるような空間への写像システムを構築したことにより実現される特性である。これにより、識別が比較的困難な対

10

20

30

40

50

象であっても、良好な識別結果を得ることが可能になる。

【 0 1 2 3 】

また、第 1 の実施形態における説明で述べた、クラス間分離度の設定方法により、様々な識別特性を得ることが可能になる。例えば、クラス c_p と、その他のクラス c_q ($q \neq p$) のクラス間分離度を全て 1 より大きい値に設定し、その他を全て 1 に設定したような場合、新しく入力したクラス c_q ($q \neq p$) のデータを、クラス c_p であると誤って判定する確率を低くすることができる。

【 0 1 2 4 】

以上のような特性は、例えば、顔画像を用いた強固なセキュリティシステムの構築等において有用である。具体的には、上位の権限を有する人物のクラスを、上記クラス c_p のように扱うことで、一般の人物が誤って上位の権限を有する人物であると判定される可能性を低くすることができ、信頼性の高いセキュリティシステムを構築することができる。また、同様のセキュリティシステムを考えた場合、同程度の権限を有する人物同士であれば、誤ってそれらの人物を取り違えて判定したとしても実害は少ない。そのため、そのような人物のクラスは、明確に区別する必要がないため、それらのクラス間分離度を 1 より小さい値に設定してもよい。このようにすることにより、相対的に、権限が異なるクラス間での分離度合いを高めることができ、権限の異なる人物を取り違えて判定する確率を低くできる。このように、本実施形態に係るパターン識別方法では、クラス間分離度の設定により、所望の識別特性を有したパターン識別方法を実現することができる。

【 0 1 2 5 】

[第 3 の実施形態]

本実施形態では、第 2 の実施形態で示したパターン識別方法の変形として、第 2 の実施形態と同様に、ラベリング処理がなされていないパターン画像を入力し、それが何れの人物であるかを識別する、パターン識別方法の例を示す。

【 0 1 2 6 】

本実施形態も第 2 の実施形態と同様に、学習モードと識別モードの 2 つモードから構成される。本実施形態において各モードで行う処理は、第 2 の実施形態とほぼ同様であるが、データ写像関係生成部 15、近似システム構築処理部 46、クラスモデル生成部 47 での処理に対応する部分、及びデータ写像処理部 77 での処理に対応する部分のみが異なる。そのため、以下ではこの異なる部分のみについて説明し、処理が同様の部分は、説明を割愛する。

【 0 1 2 7 】

< 学習モード >

図 9 は、本実施形態に係る情報処理装置を構成する各部のうち、学習モード時において動作する各部についてのみ、その機能構成を示したブロック図である。なお、図 9 において、図 1 に示した構成と共通の部分については、同じ番号を付けており、その説明は省略する。

【 0 1 2 8 】

図 10 は、本実施形態に係る情報処理装置が学習モード時に行う処理のフローチャートである。なお、図 10 において、図 2 に示した構成と共通の部分については、同じ番号を付けており、その説明は省略する。

【 0 1 2 9 】

以下、図 9、10 を用いて、学習モード時における処理について説明する。

【 0 1 3 0 】

本実施形態でも、N 個のパターン画像（第 1 の実施形態で取り扱ったパターン画像と同じで、ラベリング処理済みのパターン画像）を入力し、入力したそれぞれのパターン画像について第 1 の実施形態と同様の処理を行う。これにより、各パターン画像に対する出力ベクトル（本実施形態ではこの出力ベクトルの次元数が h であったとする）を求める。ここまでの処理については、第 1 の実施形態での処理と同様であるので、説明を省略する。

【 0 1 3 1 】

第2の実施形態では、この求めた歪曲測地線距離関係を近似的に保存する、新たな空間への写像システムを、MDSと3層NNの構築の、2段階のステップにより構築した。

【0132】

本実施形態では、この写像システムとして線形写像システムを用い、線形写像システム構築処理部96において、歪曲測地線距離関係行列と、画像正規化部11において正規化した正規化後ベクトルのセットから、この線形写像システムを構築する。ここでの処理は、図10のステップS106に対応する。

【0133】

先ず、ステップS106において線形写像システム構築処理部96が行う処理について説明する。ここで、構築すべき線形システムを、行列表記としてAとする。Aは、入力
10 正規化後ベクトルの次元、つまり、ここでは400次元であるので、写像後の空間の次元をhとすると、400×hの行列となる。このシステムに、400次元のベクトルxを入力した時に、出力として得られるベクトルzは、 $z = A^T x$ と表せ、これはh次元のベクトルとなる。ここで、 A^T は、Aの転置行列である。この時、任意のi、j番目の、それぞれの正規化後ベクトル x_i 、 x_j を写像した出力ベクトル z_i 、 z_j ($= A^T x_i$ 、 $A^T x_j$)間のユークリッド距離が、歪曲測地線距離 $d_d(i, j)$ を近似するような線形写像システムを構築する。そこで、本実施形態では、以下の式1に示す誤差関数J(A)の最小化問題として、この線形写像Aを求める。

【0134】

【数1】

$$J(A) = \frac{1}{2} \cdot \frac{1}{\sum_{i < j} d_d(i, j)} \sum_{i < j} \frac{(A^T x_i - A^T x_j - d_d(i, j))^2}{d_d(i, j)}$$

【0135】

本実施形態では、これを最小化するAを、最急降下法により求める。ここで、Aのk列目の列ベクトルを a_k (a_k は400次元のベクトルで、 $k = 1, 2, \dots, h$)とする。最急降下法では、まずAの全成分をランダムに初期化する。そして、式2に基づき、Aの成分を逐次更新していく。

【0136】

【数2】

$$a'_k = a_k - \eta \cdot \nabla_{a_k} J(A)$$

【0137】

ここで a'_k は、1回更新した後の、Aのk列目の列ベクトルであり、 η は、1回更新における更新量を決める正の比例定数である。また、 ∇_{a_k} は、ベクトル a_k での偏微分であり、 $\nabla_{a_k} J(A)$ は、式3により求められる。

【0138】

【数3】

$$\nabla_{a_k} J(A) = \frac{1}{\sum_{i < j} d_d(i, j)} \sum_{i < j} \left[\frac{r_{i,j} - d_d(i, j)}{d_d(i, j)} \cdot \frac{v_{i,j}}{r_{i,j}} \cdot (a_k^T v_{i,j}) \right]$$

ただし $r_{i,j} = |A^T v_{i,j}|$, $v_{i,j} = x_i - x_j$

【0139】

本実施形態では、式2に示した更新を、式1の誤差関数J(A)の値が収束するまで逐次行い、収束後の行列Aを得る。ここでの収束は、第2の実施形態における、3層NNの誤差収束判定と同様に、予め設定されたステップ数分の更新において、ほぼ誤差が減少しなくなった時に、誤差が収束したと判定すればよい。 η の値はこの収束と関わっており、
50

これが大きすぎると、誤差がうまく収束しない可能性が高くなるが、小さすぎると、収束までに多くの更新を要する。そこでこの h は、現実的に許される程度の更新回数で収束する程度に小さい値に設定しておくことが好ましい。

【 0 1 4 0 】

上記手法により、誤差関数 $J(A)$ の最小化問題として、線形写像行列 A を求めることができる。上記説明ではこの A の列数を h として一般化した但、この h の値を定める必要がある。一般に、この h が大きい方が近似性能が高い、即ち、誤差関数 $J(A)$ の値を小さくすることができる。しかし、 h が小さい方が冗長性を削減した空間に写像できる。そこで本実施形態では、様々な値の h において上記手法により A を求め、その中で予め設定された条件を満たすもののうち、 h が最も小さい値である A を選択するようにする。具体的には先ず、 h の値の初期値を 1 とし、 A を求めるごとに h の値を 1 ずつ増加させる。そして、各 h の値で求めた A において、次の式 4 に示す条件を満たすかどうかを検証する。

【 0 1 4 1 】

【数 4】

$$\forall i, j, k \quad d_a(i, j) \leq d_a(i, k) \rightarrow r_{i, j} \leq r_{i, k}$$

【 0 1 4 2 】

式 4 は、写像後の空間における任意の 3 点の距離関係が、少なくとも歪曲測地線距離関係の順序を満たすか否かの条件を意味する。 h を 1 つずつ増加させて A を求め、上記式 4 の関係を満たす A が求められた場合、そこで演算を終了し、その時の A を、この線形写像システム構築処理部 9 6 において求めるべき線形写像システムとして自身が有するメモリ内に記録し、保持しておく。本実施形態では、上記式 1 のような誤差関数を定義し、それを最小化する線形写像 A を、最急降下法により求めた。しかし、これに限るものではなく、歪曲測地線距離関係をできるだけ保存、特に順序を保存するような線形写像を求める方法であれば、その他の誤差関数を利用したり、解析的に A を求めたりしても構わない。

【 0 1 4 3 】

次にステップ S 1 0 7 では、クラスモデル生成部 9 7 により、線形写像システム構築処理部 9 6 の処理において構築した線形写像システムと、画像正規化部 1 1 により正規化された全正規化後ベクトルとを用い、各ラベルに対応するクラスモデルを生成する。

【 0 1 4 4 】

ここでは、第 2 の実施形態と同様に、先ず、構築した線形写像システム A に、正規化後ベクトルを入力し、その時の線形写像システムの出力ベクトルを算出する。そして、その h 次元の出力ベクトルを写像後ベクトルとし、対応するラベルと共に、後述の識別モード時に用いるプロトタイプデータとして、メモリ 9 8 に記録する処理を行う。このプロトタイプデータの記録についても第 2 の実施形態と同様に、トレーニングデータセット内からランダムに選択したデータに対して行ってもよいが、本実施形態でも全てのデータに対して行うようにする。

【 0 1 4 5 】

以上の処理により、線形写像システム構築処理部 9 6 での処理において構築した線形写像システム A と、クラスモデル生成部 9 7 において記録した複数のプロトタイプデータが得られ、これをもって、学習モードでの処理が終了する。つまり学習モードでは、入力されたデータが、冗長性が削減された同一クラスのデータであれば近づき、クラス間の分離度合いが所望の分離度合いとなるような空間に写像する線形写像システム A を構築し、その写像先で各クラスのモデルを生成することになる。

【 0 1 4 6 】

< 識別モード >

本実施形態に係る識別モードは、処理部の基本的な構成は、第 2 の実施形態における識別モードの構成と同様であり、図 7 のデータ写像処理部 7 7 に対応する処理部における処理の内容のみが異なる。そのため、ここでは特に処理部の構成を図示せず、データ写像処理部 7 7 に対応する処理部における処理の内容のみ説明し、その他の処理については、説

10

20

30

40

50

明を省略する。

【 0 1 4 7 】

本実施形態も第 2 の実施形態と同様に、データを入力し、画像の正規化を行う。そして、本実施形態のデータ写像処理部では、学習モードにおいて得られた線形写像システム A を用いて写像後ベクトル z を算出する。写像後ベクトル z は、学習モードにおいて得られた線形写像システム A に、正規化後ベクトル x を入力した時に得られる h 次元のベクトルである。つまり、第 2 の実施形態においては、3 層 NN を用いて写像後ベクトルを求めたところを、線形写像システムを用いて求めるという部分が異なるのみである。この後の第 2 の実施形態における識別処理部 7 8、識別結果出力部 7 9 に対応する処理は、第 2 の実施形態と同様に、最近傍のプロトタイプを検索し、その結果を出力する。

10

【 0 1 4 8 】

以上、第 3 の実施形態における処理の、第 2 の実施形態における処理との差異について説明した。識別処理において用いる手法に関しては第 2 の実施形態と同様に様々な手法が適用可能であり、また、想定外の入力パターンに対する対応も、第 2 の実施形態において説明した方法と同様にすればよい。

【 0 1 4 9 】

[第 4 の実施形態]

本実施形態では、第 3 の実施形態で示したパターン識別方法の変形として、第 3 の実施形態における線形写像を、カーネル関数を用いて非線形写像に拡張した場合のパターン識別方法の例を示す。

20

【 0 1 5 0 】

第 3 の実施形態では、線形写像を用いてデータ間の歪曲測地線距離関係をできるだけ保存（特に距離の順序において）し、且つ冗長性を削減できる写像を構築した。線形写像を用いた場合、データの分布が比較的単純な形状（非線形であっても）であれば、上記目的を達成できる。しかし、データの分布が非常に複雑な形状である場合は、目標となる写像を構築できない可能性が高くなる。

【 0 1 5 1 】

そこで、本実施形態では、第 3 の実施形態における線形写像部分、つまり線形写像行列 A を用いて入力データを写像する部分を、カーネル関数を用いた非線形な写像に変更する。ここでカーネル関数とは、ある集合 X を対象とした時に、 $x \in X$ を定義域とする実対称関数で、半正定値性を満たす関数である。このようなカーネル関数の例として、多項式カーネル $K(x, x') = (x \cdot x' + 1)^p$ や、ガウシアンカーネル $K(x, x') = \exp(-|x - x'|^2 / \sigma^2)$ が一般的である。

30

【 0 1 5 2 】

本実施形態では、このようなカーネル関数を用い、入力データ x から出力データ z への非線形写像システムを構築する。このように本実施形態は、第 3 の実施形態とは、写像システムの部分が線形写像であるのかカーネル関数を用いた非線形写像であるのかが異なるのみである。そこで、本実施形態の説明では、カーネル関数を用いた非線形線形写像の構築と、それを用いた非線形写像のみを説明し、その他の部分に関しては説明を省略する。

40

【 0 1 5 3 】

第 3 の実施形態では、入力データ x から出力データ z への写像システムとして、 $z = A^T x$ という線形写像を用いた。これに対し、本実施形態の非線形写像は、入力されたデータ数 N 個の h 次元ベクトル x_n ($n = 1, 2, \dots, N$) と、それぞれに対応する入力ベクトル x_n 、及びカーネル関数 $K(x, x')$ 用い、 $z = \sum_{n=1}^N x_n \cdot K(x, x_n)$ と表される。ここで、 \sum は、 $n = 1$ から $n = N$ までの総和を意味する。

【 0 1 5 4 】

この写像は、どのようなカーネル関数を用いるか（関数自体の選択や、上記カーネル関数例での p や σ 等のパラメータ）にも依存するが、それを固定して考えると、 N 個の h 次元ベクトル x_n のみにより決まる。そこで本実施形態ではカーネル関数として上記ガウシアンカーネルを用い、データ間の歪曲測地線距離関係をできるだけ保存し、且つ冗長性を

50

削減できる写像の構築を、N個のh次元ベクトル \mathbf{u}_n を最適化することにより行う。ガウシアンカーネルのパラメータは任意の定数で構わないが、凡そ入力データ間のユークリッド距離オーダーの定数にしておくことが好ましい。

【0155】

このN個のh次元ベクトル \mathbf{u}_n の最適化は、式5に示す誤差関数 $J(\Gamma)$ の最小化問題の解として得られる。

【0156】

【数5】

$$J(\Gamma) = \frac{1}{2} \cdot \frac{1}{\sum_{i < j} d_d(i, j)} \sum_{i < j} \frac{(\Gamma^T \kappa_i - \Gamma^T \kappa_j - d_d(i, j))^2}{d_d(i, j)} \quad 10$$

【0157】

ここで \mathbf{K} はk列目の列ベクトルがN次元ベクトル \mathbf{K}_k ($k = 1, 2, \dots, h$) のN行h列の行列である。また式中の \mathbf{x}_i は、 $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_k)$ をk番目の要素とするN次元のベクトルであり、 $\mathbf{x}_i = \{ \mathbf{K}(\mathbf{x}_i, \mathbf{x}_1), \mathbf{K}(\mathbf{x}_i, \mathbf{x}_2), \dots, \mathbf{K}(\mathbf{x}_i, \mathbf{x}_N) \}^T$ である。本実施形態においてもこの誤差関数を最小化する Γ を、最急降下法により求める。そこでまず、 Γ の全成分をランダムに初期化する。そして、式6に基づき、 Γ の成分を逐次更新していく。

【0158】

【数6】

$$\mathbf{r}'_k = \mathbf{r}_k - \eta \cdot \nabla_{\mathbf{r}_k} J(\Gamma)$$

【0159】

ここで \mathbf{r}'_k は、1回更新した後の \mathbf{r}_k のk列目の列ベクトルであり、 η は第3の実施形態と同様に、1回更新における更新量を決める正の比例定数である。また、 $\nabla_{\mathbf{r}_k} J(\Gamma)$ は、ベクトル \mathbf{r}_k での偏微分であり、 $\nabla_{\mathbf{r}_k} J(\Gamma)$ は第3の実施形態と同様に、式7により求められる。

【0160】

【数7】

$$\nabla_{\mathbf{r}_k} J(\Gamma) = \frac{1}{\sum_{i < j} d_d(i, j)} \sum_{i < j} \left[\frac{q_{i,j} - d_d(i, j)}{d_d(i, j)} \cdot \frac{\mathbf{u}_{i,j}}{q_{i,j}} \cdot (\mathbf{r}_k^T \mathbf{u}_{i,j}) \right] \quad 30$$

ただし $q_{i,j} = |\Gamma^T \mathbf{u}_{i,j}|$, $\mathbf{u}_{i,j} = \kappa_i - \kappa_j$

【0161】

本実施形態でも、第3の実施形態と同様に、式6に示した更新を、式5の誤差関数 $J(\Gamma)$ の値が収束するまで逐次行い、収束後の行列 Γ を得る。この収束後の行列 Γ のn行目の行ベクトルが、求めるh次元ベクトル \mathbf{u}_n となる。収束の判定や、 η の設定も、第3の実施形態と同様にすればよいので、説明を省略する。

【0162】

の列数hは、第3の実施形態と同様の手法で求めてもよいが、第2の実施形態において用いた3層NNの中間層のニューロン数決定手法と同様の交差検定法を用いる方が好適である。この場合、まずデータを調整用セットと、検定用セットに分割し、調整用セットを用いて、第3の実施形態と同様に、様々なhの値において上記手法により \mathbf{u}_n を求める。そして、各hの値で得られた \mathbf{u}_n で決まる写像 $z = \sum_k \mathbf{r}_k \cdot \mathbf{K}(\mathbf{x}, \mathbf{x}_k)$ を、検定用セットに対して適用し、それらが予め設定された条件を満たすもののうち、hが最も小さい値である h を選択するようにすればよい。ここで用いる条件としては、第3の実施形態にお

10

20

30

40

50

いて示した式 4 の条件や、第 1 の実施形態において用いた入力された全データのペアの、それぞれの距離誤差率の内、最大の距離誤差率が、予め設定された値以下であるといった条件を用いればよい。

【 0 1 6 3 】

本実施形態も、上記式 5 のような誤差関数を定義し、それを最小化する行列 Γ を、最急降下法により求めた。しかし、これに限るものではなく、歪曲測地線距離関係を、特に順序を保存するように、写像する非線形変換を決定する行列 Γ を求める方法であれば、その他の誤差関数を利用したり、解析的に行列 Γ を求めたりしても構わない。特に、式 5 に示した誤差関数に関しては、よりスパースな解を得るため、 Γ に関する L 1 ノルムを正則化項として付加し、式 8 のようにしてもよい。

【 0 1 6 4 】

【数 8】

$$J(\Gamma) = \frac{1}{2} \cdot \frac{1}{\sum_{i < j} d_d(i, j)} \sum_{i < j} \frac{(\Gamma^T \kappa_i - \Gamma^T \kappa_j - d_d(i, j))^2}{d_d(i, j)} + \lambda \cdot \sum_k |\gamma_k|_{L1}$$

【 0 1 6 5 】

ここで、 $|\gamma_k|_{L1}$ は γ_k の L 1 ノルムであり、第 2 項の \sum_k は、 $k = 1$ から $k = h$ までの総和を意味する。また、 λ は正則化の効果を定める正のパラメータであり、正則化の効果を定める定数である。この λ の値を大きくすることで正則化の効果が強まるが、実際に用いる値としては、求めるスパースネスと、最終的な写像性能に応じて実験的に決めてやればよい。誤差関数を式 8 とした場合には、式 7 に示した、 $J(\Gamma)$ は、式 9 のようになる。

【 0 1 6 6 】

【数 9】

$$\nabla_{\gamma_k} J(\Gamma) = \frac{1}{\sum_{i < j} d_d(i, j)} \sum_{i < j} \left[\frac{q_{i,j} - d_d(i, j)}{d_d(i, j)} \cdot \frac{u_{i,j}}{q_{i,j}} \cdot (\gamma_k^T u_{i,j}) \right] + \Lambda_k^T \gamma_k$$

【 0 1 6 7 】

ここで Λ_k は、 γ_k の n 番目の要素を $\gamma_{k,n}$ とした場合、この行列 Λ_k の、 n 行 n 列目の対角成分が $1/|\gamma_{k,n}|$ で、それ以外の成分は全て 0 となる、 N 次の対角行列である。よって、よりスパースな解を得るためには、この式 9 を用い、式 6 に示した更新を、式 8 の誤差関数 $J(\Gamma)$ の値が収束するまで逐次行い、収束後の行列 Γ を得るようにすればよい。

【 0 1 6 8 】

第 3 の実施形態におけるクラスモデル生成部 97 や、データ写像処理部 77 に対応する本実施形態での処理では、第 3 の実施形態における線形写像 $z = A^T x$ を、学習モード時のカーネル関数 K を用いて、 $z = \sum_k \gamma_k \cdot K(x, x_k)$ と置き換えるだけでよい。

【 0 1 6 9 】

上記手法により、第 3 の実施形態における線形写像を、カーネル関数を用いた非線形な写像に置き換えることができ、より複雑なパターンの分布に対応可能となる。本実施形態では、カーネル関数を固定として説明したが、様々なカーネル関数（関数自体の選択や、上記カーネル関数例での p や γ 等のパラメータも含めて）を用いて上記写像を構築し、その中で、交差検定における誤差関数の値が最小のものを選ぶようにしてもよい。

【 0 1 7 0 】

上記説明した、第 2 から第 4 の実施形態のパターン識別方法の例では、人物の顔を切り出したグレースケール画像を入力データとして用いた。しかし、これに限るものではなく、その他のカテゴリの画像データや、音声データに対しても適用可能であることは明らかである。また、例えば Web コンテンツ等の一般的なデータであっても、各データの距離

10

20

30

40

50

、及びいくつかのパラメータによって定まるそのデータに対する多次元空間への写像が定義できれば、第2から第4の実施形態における入力データとして用いることができる。この場合、式1、5、又は式8に示したような誤差関数を用い、写像を決めるパラメータを、この誤差関数が最小になるように定めてやればよい。

【0171】

〔第5の実施形態〕

図1、4、7、9に示した各部（メモリを除く）は、上記実施形態では全てハードウェアでもって構成されているものとして説明した。しかしこれら各部をソフトウェアでもって実装しても良い。即ち、各部に対応する処理をコンピュータに実行させるためのコンピュータプログラムとして実装しても良い。

10

【0172】

図11は、図1、4、7、9に示した各部をコンピュータプログラムでもって実装した場合に、このコンピュータプログラムを実行するコンピュータのハードウェア構成例を示すブロック図である。

【0173】

CPU1101は、RAM1102やROM1103に格納されているプログラムやデータを用いて本コンピュータ全体の制御を行うと共に、図1、4、7、9に示した各部が行う各処理を実行する。

【0174】

RAM1102は、外部記憶装置1106からロードされたプログラムやデータ、I/F1107を介して外部から受信したプログラムやデータを一時的に記憶するためのエリアを有する。また、RAM1102は、CPU1101が各種の処理を実行する際に用いるワークエリアを有する。即ち、RAM1102は、各種のエリアを適宜提供することができる。また、RAM1102は、メモリ16、48、98としても機能する。

20

【0175】

ROM1103には、本コンピュータの設定データやブートプログラムなどが格納されている。

【0176】

操作部1104は、キーボードやマウスなどにより構成されており、本コンピュータのユーザが操作することで、各種の指示やデータを入力することができる。例えば、上記各説明における「設定する処理」では、ユーザが操作部1104を用いて入力したものを受け付ける、というようにしても良い。

30

【0177】

表示部1105は、CRTや液晶画面などにより構成されており、CPU1101による処理結果を画像や文字などでもって表示することができる。例えば、上記識別モードにおける処理結果としての識別結果を画像や文字などでもって表示することができる。また、パターン画像の一覧をこの表示部1105の表示画面上に表示させ、本コンピュータにおいて処理すべき対象としてのパターン画像をこの一覧表示されたパターン画像群から操作部1104でもって選択させるようにしても良い。

【0178】

外部記憶装置1106は、ハードディスクドライブ装置に代表される大容量情報記憶装置である。ここには、OS（オペレーティングシステム）や、図1、4、7、9に示した各部の機能をCPU1101に実行させるためのプログラムやデータ、また、予め設定されているものとして説明した関数プログラムやデータなども保存されている。また、パターン画像など、処理対象のデータについてもこの外部記憶装置1106に保存されている。なお、外部記憶装置1106は、メモリ16、48、98の機能をも兼ねても良い。

40

【0179】

外部記憶装置1106に保存されているプログラムやデータはCPU1101による制御に従って適宜RAM1102にロードされる。そしてCPU1101はこのロードされたプログラムやデータを用いて処理を実行する。これにより本コンピュータは、図1、4

50

、 7 , 9 に示した各部による処理を実行することができる。

【 0 1 8 0 】

I / F 1 1 0 7 は、本コンピュータを外部の機器と接続するためのものである。例えば、この I / F 1 1 0 7 を LAN やインターネット等のネットワークに接続することで、例えば、ネットワーク上の機器からパターン画像などをこの I / F 1 1 0 7 を介してダウンロードすることができる。また、本コンピュータによる処理結果をこの I / F 1 1 0 7 を介してネットワーク上の機器に対して送信することもできる。

【 0 1 8 1 】

1 1 0 8 は上述の各部を繋ぐバスである。なお、図 1 1 に示した構成は一例であり、以上説明したコンピュータプログラムを実行可能な構成であれば、コンピュータは如何なる構成を有していても良い。

10

【 0 1 8 2 】

〔 その他の実施形態 〕

また、本発明の目的は、以下のようにすることによって達成されることはいうまでもない。即ち、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録したコンピュータ読み取り可能な記録媒体（または記憶媒体）を、システムあるいは装置に供給する。そして、そのシステムあるいは装置のコンピュータ（または CPU や MPU ）が記録媒体に格納されたプログラムコードを読み出し実行する。この場合、記録媒体から読み出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記録した記録媒体は本発明を構成することになる。

20

【 0 1 8 3 】

また、コンピュータが読み出したプログラムコードを実行することにより、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているオペレーティングシステム（ OS ）などが実際の処理の一部または全部を行う。その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【 0 1 8 4 】

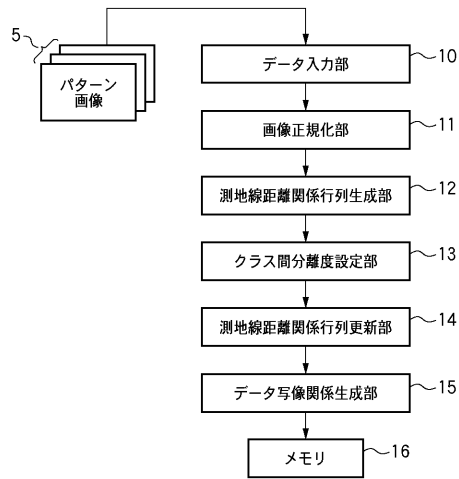
さらに、記録媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張カードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれたとする。その後、そのプログラムコードの指示に基づき、その機能拡張カードや機能拡張ユニットに備わる CPU などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

30

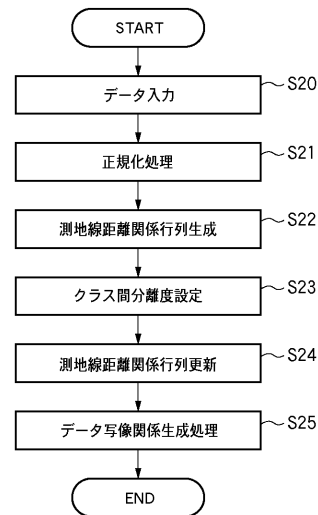
【 0 1 8 5 】

本発明を上記記録媒体に適用する場合、その記録媒体には、先に説明したフローチャートに対応するプログラムコードが格納されることになる。

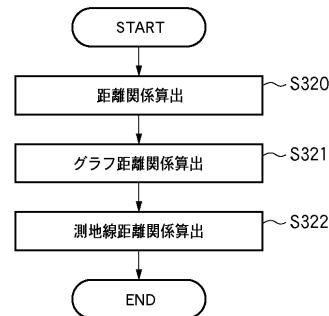
【図 1】



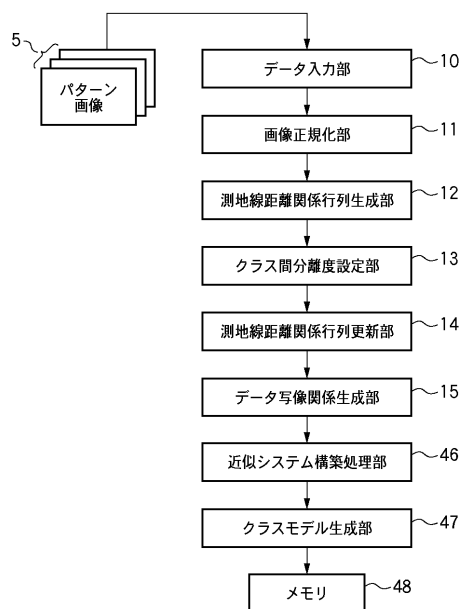
【図 2】



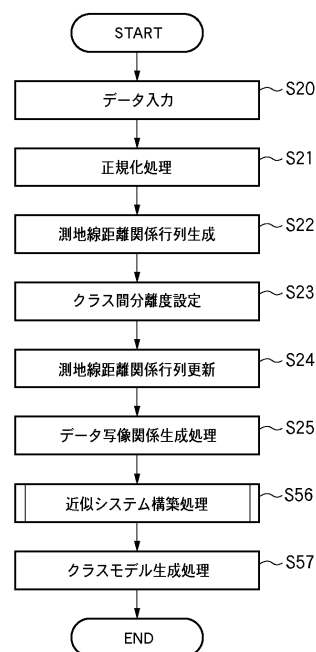
【図 3】



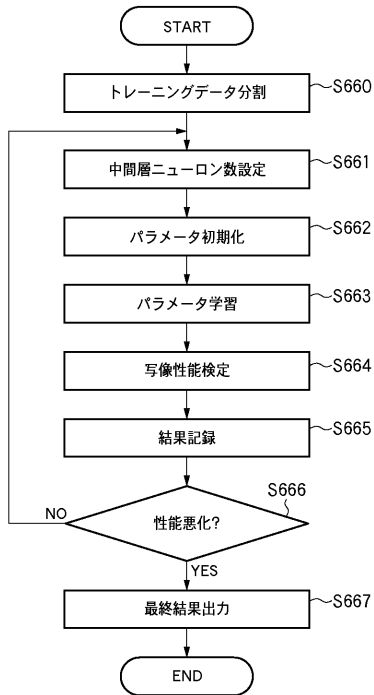
【図 4】



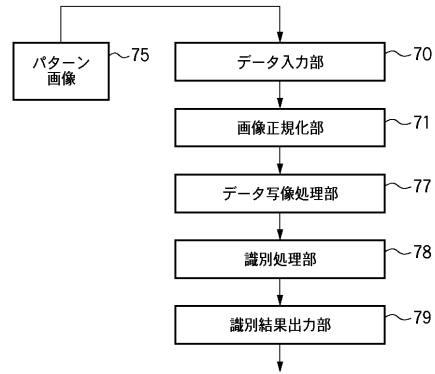
【図 5】



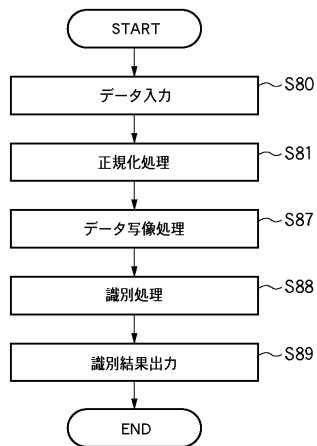
【図 6】



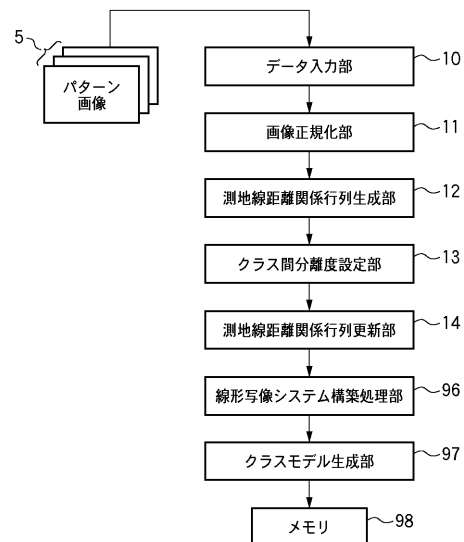
【図 7】



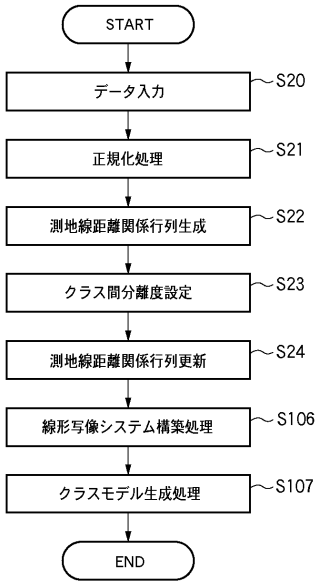
【図 8】



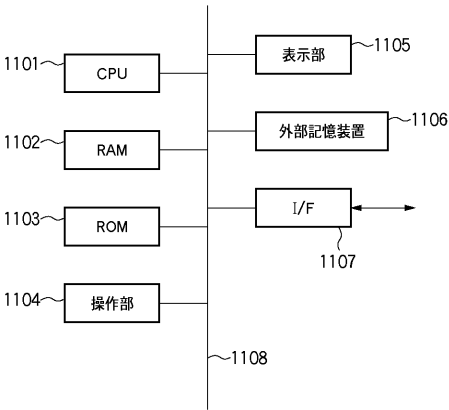
【図 9】



【図 10】



【図 11】



フロントページの続き

- (72)発明者 御手洗 裕輔
東京都大田区下丸子3丁目30番2号 キヤノン株式会社内
- (72)発明者 真継 優和
東京都大田区下丸子3丁目30番2号 キヤノン株式会社内
- (72)発明者 森 克彦
東京都大田区下丸子3丁目30番2号 キヤノン株式会社内

審査官 新井 寛

- (56)参考文献 特開2004-086262(JP,A)
特表2005-535017(JP,A)
Bisser Raytchev, et al., Multi-View Face Recognition By Nonlinear Dimensionality Reduction And Generalized Linear Models, Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition (FGR'06), 2006年 4月 6日, p.625-630

- (58)調査した分野(Int.Cl., DB名)
G06N 3/00
G06T 7/00
G06F 17/30
IEEE Xplore