

(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2014 年 8 月 28 日(28.08.2014)



(10) 国際公開番号

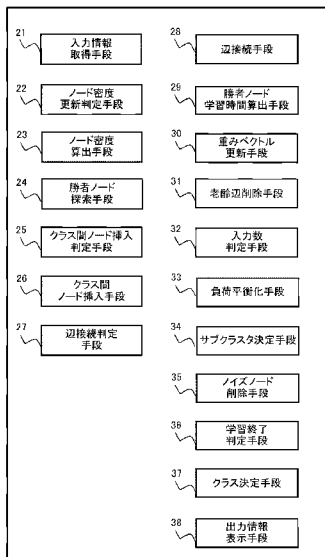
WO 2014/129116 A1

- (51) 国際特許分類:
G06N 3/08 (2006.01) G06N 3/04 (2006.01)
- (21) 国際出願番号: PCT/JP2014/000390
- (22) 国際出願日: 2014 年 1 月 27 日(27.01.2014)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (30) 優先権データ:
特願 2013-033176 2013 年 2 月 22 日(22.02.2013) JP
- (71) 出願人: 国立大学法人東京工業大学(TOKYO INSTITUTE OF TECHNOLOGY) [JP/JP]; 〒1528550 東京都目黒区大岡山 2-1-2 Tokyo (JP).
- (72) 発明者: 長谷川 修(HASEGAWA, Osamu); 〒1528550 東京都目黒区大岡山 2-1-2 国立大学法人東京工業大学内 Tokyo (JP). 張 洪偉(ZHANG, Hongwei); 〒1528550 東京都目黒区大岡山 2-1-2 国立大学法人東京工業大学内 Tokyo (JP).
- (74) 代理人: 家入 健(IEIRI, Takeshi); 〒2210835 神奈川県横浜市神奈川区鶴屋町三丁目 3 番 8 アサヒビルディング 10 階 響国際特許事務所 Kanagawa (JP).
- (81) 指定国 (表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) 指定国 (表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, RU, TJ, TM), ヨーロッパ (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR,

[続葉有]

(54) Title: INFORMATION PROCESSING DEVICE, INFORMATION PROCESSING METHOD, AND NON-TRANSITORY COMPUTER-READABLE MEDIUM

(54) 発明の名称: 情報処理装置、情報処理方法及び非一時的なコンピュータ可読媒体



- 21 Input information acquisition means
22 Node density update assessment means
23 Node density computation means
24 Winner node searching means
25 Inter-class node insertion assessment means
26 Inter-class node insertion means
27 Edge connected assessment means
28 Edge connect means
29 Winner node learning time computation means
30 Weighing vector update means
31 Old edge deletion means
32 Input number assessment means
33 Load equilibration means
34 Sub-cluster determination means
35 Noise node deletion means
36 Learning termination assessment means
37 Class determination means
38 Output information display means

(57) Abstract: An information processing device (1) comprises: an inter-class node insertion means (26) for inserting an input vector as an inter-class insertion node into a network; a winner node learning time computation unit (29) which increments a node learning time for a first winner node by a prescribed value when edges between the first winner node and a second winner node are connected; and a load equilibration means (33) for, for each prescribed period according to an aggregate of the input vectors, detecting a node whereat the value of the learning time of the node is relatively large and skewed, inserting a new node near the detected node and an adjacent node thereto, reducing the learning time of the detected node and the learning time of the adjacent node thereto, deleting the edge between the detected node and the adjacent node thereto, and connecting the edges between the newly inserted node and the detected node, and between the newly inserted node and the adjacent node to the detected node.

(57) 要約: 情報処理装置 (1) は、入力ベクトルをクラス間挿入ノードとしてネットワークに挿入するクラス間挿入手段 (26) と、第 1 勝者ノード及び第 2 勝者ノード間に辺を接続したときに、第 1 勝者ノードについてノードの学習時間を所定の値増加する勝者ノード学習時間算出手段 (29) と、入力ベクトルの総数に応じた所定の周期ごとに、ノードの学習時間の値が相対的に大きくかつ偏っているノードを検出し、その検出したノード及びその隣接ノードの近くに新たなノードを挿入し、検出したノードの学習時間及びその隣接ノードの学習時間をそれぞれ減少し、検出したノード及びその隣接ノード間の辺を削除し、新たに挿入したノード及び検出したノード間と、新たに挿入したノード及び検出したノードのその隣接ノード間と、に辺をそれぞれ接続する負荷平衡化手段 (33) と、を備える。

WO 2014/129116 A1



GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT,
NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI
(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML,
MR, NE, SN, TD, TG).

添付公開書類:

— 国際調査報告 (条約第 21 条(3))

明 細 書

発明の名称：

情報処理装置、情報処理方法及び非一時的なコンピュータ可読媒体

技術分野

[0001] 本発明は任意のクラスに属する入力ベクトルを順次入力して、当該入力ベクトルの入力分布構造を学習する情報処理装置、情報処理方法、及び非一時的なコンピュータ可読媒体に関する。

背景技術

[0002] 学習中に必要に応じてニューロンを増殖させる学習手法として、自己組織化ニューラルネットワーク（S0INN：Self-Organizing Incremental Neural Network）と呼ばれる手法が提案されている（非特許文献1を参照。）。非特許文献1に開示されたS0INNでは、ノード数を自律的に管理することにより非定常的な入力を学習することができ、複雑な分布形状を有するクラスに対しても適切なクラス数及び位相構造を抽出できるなど多くの利点を有する。S0INNの応用例として、例えばパターン認識においては、ひらがな文字のクラスを学習させた後に、カタカナ文字のクラスなどを追加的に学習させることができる。

[0003] さらに、S0INNに関する技術として、特許文献1に開示されたE-S0INN（Enhanced S0INN）と呼ばれる手法が提案されている。E-S0INNでは、学習を随時追加するオンライン追加学習が可能であり、バッチ学習ではなく学習効率が良いという利点を有している。このため、E-S0INNでは、学習環境が新しい環境に変化した場合においても追加学習が可能である。また、E-S0INNでは、入力データに対するノイズ耐性が高いという利点をも有している。

先行技術文献

特許文献

[0004] 特許文献1：特許第5 1 3 0 5 2 3号公報

非特許文献

- [0005] 非特許文献1: F. Shen and O. Hasegawa, "An incremental network for on-line unsupervised classification and topology learning," Neural Networks, Vol.19, No.1, pp.90-106, 2006.

発明の概要

発明が解決しようとする課題

- [0006] しかしながら、非特許文献1に開示される技術(S0INN)や特許文献1に開示される技術(E-S0INN)では、入力データの入力順序によって学習結果が異なってしまうという安定性の問題があった。また、例えば1000次元以上の高次元データを入力データとする場合に学習精度が低下するという問題もあった。以下、これらの問題をより詳細に説明する。
- [0007] 安定性に関する1つ目の問題点として、入力データの入力順序に応じて学習結果が大きく影響を受けてしまうことがある。即ち、学習環境は変化させずに、入力順序のみを変更して入力データを入力して学習を繰り返した場合に、学習結果(ネットワークにおけるノードの個数及びノードの位置)が異なってしまう。
- [0008] 図1は、実験用の人工データセットを示す。図2(A)、(B)は、図1のガウスデータ分布セットに対するE-S0INNによる学習結果をそれぞれ示し、入力データの分布に関して2つのガウス分布が重複してしまっている。図2(A)は、50万回の学習を経た後の、E-S0INNによる定常的な学習結果を示す。図2(B)は、100万回の学習を経た後の、E-S0INNによる定常的な学習結果を示す。図2(A)、(B)から明らかなように、学習の結果生成されたこれらの分布は、図1に示した入力分布と一致していない。即ち、図1に示した入力データには互いに同一の2つのガウス分布を含み、図2(A)、(B)は、図1の入力データに対する学習結果をそれぞれ示している。しかし、図2(A)、(B)のそれぞれの学習結果では、各分布の中心におけるノードの個数が同一とならないために、図1の2つのガウス分布のそれぞれの中心と、図2(A)の2つのガウス分布のそれぞれの中心とが一致せず、また、図1の2つのガウス分布のそれぞれの中心と、図2(B)の2つの

ガウス分布のそれぞれの中心とも一致しない。本来であれば、図 1 の 2 つのガウス分布のそれぞれの中心は、図 2 (A)、(B) の 2 つのガウス分布のそれぞれの中心と一致すべきである。

[0009] さらに、E-S0INNの安定性に関する他の問題点を説明する。E-S0INNは、ノードの累積ポイントの平均値を使用して、ノードの密度を表現する（特許文献 1 の段落 0067、数 12、数 13などを参照。）。E-S0INNは、ノード i のノード密度のポイント値である " p_i " を、以下の式 (1) を用いて算出する。なお、ノード密度のポイント値 p_i は、ノード i が勝者ノードである場合に式 (1) の算出値が与えられ、そうでない場合には値 0 が与えられる。

[数1]

$$p_i = \begin{cases} \frac{1}{(1+\bar{d}_i)^2}, & \text{if } i \text{ is a winner;} \\ 0, & \text{if } i \text{ is not a winner;} \end{cases} \quad \dots (1)$$

[0010] 式 (1) における $\bar{d}_i$ ($\bar{d}_i$ は、 d_i の上に記号「 $\bar{\cdot}$ (バー)」が付いていることを示す。) は、ノード i からその隣接ノード j までの平均距離を示し、以下の式 (2) を用いて算出する。なお、隣接ノード j とは、ノード i と辺によって直接的に接続されるノードを示す。式 (2) において、 m はノード i の隣接ノード j の個数を示し、 W_i はノード i の重みベクトルを示し、 W_j は隣接ノード j の重みベクトルを示す。

[数2]

$$\bar{d}_i = \frac{1}{m} \sum_{j=1}^m \|W_i - W_j\| \quad \dots (2)$$

[0011] E-S0INNによるこの手法を用いて、図 2 (A)、(B) に示したノード分布に対してのノードの密度を算出すると、算出されるノードの密度は明らかに

不正確なものになる。これは、このノードの分布に対するノードの密度を算出しようとした場合に、当該算出に先立って生成されたノードの分布が入力分布にそもそも一致しておらず、さらに、この入力分布との不一致に起因して、E-S0INNは2つのガウス分布を一つのクラスに誤って統合してしまうことがあるためである。

[0012] さらに、E-S0INNの安定性に関する他の問題点を説明する。E-S0INNは、クラス間に存在する大きく重複した領域を削除しようとする。しかし、E-S0INNのアルゴリズムは安定的ではないために、サブクラスタの統合及び分離の繰り返しを引き起こしてしまう。

[0013] また、従来技術に関する他の問題点を説明する。従来技術では、高次元データを入力データとした場合に学習精度が低下することがある。例えばE-S0INNでは、ユークリッド距離を使用して最近傍ノードを探索する。しかし、ユークリッド距離の概念を使用して距離を測定すると、与えられた高次元データのデータセットにおける任意の2つの位置が収束してしまうために、入力データの次元数が増加するに従って学習結果の精度が低下してしまう。即ち、高次元空間においてユークリッド距離を使用した場合には、最も近い位置と最も遠い位置との区別が、とりわけ意味をなくしてしまう。このため、E-S0INNは、例えばDNAクラスタリングや文書クラスタリングなどの高次元データを入力ベクトルとする、いくつかのクラスタリングの学習タスクには適していない。

[0014] 本発明は、上述した問題を解決するためになされたものであり、学習結果について優れた安定性を有する情報処理装置、情報処理方法及び非一時的なコンピュータ可読媒体を提供することを第1の目的とする。さらに、高次元データに対しても良好な精度を有する情報処理装置、情報処理方法及び非一時的なコンピュータ可読媒体を提供することを第2の目的とする。

課題を解決するための手段

[0015] 本発明の一実施の形態に係る情報処理装置は、多次元ベクトルで記述されるノード及び当該ノード間を接続する辺が配置されるネットワーク構造を有

し、任意のクラスに属する入力ベクトルを順次入力して、当該入力ベクトルの入力分布構造を学習する情報処理装置において、入力される前記入力ベクトルに最も近い距離に位置するノード及び2番目に近い距離に位置するノードを第1勝者ノード及び第2勝者ノードとしてそれぞれ探索する勝者ノード探索手段と、前記入力ベクトルと前記第1勝者ノード間の距離及び前記入力ベクトルと前記第2勝者ノード間の距離に基づいて、クラス間ノード挿入を実行するか否かを判定するクラス間ノード挿入判定手段と、前記クラス間ノード挿入判定手段による判定の結果、前記クラス間ノード挿入を実行する場合に、前記入力ベクトルの成分と同一の成分を重みベクトルとして有するクラス間挿入ノードを生成し、当該生成したクラス間挿入ノードを前記ネットワークに挿入するクラス間ノード挿入手段と、前記クラス間ノード挿入判定手段による判定の結果、前記クラス間ノード挿入を実行しない場合に、前記第1勝者ノード及び前記第2勝者ノード間に辺を接続したときに、前記第1勝者ノードについてノードの学習時間を所定の値増加する勝者ノード学習時間算出手段と、前記クラス間ノード挿入判定手段による判定の結果、前記クラス間ノード挿入を実行しない場合に、注目するノードについて当該注目するノードと辺によって直接的に接続されるノードを隣接ノードとして、前記第1勝者ノードの重みベクトル及び前記第1勝者ノードの隣接ノードの重みベクトルをそれぞれ前記入力ベクトルに更に近づけるように更新する重みベクトル更新手段と、前記入力ベクトルの総数に応じた所定の周期ごとに、前記ノードの学習時間の値が相対的に大きくかつ偏っているノードを検出し、当該検出したノード及びその隣接ノードの重みベクトルに基づく重みベクトルを有する新たなノードを生成して前記ネットワークに挿入し、前記検出したノードの学習時間及び前記検出したノードの隣接ノードの学習時間をそれぞれ減少し、前記検出したノード及びその隣接ノード間の辺を削除し、前記新たに挿入したノード及び前記検出したノード間と、前記新たに挿入したノード及び前記検出したノードのその隣接ノード間と、に辺をそれぞれ接続する負荷平衡化手段と、を備える、ことを特徴とするものである。これによっ

て、入力データの入力順序に影響を受けずに、入力パターンの接続構造を正確に記述することができるため、従来技術に比べてより優れた安定性能を有する。

[0016] また、前記クラス間ノード挿入判定手段による判定の結果、前記クラス間ノード挿入を実行しない場合に、付与されるノードのサブクラスタラベルに基づいて、前記第1勝者ノード及び前記第2勝者ノードが属するサブクラスタをそれぞれ判定し、当該判定の結果、前記第1勝者ノード及び前記第2勝者ノードが互いに異なるサブクラスタに属する場合には、前記第1勝者ノード及び前記第2勝者ノードのノード密度に関して、前記第1勝者ノードを含むサブクラスタの平均ノード密度を基準とする前記第1勝者ノードについてのノード密度の条件、又は、前記第2勝者ノードを含むサブクラスタの平均ノード密度を基準とする前記第2勝者ノードについてのノード密度の条件のうちの少なくとも1つを満たすか否かを更に判定する辺接続判定手段と、（1）前記辺接続判定手段による判定の結果、前記第1勝者ノード及び前記第2勝者ノードの少なくとも1つがいずれのサブクラスタに属していない、又は、同一のサブクラスタに属する場合には、前記第1勝者ノード及び前記第2勝者ノード間に辺を接続し、若しくは、（2）前記辺接続判定手段による判定の結果、前記第1勝者ノード及び前記第2勝者ノードについてのノード密度の条件のうちの少なくとも1つを満たす場合には、前記第1勝者ノード及び前記第2勝者ノード間に辺を接続し、前記第1勝者ノード及び前記第2勝者ノードについてのノード密度の条件のうちの少なくとも1つを満たさない場合には、前記第1勝者ノード及び前記第2勝者ノード間に辺を接続しない辺接続手段と、注目するノード及び当該注目するノードの隣接ノードからの平均距離に基づいて、当該注目するノードのノード密度を算出するノード密度算出手段と、局所的に最大のノード密度を有するノードを頂点として、当該頂点に対してそれぞれ異なるサブクラスタラベルを付与し、当該サブクラスタラベルが付与されなかった全てのノードに対してノード密度が最大の隣接ノードと同一のサブクラスタラベルを付与し、前記頂点のうちでノード

密度が所定の閾値よりも大きな頂点を基準としてボロノイ領域を生成し、当該生成したボロノイ領域において、前記基準とした頂点を含むサブクラスタと前記基準とした頂点とは異なる他の頂点を含むサブクラスタとが重複領域を有し、かつ、当該重複領域に位置するノードの平均ノード密度の条件を満たす場合に、前記基準とした頂点を含むサブクラスタのサブクラスタラベルを、前記他の頂点を含むサブクラスタのサブクラスタラベルとして付与するサブクラスタ決定手段と、を更に備える、ようにしてもよい。これにより、異なるサブクラスタに混合クラスを分離するときの乱れを回避することができるため、学習結果をより安定させることができる。

[0017] また、ユークリッド距離に基づいてノード間の距離を算出する重み付きの第1の距離算出式と、前記ユークリッド距離とは異なる他の距離尺度に基づいてノード間の距離を算出する重み付きの第2の距離算出式と、を含むノード間距離算出式を更に備え、前記ネットワークにおける任意の2つのノード間の前記ユークリッド距離に基づく最小距離値及び最大距離値を用いて前記第1の距離算出式を正規化し、前記ネットワークにおける任意の2つのノード間の前記他の距離尺度に基づく最小距離値及び最大距離値を用いて前記第2の距離算出式を正規化し、前記入力ベクトルの次元数が増加するに従って、前記第1の距離算出式の重みを減少させ、前記ノード間距離算出式を用いて、前記入力ベクトル及び前記ノード間の距離、又は、前記ネットワークにおける任意の2つのノード間の距離を算出するようにしてもよい。これにより、低次元空間におけるユークリッド距離の優位性を得られる上に、学習対象とする問題に応じた柔軟な距離尺度の設定が可能になる。さらに、この枠組みは入力ベクトルの事前の分析を必要としないために、オンライン追加の自己増殖型学習タスク、あるいは限られた入力ベクトルのサンプルによる実世界の学習タスクに適している。

[0018] また、前記第2の距離算出式は、コサイン距離に基づいてノード間の距離を算出するようにしてもよい。これによって、高次元データに対しても良好な精度を有することができる。

[0019] また、前記ユークリッド距離に基づく前記2つのノード間の最小距離値及び最大距離値と、前記他の距離尺度に基づく前記2つのノード間の最小距離値及び最大距離値と、のうちで少なくとも1つの値が変化したか否かを確認し、少なくとも1つの値が変化した場合にはノード密度を更新すると判定するノード密度更新判定手段と、を更に備え、前記ノード密度算出手段は、（1）前記第1の距離算出式及び前記第2の距離算出式をそれぞれ用いて前記第1勝者ノードの隣接ノードからの平均距離のベクトルの各成分を算出し、（2）当該算出した前記第1勝者ノードの前記隣接ノードからの平均距離のベクトルに基づいて前記第1勝者ノードのノード密度のポイント値のベクトルを算出し、（3）当該算出した前記第1勝者ノードの前記ノード密度のポイント値のベクトルに基づいて前記第1勝者ノードのノード密度の累積ポイント値のベクトルを算出し、（4）当該算出した前記第1勝者ノードの前記ノード密度の累積ポイント値のベクトルに基づいて前記第1勝者ノードのノード密度を算出し、（5）前記ノード密度更新判定手段による判定の結果、ノード密度を更新する場合には、変化前後における前記ユークリッド距離に基づく前記2つのノード間の最小距離値及び最大距離値と、変化前後における前記他の距離尺度に基づく前記2つのノード間の最小距離値及び最大距離値と、に基づいて、前記ネットワークに存在する全てのノードについて、注目するノードの前記ノード密度の累積ポイント値のベクトルを更新し、（6）当該更新した前記注目するノードの前記ノード密度の累積ポイント値のベクトルに基づいて前記注目するノードの前記ノード密度を再び算出するようにしてもよい。これにより、ノードが挿入されたことに起因して、ユークリッド距離に基づく2つのノード間の最小距離値及び最大距離値と、他の距離尺度に基づく2つのノード間の最小距離値及び最大距離値と、のうちで少なくとも1つの値が変化した場合においても、これらの変化を反映させたより正確なノード密度を自動的に算出することができる。

[0020] 本発明の一実施の形態に係る情報処理方法は、多次元ベクトルで記述されるノード及び当該ノード間を接続する辺が配置されるネットワーク構造を有

し、任意のクラスに属する入力ベクトルを順次入力して、当該入力ベクトルの入力分布構造を学習する情報処理装置における情報処理方法であって、勝者ノード探索手段が、入力される前記入力ベクトルに最も近い距離に位置するノード及び2番目に近い距離に位置するノードを第1勝者ノード及び第2勝者ノードとしてそれぞれ探索するステップと、クラス間ノード挿入判定手段が、前記入力ベクトルと前記第1勝者ノード間の距離及び前記入力ベクトルと前記第2勝者ノード間の距離に基づいて、クラス間ノード挿入を実行するか否かを判定するステップと、クラス間ノード挿入手段が、前記クラス間ノード挿入判定手段による判定の結果、前記クラス間ノード挿入を実行する場合に、前記入力ベクトルの成分と同一の成分を重みベクトルとして有するクラス間挿入ノードを生成し、当該生成したクラス間挿入ノードを前記ネットワークに挿入するステップと、勝者ノード学習時間算出手段が、前記クラス間ノード挿入判定手段による判定の結果、前記クラス間ノード挿入を実行しない場合に、前記第1勝者ノード及び前記第2勝者ノード間に辺を接続したときに、前記第1勝者ノードについてノードの学習時間を所定の値増加するステップと、重みベクトル更新手段が、前記クラス間ノード挿入判定手段による判定の結果、前記クラス間ノード挿入を実行しない場合に、注目するノードについて当該注目するノードと辺によって直接的に接続されるノードを隣接ノードとして、前記第1勝者ノードの重みベクトル及び前記第1勝者ノードの隣接ノードの重みベクトルをそれぞれ前記入力ベクトルに更に近づけるように更新するステップと、負荷平衡化手段が、前記入力ベクトルの総数に応じた所定の周期ごとに、前記ノードの学習時間の値が相対的に大きくかつ偏っているノードを検出し、当該検出したノード及びその隣接ノードの重みベクトルに基づく重みベクトルを有する新たなノードを生成して前記ネットワークに挿入し、前記検出したノードの学習時間及び前記検出したノードの隣接ノードの学習時間をそれぞれ減少し、前記検出したノード及びその隣接ノード間の辺を削除し、前記新たに挿入したノード及び前記検出したノード間と、前記新たに挿入したノード及び前記検出したノードのその隣接ノ

ード間と、に辺をそれぞれ接続するステップと、を実行する、ことを特徴とするものである。これによって、入力データの入力順序に影響を受けずに、入力パターンの接続構造を正確に記述することができるため、従来技術に比べてより優れた安定性能を有する。

[0021] 本発明の一実施の形態に係る非一時的なコンピュータ可読媒体は、上記情報処理をコンピュータに実行させることを特徴とするものである。これによって、入力データの入力順序に影響を受けずに、入力パターンの接続構造を正確に記述することができるため、従来技術に比べてより優れた安定性能を有する。

発明の効果

[0022] 本発明によれば、学習結果について優れた安定性を有する情報処理装置、情報処理方法及び非一時的なコンピュータ可読媒体を提供することができる。さらに、高次元データに対しても良好な精度を有する情報処理装置、情報処理方法及び非一時的なコンピュータ可読媒体を提供することができる。

図面の簡単な説明

[0023] [図1]本発明の課題を説明するための図である。
[図2A]本発明の課題を説明するための図である。
[図2B]本発明の課題を説明するための図である。
[図3]本発明の課題を説明するための図である。
[図4]本発明の課題を説明するための図である。
[図5]本実施の形態に係るサブクラスタ決定処理を説明するための図である。
[図6]本実施の形態に係るボロノイ領域の一例を説明するための図である。
[図7]LBS0INN及びE-S0INNの学習時間の分布を示すグラフである。
[図8]人工データセットⅠに対するノード密度の分布を示すグラフである。
[図9]人工データセットⅠに対するE-S0INNの実験結果を示すグラフである。
[図10]人工データセットⅠに対するLBS0INNの実験結果を示すグラフである。
[図11]本実施の形態に係る実験に使用する人工データセットⅠⅠを示す図である。

[図12A]人工データセットⅠⅠに対するLBS0INN及びE-S0INNの学習結果を示す図である。

[図12B]人工データセットⅠⅠに対するLBS0INN及びE-S0INNの学習結果を示す図である。

[図12C]人工データセットⅠⅠに対するLBS0INN及びE-S0INNの学習結果を示す図である。

[図12D]人工データセットⅠⅠに対するLBS0INN及びE-S0INNの学習結果を示す図である。

[図12E]人工データセットⅠⅠに対するLBS0INN及びE-S0INNの学習結果を示す図である。

[図12F]人工データセットⅠⅠに対するLBS0INN及びE-S0INNの学習結果を示す図である。

[図12G]人工データセットⅠⅠに対するLBS0INN及びE-S0INNの学習結果を示す図である。

[図12H]人工データセットⅠⅠに対するLBS0INN及びE-S0INNの学習結果を示す図である。

[図12I]人工データセットⅠⅠに対するLBS0INN及びE-S0INNの学習結果を示す図である。

[図12J]人工データセットⅠⅠに対するLBS0INN及びE-S0INNの学習結果を示す図である。

[図13]本実施の形態に係る実験に使用する人工データセットⅠⅠⅠを示す図である。

[図14]定常的な環境下における人工データセットⅠⅠに対するLBS0INNの学習結果を示す図である。

[図15]非定常的な環境下における人工データセットⅠⅠに対するLBS0INNの学習結果を示す図である。

[図16]本実施の形態に係る実験に使用するA T & T顔画像データセットのオリジナル画像である。

[図17] A T & T 顔画像データセットのオリジナル画像の特徴ベクトルを示す画像である。

[図18A] A T & T 顔画像データセットに対するE-S0INNによるクラスの個数の分布を示すグラフである。

[図18B] A T & T 顔画像データセットに対するLBS0INNによるクラスの個数の分布を示すグラフである。

[図19]異なる各クラスに対して異なるパーセントを入力したときの精度を示す。

[図20]本実施の形態に係る情報処理装置を実施するための構成例を示す図である。

[図21]本実施の形態に係る情報処理装置の機能ブロックを示す図である。

[図22]本実施の形態に係る情報処理装置による学習処理の処理概要を示すフローチャートである。

[図23]本実施の形態に係るサブクラスタの重複領域に位置するノードを説明するための図である。

[図24A]本実施の形態に係るサブクラスタの統合処理の様子を説明するための図である。

[図24B]本実施の形態に係るサブクラスタの統合処理の様子を説明するための図である。

[図24C]本実施の形態に係るサブクラスタの統合処理の様子を説明するための図である。

発明を実施するための形態

[0024] 本発明の実施の形態に係る具体的な構成・処理の説明に先立ち、本発明の理解をより容易とするために、本発明に関連する技術（以下、E-S0INNと称する。）の概要を簡単に説明する。

E-S0INNは、入力データの入力順序に学習結果が大きく依存するという問題を有している。図3を参照して、E-S0INNによる学習結果を説明する。図3は、E-S0INNによる処理（特許文献1の図5のS611、S607、S612な

どの処理を指す。具体的には、明細書の段落0071～0079に記載の処理を指す。)を実行した後の、ネットワークにおけるノードのノード密度の分布を示している。この例では、サブクラスタ間の重複領域が存在し、また、ノード密度の分布は大きな変動を有している。

[0025] 図3において、符号A及び符号Bは、E-SOINNによる処理によって生成された2つのサブクラスタをそれぞれ示す。なお、ネットワークにおいて辺によって接続されるノードの集合をクラスタとし、クラスタの部分集合をサブクラスタとする。ここで、サブクラスタとは、同一のサブクラスタラベルが付与されたノードからなる集合である。E-SOINNは、サブクラスタA及びサブクラスタBを統合するために、所定の条件(特許文献1の明細書の段落0076、数(14)、数(15)などにより示される。)を定めている。

[0026] E-SOINNは、新たな入力データがネットワークに入力されたときに、その第1勝者ノードがサブクラスタAに属し、かつ、第2勝者ノードがサブクラスタBに属し、さらに、(サブクラスタA及びBの重複領域に入力データが初めて現れたために、)サブクラスタA及びサブクラスタBと、サブクラスタC及びサブクラスタDとが統合されていないという状況下では、明らかに、サブクラスタA及びBを統合する条件をととても容易に満たしてしまう。E-SOINNは、サブクラスタA及びサブクラスタBを統合しようとするが、サブクラスタAはサブクラスタCに本来属すべきであり、サブクラスタBはサブクラスタDに本来属すべきである。従って、この問題は、サブクラスタの統合及び分離の繰り返しを引き起こしてしまう。図4は、3つのガウス分布を有する入力データに対するE-SOINNによる学習結果を示し、3つのガウス分布が互いに重複を有しており、また、複数のサブクラスタへの統合及び分離の乱れが行われている。

[0027] 以下、自己組織化ニューラルネットワーク(SOINN: Self-Organizing Incremental Neural Network)に関する技術として、本発明に係る技術を、LBSOINN(Load Balance Self-Organizing Incremental Neural Network)と称する。

[0028] LBS0INNは、ネットワークにおけるノードの負荷をノード学習時間として扱い、このノード学習時間を使用したノードの負荷の平衡化手法を提案する。LBS0INNは、この負荷平衡化手法によって、ネットワークをより安定化させることができる。このため、LBS0INNを負荷平衡 (Load Balance) と称している。また、LBS0INNは、ボロノイテセレーション (Voronoi Tessellation) に基づくサブクラスタ決定手法を使用することによって、クラス間の重複領域を削除する際のネットワーク構造の乱れを回避する。さらに、LBS0INNは、2つのノード間の類似度を示す距離尺度に関して、高次元データをオンライン追加学習するのに適した、新たな距離尺度を導入する。

[0029] また、詳細は後述するが、本発明に関する効果を確認するための実験を行い、LBS0INNの有効性を確認した。実験では、人工データセットに加え実世界での例に対してもLBS0INNを適用し、クラスタリングにおけるLBS0INNの有効性を実証した。全ての実験結果から、LBS0INNがE-S0INNよりも優れた安定性を有し、また、高次元空間における良好なパフォーマンスを有していることが示された。

[0030] 実施の形態1.

以下、図面を参照して本発明の実施の形態について説明する。

図20は、LBS0INNの一例としての、本実施の形態1に係る情報処理装置を実現するためのシステム構成の一例を示す図である。情報処理装置1は、専用コンピュータ、パーソナルコンピュータ (PC) などのコンピュータ10により実現可能である。但し、コンピュータは、物理的に単一である必要はなく、分散処理を実行する場合には、複数であってもよい。図20に示すように、コンピュータ10は、CPU11 (Central Processing Unit)、ROM12 (Read Only Memory) 及びRAM13 (Random Access Memory) を有し、これらがバス14を介して相互に接続されている。尚、コンピュータを動作させるためのOSソフトなどは、説明を省略するが、この情報処理装置を構築するコンピュータも当然備えているものとする。

[0031] バス14には又、入出力インターフェイス15も接続されている。入出力

インターフェイス 15 には、例えば、キーボード、マウス、センサなどよりなる入力部 16、CRT、LCD などよりなるディスプレイ、並びにヘッドフォンやスピーカなどよりなる出力部 17、ハードディスクなどより構成される記憶部 18、モデム、ターミナルアダプタなどより構成される通信部 19 などが接続されている。

[0032] CPU 11 は、ROM 12 に記憶されている各種プログラム、又は記憶部 18 から RAM 13 にロードされた各種プログラムに従って各種の処理、本実施の形態においては、例えば後述するノード密度算出手段 23 や負荷平衡化手段 33 などの各手段の処理を実行する。RAM 13 には又、CPU 11 が各種の処理を実行する上において必要なデータなども適宜記憶される。

[0033] 通信部 19 は、例えば図示しないインターネットを介しての通信処理を行ったり、CPU 11 から提供されたデータを送信したり、通信相手から受信したデータを CPU 11、RAM 13、記憶部 18 に出力したりする。記憶部 18 は CPU 11 との間でやり取りし、情報の保存・消去を行う。通信部 19 は又、他の装置との間で、アナログ信号又はデジタル信号の通信処理を行う。

[0034] 入出力インターフェイス 15 はまた、必要に応じてドライブ 20 が接続され、例えば、磁気ディスク 201、光ディスク 202、フレキシブルディスク 203、又は半導体メモリ 204 などが適宜装着され、それらから読み出されたコンピュータプログラムが必要に応じて記憶部 18 にインストールされる。

[0035] 続いて、本実施形態に係る情報処理装置 1 における各処理について、その主な処理を図 21 に示す機能ブロック図を用いて説明する。ハードウェア上は、各処理は実際にはソフトウェアと上記 CPU 11 などのハードウェア資源とが協働して実現している。

[0036] 情報処理装置 1 は、 n 次元ベクトルで記述されるノードが配置される非階層構造のニューラルネットワークを有し、入力情報取得手段 21、ノード密度更新判定手段 22、ノード密度算出手段 23、勝者ノード探索手段 24、

クラス間ノード挿入判定手段 25、クラス間ノード挿入手段 26、辺接続判定手段 27、辺接続手段 28、勝者ノード学習時間算出手段 29、重みベクトル更新手段 30、老齢辺削除手段 31、入力数判定手段 32、負荷平衡化手段 33、サブクラス決定手段 34、ノイズノード削除手段 35、学習終了判定手段 36、クラス決定手段 37、出力情報表示手段 38を含む。以下、各手段の詳細を説明する。

[0037] 本実施形態においては、情報処理装置 1 が有するニューラルネットワークは、入力ベクトルをニューラルネットワークに入力し、入力される入力ベクトルに基づいて、ニューラルネットワークに配置されるノードを自動的に増加させる自己増殖型ニューラルネットワークであり、また、ニューラルネットワークは非階層構造を有するものとして以下説明する。

[0038] 自己増殖型ニューラルネットワークを用いてノードを自動的に増加させることができるため、入力ベクトル空間からランダムに入力ベクトルが与えられる定常的な環境に限られず、例えば一定期間毎に入力ベクトルの属するクラスが切替えられて、切替後のクラスからランダムに入力ベクトルが与えられる非定常的な環境にも対応することができる。さらにまた、非階層構造を採用することで、他の層での学習を開始するタイミングを指定せずに追加学習を実施することができる。即ち、完全なオンラインでの追加学習を実施することができる。

[0039] 入力情報取得手段 21 は、情報処理装置 1 に入力として与えられる情報として、任意のクラスに属する n 次元の入力ベクトルを取得する。そして、取得された入力ベクトルを一時記憶部（例えば RAM 13）に格納し、一時記憶部に格納されたニューラルネットワークに対して順次入力する。

[0040] ノード密度更新判定手段 22 は、一時記憶部に格納されたノード、各距離尺度に基づくノード間の最小距離値及び最大距離値について、各距離尺度に基づくノード間の最小距離値及び最大距離値のうちで少なくとも 1 つの値が変化したか否かを確認し、少なくとも 1 つの値が変化した場合にはノード密度を更新すると判定し、その結果を一時記憶部に格納する。なお、各距離尺

度に基づくノード間の最小距離値及び最大距離値の詳細については後述する。

[0041] ノード密度算出手段23は、一時記憶部に格納されたノード、各距離尺度に基づくノード間の最小距離値及び最大距離値、隣接ノードからのノードの平均距離のベクトル、ノードのノード密度のポイント値のベクトル、ノード密度の累積ポイント値のベクトル、ノード密度について、第1の距離算出式及び第2の距離算出式をそれぞれ用いて第1勝者ノードの隣接ノードからの平均距離のベクトルの各成分を算出し、当該算出した第1勝者ノードの隣接ノードからの平均距離のベクトルに基づいて第1勝者ノードのノード密度のポイント値のベクトルを算出し、当該算出した第1勝者ノードのノード密度のポイント値のベクトルに基づいて第1勝者ノードのノード密度の累積ポイント値のベクトルを算出し、当該算出した第1勝者ノードのノード密度の累積ポイント値のベクトルに基づいて第1勝者ノードのノード密度を算出し、その結果を一時記憶部に格納する。なお、第1の距離算出式及び第2の距離算出式、第1勝者ノードのノード密度の算出処理の詳細については後述する。

[0042] また、ノード密度算出手段23は、一時記憶部に格納されたノード密度更新判定手段22による判定の結果、ノード密度を更新する場合には、一時記憶部に格納されたノード、ノード密度の累積ポイント値のベクトル、ノードの学習時間、各距離尺度に基づくノード間の最小距離値及び最大距離値について、変化前後におけるユークリッド距離に基づく2つのノード間の最小距離値及び最大距離値と、変化前後における他の距離尺度に基づく2つのノード間の最小距離値及び最大距離値と、に基づいて、ネットワークに存在する全てのノードについて、注目するノードのノード密度の累積ポイント値のベクトルを更新し、当該更新した注目するノードのノード密度の累積ポイント値のベクトルに基づいて注目するノードのノード密度を再び算出し、その結果を一時記憶部に格納する。なお、注目するノードの累積ポイント値のベクトルの算出処理の詳細については後述する。

- [0043] 勝者ノード探索手段24は、一時記憶部に格納される入力ベクトル及びノードについて、入力ベクトルに最も近い距離に位置するノードを第1勝者ノードとし、2番目に近い距離に位置するノードを第2勝者ノードとして探索し、その結果を一時記憶部に格納する。なお、本実施の形態に係る距離尺度の詳細については後述する。
- [0044] クラス間ノード挿入判定手段25は、一時記憶部に格納された入力ベクトル及びノードについて、入力ベクトルと第1勝者ノード間の距離及び入力ベクトルと第2勝者ノード間の距離に基づいて、クラス間ノード挿入を実行するか否かを判定し、その結果を一時記憶部に格納する。なお、本実施の形態に係るクラス間ノード挿入判定処理の詳細については後述する。
- [0045] クラス間ノード挿入手段26は、一時記憶部に格納されたクラス間ノード挿入判定手段25による判定結果に応じて、一時記憶部に格納された入力ベクトルについて、入力ベクトルが新たにネットワークに追加するノードであるものと見なし、その入力ベクトルの成分と同一の成分を重みベクトルとして有するクラス間挿入ノードを生成し、生成したクラス間挿入ノードをネットワークに挿入し、その結果を一時記憶部に格納する。
- [0046] 辺接続判定手段27は、一時記憶部に格納されたクラス間ノード挿入判定手段25による判定結果に応じて、一時記憶部に格納されたノード、ノードのサブクラスタラベルについて、ノードのサブクラスタラベルに基づいて、第1勝者ノード及び第2勝者ノードが属するサブクラスタをそれぞれ判定し、その結果を一時記憶部に格納する。ここで、ノードのサブクラスタラベルとは、そのノードが属するサブクラスタを示す。クラスタとは、混合クラスに含まれるノードについて、辺によって接続されるノードの集合を示し、サブクラスタは、同一のサブクラスタラベルが付与されたノードからなるクラスタの部分集合を示す。
- [0047] また、辺接続判定手段27は、一時記憶部に格納された判定の結果、第1勝者ノード及び第2勝者ノードが互いに異なるサブクラスタに属する場合には、一時記憶部に格納されたノード、ノードの密度、ノードのサブクラスタ

ラベルについて、第1勝者ノード及び第2勝者ノードのノード密度に関して、第1勝者ノードを含むサブクラスタの平均ノード密度を基準とする第1勝者ノードのノード密度の条件、又は、第2勝者ノードを含むサブクラスタの平均ノード密度を基準とする第2勝者ノードのノード密度の条件のうちの少なくとも1つを満たすか否かを更に判定し、その結果を一時記憶部に格納する。なお、辺接続判定手段27による辺接続判定処理の詳細については後述する。

[0048] 辺接続手段28は、一時記憶部に格納された辺接続判定手段27による判定の結果、第1勝者ノード及び第2勝者ノードの少なくとも1つがいずれのサブクラスタに属していない、又は、同一のサブクラスタに属する場合には、一時記憶部に格納されたノード、ノード間の辺について、第1勝者ノード及び第2勝者ノード間に辺を接続し、その結果を一時記憶部に格納する。

[0049] また、辺接続手段28は、一時記憶部に格納された辺接続判定手段27による判定の結果、上記の第1勝者及び第2勝者ノードのノード密度の条件のうちの少なくとも1つを満たす場合には、一時記憶部に格納されたノード、ノード間の辺について、第1勝者ノード及び第2勝者ノード間に辺を接続し、その結果を一時記憶部に格納する。

[0050] また、辺接続手段28は、一時記憶部に格納された辺接続判定手段27による判定の結果、上記の第1勝者及び第2勝者ノードのノード密度の条件のうちの少なくとも1つを満たさない場合には、一時記憶部に格納されたノード、ノード間の辺について、第1勝者ノード及び第2勝者ノード間に辺を接続せず（第1勝者ノード及び第2勝者ノード間に辺が存在する場合には辺を削除する）、その結果を一時記憶部に格納する。なお、辺接続手段28による勝者ノード接続処理の詳細については後述する。

[0051] 勝者ノード学習時間算出手段29は、一時記憶部に格納されたノード、ノードの学習時間について、第1勝者ノードの学習時間を所定の値増加し、その結果を一時記憶部に格納する。なお、本実施形態において、ノードの学習時間の詳細については後述する。

- [0052] 重みベクトル更新手段30は、一時記憶部に格納されたノード、ノードの重みベクトルについて、注目するノードについて注目するノードと辺によって直接的に接続されるノードを隣接ノードとして、第1勝者ノードの重みベクトル及び第1勝者ノードの隣接ノードの重みベクトルをそれぞれ入力ベクトルに更に近づけるように更新し、その結果を一時記憶部に格納する。
- [0053] 老齢辺削除手段31は、一時記憶部に格納されたノード、ノード間の辺、辺の年齢について、第1勝者ノードと直接的に接続される全ての辺の年齢を所定の値増加し、その結果を一時記憶部に格納する。また、老齢辺削除手段31は、一時記憶部に格納された辺について、予め設定され一時記憶部に格納された所定の閾値を超えた年齢を有する辺を削除し、その結果を一時記憶部に格納する。
- [0054] 入力数判定手段32は、一時記憶部に格納された与えられた入力ベクトルの総数について、与えられた入力ベクトルの総数が予め設定され一時記憶部に格納された所定の単位数の倍数であるか否かを判定し、その結果を一時記憶部に格納する。
- [0055] 負荷平衡化手段33は、一時記憶部に格納されたノード、ノード間の辺、ノードのサブクラスタラベル、ノードの重みベクトル、ノードの学習時間について、入力ベクトルの総数に応じた所定の周期ごとに、ノードの学習時間の値が相対的に大きくかつ偏っているノードを検出し、当該検出したノード及びその隣接ノードの重みベクトルに基づく重みベクトルを有する新たなノードを生成してネットワークに挿入し、検出したノードの学習時間及び検出したノードの隣接ノードの学習時間をそれぞれ減少し、検出したノード及びその隣接ノード間の辺を削除し、新たに挿入したノード及び検出したノード間と、新たに挿入したノード及び検出したノードのその隣接ノード間と、に辺をそれぞれ接続し、その結果を一時記憶部に格納する。なお、負荷平衡化手段33によるノードの負荷平衡化処理の詳細については後述する。
- [0056] サブクラスタ決定手段34は、一時記憶部に格納されたノード、ノード間の辺、ノードのサブクラスタラベル、ノード密度、ボロノイ領域について、

局所的に最大のノード密度を有するノードを頂点として、当該頂点に対してそれぞれ異なるサブクラスタラベルを付与し、当該サブクラスタラベルが付与されなかった全てのノードに対してノード密度が最大の隣接ノードと同一のサブクラスタラベルを付与し、頂点のうちでノード密度が所定の閾値よりも大きな頂点を基準としてボロノイ領域を生成し、当該生成したボロノイ領域において、基準とした頂点を含むサブクラスタと基準とした頂点とは異なる他の頂点を含むサブクラスタとが重複領域を有し、かつ、当該重複領域に位置するノードの平均ノード密度の条件を満たす場合に、基準とした頂点を含むサブクラスタのサブクラスタラベルを、他の頂点を含むサブクラスタのサブクラスタラベルとして付与し、その結果を一時記憶部に格納する。なお、サブクラスタ決定手段 34 による処理の詳細については後述する。

[0057] ノイズノード削除手段 35 は、一時記憶部に格納されたノード、ノード密度、ノード間の辺、隣接ノードの個数について、注目するノードについて、注目するノードの隣接ノードの個数及びノード密度に基づいて、注目するノードを削除し、その結果を一時記憶部に格納する。

[0058] 学習終了判定手段 36 は、情報処理装置 1 による学習処理を終了するか否かを判定する。

[0059] クラス決定手段 37 は、一時記憶部に格納されたノード、ノード間の辺、ノードのクラスについて、ノード間に生成された辺に基づいて、ノードの属するクラスを決定し、その結果を一時記憶部に格納する。

[0060] 出力情報表示手段 38 は、一時記憶部に格納されたノード及びノードのクラスについて、ノードの属するクラスのクラス数、及び各クラスのプロトタイプベクトルを出力する。

[0061] 次に、本実施の形態に係る情報処理装置 1 が有する特徴的な構成・処理を説明する。以下では、＜1：ノード間の負荷平衡化処理＞、＜2：サブクラスタの決定処理＞、＜3：新たな距離尺度の組み合わせの枠組み＞、＜4：類似度閾値の定義＞、＜5：新たなノード密度の定義及びその算出処理＞、＜6：辺接続判定処理及び辺接続処理＞の順に説明する。

[0062] <1 : ノード間の負荷平衡化処理>

E-S0INNは、第1勝者又は第2勝者の類似度閾値による判定のみを単純に使用して、新たな入力ベクトルであるクラス間ノードをネットワークに挿入すべきか否かを決定する。このため、ネットワークの学習を新たに開始しようとしたときにノードの接続構造が既に構築されていた場合には、新たなノードをネットワークに挿入することが困難になる。この構築済みのノードの接続構造は入力データの入力順序に大きく影響されるため、E-S0INNの学習結果は安定しない。また、図2に示したように、この欠点はノードの密度を算出する際には、入力データに対する低い精度をもたらしてしまう。

[0063] ノードの接続構造を正確に表現するためには、ノードの個数を十分な数にすることが重要であり、このこと自体はよく知られている。ネットワークのいくつかの領域においてノードの個数が十分でない場合、それらの領域におけるノードの学習時間の値が大きくかつ偏った状況になる。本実施の形態では、この状況をノードの負荷が不均衡な状態にあるものと定義する。本実施の形態では、この負荷の不均衡を解消するために、以下に示すノード間の負荷平衡化処理を導入する。

[0064] <1-1 : ノード間の負荷平衡化処理 (S101~S107) >

入力ベクトルの入力総数が所定の単位数(λ)の整数倍となった場合、負荷平衡化手段33は、一時記憶部に格納されたノード、ノード間の辺、ノードのサブクラスタラベル、ノードの重みベクトル、ノードの学習時間、ノードの密度について、ノードが属するサブクラスタの平均学習時間と比較して当該ノードの学習時間の値が相対的に大きくかつ偏ったノード(負荷が不均衡な状態にあるノード)を検出し、当該検出したノードとその隣接ノードの間に新たなノードを挿入し、当該検出したノードとその隣接ノードの学習時間を減少し、その結果を一時記憶部に格納する。これによって、負荷平衡化手段33は、負荷が不均衡な状態にあるノードについて負荷を平衡化させる。以下、ノードの負荷平衡化処理を具体的に説明する。なお、ノードの学習時間及びノードの密度の詳細については後述する。

[0065] S 1 0 1 : 負荷平衡化手段 3 3 は、一時記憶部に格納されたノード、ノードのサブクラスタラベル、ノードの学習時間について、ノードの学習時間に基づいて、各サブクラスタにおいて相対的に大きなノードの学習時間を有するノードを探索し、その結果を一時記憶部に格納する。負荷平衡化手段 3 3 は、例えば、各サブクラスタにおいて最大のノードの学習時間を有するノード q を探索し、その結果を一時記憶部に格納する。さらに、負荷平衡化手段 3 3 は、一時記憶部に格納されたノード、ノードの学習時間について、検出したノード q の学習時間が、相対的に大きくかつ偏っているか否かを判定し、その結果を一時記憶部に格納する。負荷平衡化手段 3 3 は、例えば、ノード q の学習時間 M_q 及びノード q が属するサブクラスタの平均学習時間 $M_{c,q}^{average}$ について、 $M_q > 3 M_{c,q}^{average}$ が成立するか否かを判定し、その結果を一時記憶部に格納する。なお、負荷平衡化手段 3 3 は、ノード q が属するサブクラスタに含まれる全てのノードの学習時間を平均することによって、ノード q が属するサブクラスタの平均学習時間 $M_{c,q}^{average}$ を算出し、その結果を一時記憶部に格納する。一時記憶部に格納された判定の結果、 $M_q > 3 M_{c,q}^{average}$ が成立する場合には、負荷平衡化手段 3 3 は、以下のようにして新たなノードを追加し、 $M_q > 3 M_{c,q}^{average}$ が成立しない場合には、ノードの追加を行わない。

[0066] S 1 0 2 : 負荷平衡化手段 3 3 は、一時記憶部に格納されたノード、ノードの学習時間について、ノードの学習時間に基づいて、ノード q の隣接ノードのうちで、例えば最大の学習時間を有する隣接ノード f を決定し、その結果を一時記憶部に格納する。例えば、負荷平衡化手段 3 3 は、一時記憶部に格納される以下の式を満たす隣接ノード f を決定し、その結果を一時記憶部に格納する。ここで、 N_q は、ノード q の隣接ノード集合を示す。

[数3]

$$f = \arg \max_{c \in N_q} M_c$$

... (3)

[0067] S 1 0 3 : 負荷平衡化手段 3 3 は、一時記憶部に格納されたノードについて、新たなノード r を生成してネットワークに追加し、その結果を一時記憶部に格納する。また、負荷平衡化手段 3 3 は、一時記憶部に格納されたノード、ノードの重みベクトルについて、ノード r の重みベクトル W_r を、ノード q の重みベクトル W_q 及び隣接ノード f の重みベクトル W_f を用いて補間して算出し、その結果を一時記憶部に格納する。例えば、負荷平衡化手段 3 3 は、一時記憶部に格納される以下の式に示す操作及び算出処理を実行し、その結果を一時記憶部に格納する。

[数4]

$$A = A \cup \{r\}, W_r = (W_q + W_f)/2.0 \quad \dots (4)$$

[0068] S 1 0 4 : 負荷平衡化手段 3 3 は、一時記憶部に格納されたノード、ノードの学習時間について、ノード r の学習時間 M_r を、ノード q の学習時間 M_q 及び隣接ノード f の学習時間 M_f を用いて補間して算出し、その結果を一時記憶部に格納する。例えば、負荷平衡化手段 3 3 は、一時記憶部に格納される以下の式に示す算出処理を実行し、その結果を一時記憶部に格納する。ここで、 d は入力ベクトルの次元数を示す。

[数5]

$$M_r = \frac{1}{d}(M_q + M_f) \quad \dots (5)$$

[0069] S 1 0 5 : 負荷平衡化手段 3 3 は、一時記憶部に格納されたノード、ノードの学習時間について、ノード q の学習時間 M_q 及び隣接ノード f の学習時間 M_f を所定の割合それぞれ減少し、その結果を一時記憶部に格納する。例えば、負荷平衡化手段 3 3 は、一時記憶部に格納される以下の式に示す算出処理を実行し、その結果を一時記憶部に格納する。ここで、 d は入力ベクトルの次元数を示す。

[数6]

$$M_q = \left(1 - \frac{1}{2d}\right)M_q, M_f = \left(1 - \frac{1}{2d}\right)M_f \quad \dots (6)$$

[0070] S106：負荷平衡化手段33は、一時記憶部に格納されたノード、ノードの密度について、ノードrの密度 h_r を、ノードqの密度 h_q 及び隣接ノードfの密度 h_f を用いて補間して算出し、その結果を一時記憶部に格納する。例えば、負荷平衡化手段33は、一時記憶部に格納される以下の式に示す算出処理を実行し、その結果を一時記憶部に格納する。

[数7]

$$h_r = \frac{1}{d}(h_q + h_f) \quad \dots (7)$$

[0071] S107：負荷平衡化手段33は、一時記憶部に格納されたノード、ノード間の辺について、新たなノードrとノードqを接続する辺と、新たなノードrと隣接ノードfを接続する辺と、を生成し、また、ノードq及び隣接ノードfの間に元々存在する辺を削除し、その結果を一時記憶部に格納する。例えば、負荷平衡化手段33は、一時記憶部に格納される以下の式(8)、式(9)に示す操作を実行し、その結果を一時記憶部に格納する。ここで、Cは辺集合を示し、例えば(r, q)はノードr及びノードq間を接続する辺を示す。

[数8]

$$C = C \cup \{(r, q), (r, f)\} \quad \dots (8)$$

[数9]

$$C = C \setminus \{(q, f)\} \quad \dots (9)$$

[0072] 上述したように、情報処理装置 1 は、ノードが属するサブクラスタの平均学習時間を基準レベルとして採用し、ネットワークにノードを新たに挿入するか否かという追加学習のタスクに適用する。このため、事前の学習結果により得られた知識（ノード及びその接続構造を含むネットワーク）は、新たに入力される入力ベクトルに影響しない。この処理によって、情報処理装置 1 によるノード接続構造の生成処理は、E-S0INNとは異なったものになる。

[0073] 情報処理装置 1 は、ネットワークに入力ベクトルを新たに挿入すべきか否かに関し、第 1 勝者又は第 2 勝者の類似度閾値を使用してノードの挿入を行うクラス間ノード挿入処理に加えて、負荷が不均衡な状態にあるノード（即ち、ノードの学習時間の値が相対的に大きくかつ偏ったノード）を検出して、この検出したノードの近くに新たなノードを挿入する負荷平衡化処理を実行する。これによって、ネットワークにおけるノードの負荷を平衡化させることができる。そして、情報処理装置 1 は、上述した S 1 0 1 ~ S 1 0 7 に示す新たなノード挿入と、後述する S 5 0 7 に示すノード挿入と、の両方がクラス間挿入に属する場合、ネットワークにおけるノードの個数を十分なものにすることができ、この結果、入力データの密度分布を正確に反映することができる。従って、入力データの入力順序に影響を受けずに、入力パターンの接続構造を正確に記述することができるため、従来技術に比べてより優れた安定性能を有する。

[0074] < 2 : サブクラスタの決定処理 >

入力ベクトルの入力個数が所定の単位数（ λ ）の整数倍となった場合、上述した負荷平衡化処理に加えて、サブクラスタ決定手段 3 4 は、一時記憶部に格納されたノード、ノード間の辺、ノードのサブクラスタラベル、ノードの密度、ボロノイ領域について、ボロノイテセレーション（Voronoi Tessell

ation) に基づくサブクラスタの決定処理を実行し、その結果を一時記憶部に格納する。なお、サブクラスタの決定処理とは、ネットワークに存在するノードに対してサブクラスタラベルを付与することによって、当該ノードが属するサブクラスタを決定する処理である。

[0075] 上述したように、E-S0INNによる処理はサブクラスタの統合及び分離の繰り返しを引き起こしてしまう。これは、E-S0INNによる処理（特許文献1の図5のS611の処理を指す。具体的には、明細書の段落0071に記載の処理を指す。）の実行後にネットワークに多くのサブクラスタが存在し、さらに、E-S0INNによる処理（特許文献1の図5のS607、S612などの処理を指す。具体的には、明細書の段落0072～0079に記載の処理を指す。）によって、真のサブクラスタの外縁に位置するサブクラスタが容易に統合されてしまうためである。情報処理装置1は、以下に示すサブクラスタの統合を目的とするサブクラスタ決定処理を導入することによって、E-S0INNが有する問題に対処する。

[0076] 本実施の形態では、以下の処理についての理解をより容易とするために、“サブクラスタ”という用語に代えて“隆起 (tuber)”という用語を使用する。後述するように、本実施の形態では、局所的に最大のノード密度を有するノード及び当該ノードと辺によって接続されるノードからなる集合を記述するために、この隆起という用語を使用する。なお、隆起とは通常、山の盛り上がりを示すが、本実施の形態では、あるクラスに含まれる複数のサブクラスタに関して各サブクラスタが含むノードのノード密度の高低を考慮して、ノード密度の盛り上がりを表現するために隆起という用語を用いる。例えば、図5において、A、B、C、Dはそれぞれ隆起を示す。A及びCはそれぞれ同一のクラスに属し、B及びDはそれぞれ他の同一のクラスに属する。

[0077] 後述するように、本実施の形態では、サブクラスタ決定処理を、各ボロノイ領域において、ノード密度が高密度の隆起から低密度の隆起へと向けて実行する。図5において、例えば、隆起Aからサブクラスタ決定処理を開始した場合、隆起Bは隆起Aと容易に統合し、その結果、ネットワークもまた乱

れてしまう。さらに、隆起のサブクラスタ決定処理は、他の隆起と統合することを回避するように、正確に停止する必要がある。また、図5において、例えば、隆起Cのサブクラスタ決定処理では、隆起Bと統合しないようにする必要がある。

[0078] 隆起の決定処理を以下に示す。この処理は、分離及び降下平滑化アルゴリズム (SDSA : Separate and Downhill smooth algorithm) とも呼ばれる。なお、以下の処理では、局所的に最大のノード密度を有するノードを、隆起の頂点と称する。言い換えると、隆起の頂点とは、対応する隆起に含まれるノードのうちで最大のノード密度を有するノードを示す。

[0079] <2-1 : サブクラスタの決定処理 (S 2 0 1 ~ S 2 0 5、S 3 0 1 ~ S 3 0 5) >

S 2 0 1 : サブクラスタ決定手段34は、一時記憶部に格納されたノード、ノードの密度について、混合クラスにおける全てのノードについて局所的に最大のノード密度を有するノードを探索し、その探索したノードを隆起の頂点としてノード集合LMに追加し、その結果を一時記憶部に格納する。サブクラスタ決定手段34は、一時記憶部に格納されたノード、ノードのサブクラスタラベル (隆起ラベル) について、探索した頂点全てに対してそれぞれ異なる隆起ラベルを付与し、その結果を一時記憶部に格納する。

[0080] S 2 0 2 : サブクラスタ決定手段34は、一時記憶部に格納されたノード、ノード間を接続する辺、ノードのサブクラスタラベル (隆起ラベル) について、S 2 0 1 において隆起ラベルが付与されなかった全てのノードに対して、ノード密度が最大の隣接ノードと同じ隆起ラベルを付与する。即ち、S 2 0 1 において、各頂点に対してそれぞれ異なる隆起ラベルが付与される。そして、S 2 0 2 において、頂点を除いた他の全てのノードのそれぞれに対して、当該ノードの隣接ノードのうちで最大のノード密度を有するノードの隆起ラベルが付与される。このようにして全てのノードに対して隆起ラベルが付与され、同一の隆起ラベルが付与されたノードからなる集合を隆起 (サブクラスタ) として扱うことで、混合クラスを、複数の隆起 (サブクラスタ

）に分離する。ここで、注目するノードとその隣接ノードが互いに異なる隆起ラベルを有する場合には、その注目するノード及びその隣接ノードは、分布の重複領域に位置する可能性がある。

[0081] S 2 0 3 : サブクラスタ決定手段 3 4 は、一時記憶部に格納されたノード、ノードの密度について、頂点のノード集合 L M においてノード密度が β より大きなノードを探索し、探索したノードをノード集合 K に追加し、その結果を一時記憶部に格納する。例えば、サブクラスタ決定手段 3 4 は、一時記憶部に格納される以下の式に示す算出処理を実行することによって値 β を算出し、その結果を一時記憶部に格納する。ここで、 γ はユーザーによって予め適当な値が決定されて設定されるパラメータ ($1 < \gamma < 2$) を示し、 h_{mean} は、頂点の集合 L M に含まれる全ての頂点のノードの密度の平均値を示し、 h_{max} はノード集合 L M に含まれる頂点のノード密度のうちで最大のノード密度を示す。

[数10]

$$\beta = \min \left(\gamma^2 h_{mean}, \frac{1}{\gamma} h_{max} \right) \quad \dots (10)$$

[0082] S 2 0 4 : サブクラスタ決定手段 3 4 は、一時記憶部に格納されたノード、ノードの重みベクトル、ボロノイ領域について、ノード集合 K に基づいて、頂点のノード集合 L M についてのボロノイ領域（図 6 を参照。）を生成し、その結果を一時記憶部に格納する。例えば、サブクラスタ決定手段 3 4 は、一時記憶部に格納される以下の式（11）に示す操作を実行し、その結果を一時記憶部に格納する。式（11）に示すボロノイ領域の生成処理では、ノード集合 K に含まれるノード K_i をシード（seed）として、このシードに対応するボロノイ領域 V_i を生成する。即ち、ノード集合 K に含まれるノード K_i は、ボロノイ領域 V_i における i 番目のシードとなる。このように、S 2 0 1 ~ 2 0 4 に示す処理では、ノード集合 L M は全ての頂点を含み、ノード集合 K は全ての頂点のうちでノード密度が β よりも高い頂点を含む。そして、

ノード集合Kをボロノイ領域のシードとして使用する。例えば図6において、このようなシードを○（白丸）を用いて示している。

[数11]

$$V_i = \{c \in LM | i = \arg \min_{j \in K} D(c, j)\} \quad \dots (11)$$

なお、ボロノイ領域は人工知能及び数学の分野における基礎的な知識であり（例えば、http://en.m.wikipedia.org/wiki/Voronoi_diagram を参照。）
、上述したボロノイ領域の生成処理は、公知のボロノイ領域生成処理を実行すればよい。このため、ここでは、その詳細な説明を省略する。

[0083] S 2 0 5 : サブクラスタ決定手段34は、一時記憶部に格納されたノード、ノード間の辺、ノードの密度、ノードのサブクラスタラベル（隆起ラベル）、ボロノイ領域について、以下のS 3 0 1 ~ S 3 0 5 に示す処理を実行し、S 2 0 4 において生成した全てのボロノイ領域におけるサブクラスタの統合を行うことによってサブクラスタを決定し、その結果を一時記憶部に格納する。ここで、ボロノイ領域においてサブクラスタを決定するとは、各ボロノイ領域に含まれる隆起についてノード密度に関する所定の条件を満足する場合に、各隆起の隆起ラベルをノード K_i の隆起ラベルに統合する処理を指す。なお、所定の条件とは、例えば、後述する式（12）又は式（13）のいずれかを満足することを指す。

[0084] S 3 0 1 : サブクラスタ決定手段34は、一時記憶部に格納されたノード、ノードのサブクラスタラベル（隆起ラベル）、ボロノイ領域について、ノードに付与された隆起ラベルに基づいて、あるボロノイ領域 V_i において、当該ボロノイ領域 V_i に含まれるノード K_i の隆起（ K_i ）とノードjを含む隆起（j）とが重複領域を有するか否かを判定し、その結果を一時記憶部に格納する。なお、ここでは、理解を容易とするため、ノード K_i が属する隆起を隆起（ K_i ）とし、ノード K_i に付与された隆起ラベルを隆起ラベル（ K_i ）とし、ノードjが属する隆起を隆起（j）とし、ノードjに付与された隆起ラベ

ルを隆起ラベル (j) として説明する。また、ここでのノード j とは、頂点の集合 L_M に含まれるノードであって、且つ、ボロノイ領域 V_i に含まれるノードを指す。

[0085] サブクラスタ決定手段 34 は、ノード K_i の隆起 (K_i) とノード j を含む隆起 (j) とが重複領域を有するか否かを、例えば、以下のようにして判定する。サブクラスタ決定手段 34 は、一時記憶部に格納されたノード、ノード間の辺、ノードのサブクラスタラベル (隆起ラベル) について、ネットワークに含まれるノード間の辺およびノードに付与された隆起ラベルに基づいて、その一端のノードの隆起ラベルが隆起ラベル (K_i) であり、かつ、他端のノードの隆起ラベルが隆起ラベル (j) である辺が存在するか否かを判定し、判定の結果、そのような辺が存在する場合には、ノード K_i の隆起 (K_i) とノード j を含む隆起 (j) とが重複領域を有するものと判定し、その結果を一時記憶部に格納する。

[0086] S302: サブクラスタ決定手段 34 は、一時記憶部に格納された判定の結果、ノード K_i の隆起 (K_i) とノード j を含む隆起 (j) とが重複領域を有する場合、ノード集合 O_{K_i} にノード j を追加する。ノード集合 O_{K_i} とは、ノード K_i の隆起 (K_i) とノード j を含む隆起 (j) とが重複領域を有する場合に、ノード K_i の隆起 (K_i) に対して統合する可能性のある隆起 (j) を格納しておくための集合である。サブクラスタ決定手段 34 は、ノード K_i の隆起 (K_i) とノード j を含む隆起 (j) とが重複領域を有しない場合には、S301 へと処理を進め、他のボロノイ領域 V_i について処理を行う。

[0087] S303: サブクラスタ決定手段 34 は、一時記憶部に格納されたノード集合 O_{K_i} について、ノード集合 O_{K_i} が空集合であるか否かを判定し、その結果を一時記憶部に格納する。サブクラスタ決定手段 34 は、一時記憶部に格納された判定の結果、ノード集合 O_{K_i} が空集合である場合には、S301 へと処理を進め他のボロノイ領域 V_i について処理を行い、ノード集合 O_{K_i} が空集合でない場合には、S304 へと処理を進める。

[0088] S304: サブクラスタ決定手段 34 は、例えば、ノード集合 O_{K_i} に含ま

れるノード j それぞれについて、一時記憶部に格納される以下の式 (12) 又は式 (13) のいずれか 1 つを満足するか否かを判定し、その結果を一時記憶部に格納する。サブクラスタ決定手段 34 は、一時記憶部に格納された判定の結果、以下の式 (12) 又は式 (13) のいずれか 1 つを満足する場合には、ノード $j \in O_{K_i}$ の隆起ラベルとして、ノード K_i の隆起ラベル (K_i) を付与する。即ち、サブクラスタ決定手段 34 は、ノード j の隆起ラベル (j) をノード K_i の隆起ラベル隆起 (K_i) に変更することによって、ノード j が属する隆起 (j) を、ノード K_i が属する隆起 (K_i) に統合する。

[数12]

$$mean_{OR} > (\theta^{\frac{h_{K_i}}{mean_{K_i}}} - 1) mean_{K_i} \dots (12)$$

[数13]

$$mean_{OR} > (\theta^{\frac{h_j}{mean_j}} - 1) mean_j \dots (13)$$

式 (12) 及び式 (13) において、 h_{K_i} は、ノード K_i のノード密度を示す。 $mean_{K_i}$ は、隆起 (K_i) に含まれるノード (ネットワークに含まれるノード) のノード密度の平均ノード密度を示す。 h_j は、ノード j のノード密度を示す。 $mean_j$ は、隆起 (j) に含まれるノード (ネットワークに含まれるノード) のノード密度の平均ノード密度を示す。

[0089] ここで、 $mean_{OR}$ の定義を説明する。ノード集合 O_{K_i} に含まれるノード j の隆起 (j) について、隆起 (K_i) と隆起 (j) とが重複領域を有する場合には、隆起ラベル (K_i) であるノードと、隆起ラベル (j) であるノードとを接続する辺が存在する。この辺によって直接的に接続される各ノードを、隆起 (K_i) と隆起 (j) の重複領域に位置するノードと称する。 $mean_{OR}$ は、このようなノードのノード密度の平均ノード密度を示す。例えば、図 23 に例示する場合では、隆起 (K_i) と隆起 (j) とが 2 つの辺によって接

続されており、これらの隆起は重複領域を有している。そして、これらの辺により接続されるノードとして、隆起 (K_i) の2つのノードと、隆起 (j) の2つのノードが存在し、これらの4個のノードが隆起 (K_i) と隆起 (j) の重複領域に位置する。 $mean_{OR}$ は、これらの4個のノードのノード密度を平均して算出する。

[0090] また、 θ は、ユーザーによって予め適切な値が決定されて設定されるパラメータを示し、その値は $[1, 2]$ の範囲内において定められる。また、 θ は許容度の要素のパラメータであり、1つのクラスにおいて、隆起 (K_i) と隆起 (j) との間の差異をどの程度許容するのかを決定するために使用される。 θ が増加するにつれてこの許容度はより小さくなる。

[0091] S 3 0 5 : サブクラスタ決定手段 3 4 は、一時記憶部に格納されたノード集合 O_{K_i} を空集合に設定し、ステップ S 3 0 1 へと処理を進め、再びステップ S 3 0 1 以降の処理を開始する。

[0092] S 3 0 1 ~ S 3 0 5 の処理の様子を、図 2 4 (A)、(B)、(C) を参照して説明する。

図 2 4 (A) に示すように、ボロノイ領域 V_1 において、隆起 (K_1)、隆起 (j_1)、隆起 (j_2)、隆起 (j_3)、隆起 (j_4)、隆起 (j_5)、隆起 (j_6)、隆起 (j_7)、隆起 (j_8) が含まれている。図において、各隆起を破線による白抜き丸を用いて示す。なお、各ノードおよびノード間の辺については図示を省略した。

[0093] サブクラスタ決定手段 3 4 は、S 3 0 1 において、隆起 (K_1) と隆起 (j_1) が重複領域を有しているか否か、・・・、隆起 K_1 と隆起 j_8 が重複領域を有しているか否か、をそれぞれ判定する。サブクラスタ決定手段 3 4 は、S 3 0 2 において、隆起 (K_1) と隆起 (j_1) が重複領域を有している場合には、隆起 (j_1) のノード j_1 を集合 O_{K_1} に追加し、隆起 (K_1) と隆起 (j_2) が重複領域を有している場合には、隆起 (j_2) のノード j_2 を集合 O_{K_1} に追加し、隆起 (K_1) と隆起 (j_3) が重複領域を有している場合には、隆起 (j_3) のノード j_3 を O_{K_1} に追加する。この結果、集合

O_{K1} は、ノード $j1$ 、 $j2$ 、 $j3$ を含む。

[0094] 次に、サブクラスタ決定手段34は、S303において、集合 O_{K1} が空集合でないために、次のS304へと進む。サブクラスタ決定手段34は、S304において、ノード $j1$ が属する隆起 ($j1$) について式 (12) または (13) が成立するか判定し、成立する場合には、ノード $K1$ の隆起ラベル ($K1$) をノード $j1$ の隆起ラベルとして付与する。また、サブクラスタ決定手段34は、S304において、ノード $j2$ が属する隆起 ($j2$) について式 (12) または (13) が成立するか判定し、成立する場合には、ノード $K1$ の隆起ラベル ($K1$) をノード $j2$ の隆起ラベルとして付与する。また、サブクラスタ決定手段34は、S304において、ノード $j3$ が属する隆起 ($j3$) について式 (12) または (13) が成立するか判定し、成立する場合には、ノード $K1$ の隆起ラベル ($K1$) をノード $j3$ の隆起ラベルとして付与する。この処理の結果、隆起ラベルに基づく各隆起の状態を、図24 (B) に示す。

[0095] 次に、サブクラスタ決定手段34は、S305において、集合 O_{K1} を空集合に設定し、ステップ301へと進む。サブクラスタ決定手段34は、S301において、隆起 ($K1$) と隆起 ($j4$) が重複領域を有しているか否か、 $\dots$ 、隆起 ($K1$) と隆起 ($j8$) が重複領域を有しているか否か、をそれぞれ判定し、S302において、これらの全ての隆起 ($j4$) ~ ($j8$) が隆起 ($K1$) と重複領域を有している場合には、隆起 ($j4$) ~ ($j8$) のノード $j4$ ~ $j8$ を集合 O_{K1} に追加する。この結果、集合 O_{K1} は、ノード $j4$ ~ $j8$ を含む。サブクラスタ決定手段34は、S303を経てS304において、ノード $j4$ ~ $j8$ がそれぞれ属する隆起 ($j4$) ~ ($j8$) について、式 (13) または (13) が成立するかそれぞれ判定し、成立する場合には、ノード $K1$ の隆起ラベル ($K1$) を、ノード $j4$ ~ $j8$ の隆起ラベルとして付与する。この処理の結果、隆起ラベルに基づく各隆起の状態を、図24 (C) に示す。このようにして、サブクラスタ決定手段34は、各ボロノイ領域において、ボロノイ領域のシードとした頂点が属する隆起を中

心として、他の隆起の統合を繰り返す。

[0096] サブクラスタ決定処理は、高いノード密度の隆起から低いノード密度の隆起へと向けて開始すべきである。このため、式(10)の目的は、ノード密度の境界 β を探索して、その境界 β の密度がどの程度高いかを決定することである。ノード密度がこの境界 β よりも高い隆起は、対応するサブクラスタの中心になりやすい。このため、この種の隆起のノードを、頂点のノード集合Kに追加する。なお、本実施の形態では、隆起に含まれるノードのうちで、局所的に最大のノード密度を有するノードのみをノード集合Kに追加することによって、メモリを節約している。さらに、そのノード密度が境界 β よりも低い全ての他の隆起は、ノード集合Kにおいて対応するボロノイ領域の要素となる。

[0097] そして、サブクラスタ決定処理を、ノード集合Kにおける要素(ノード K_i)から開始する。加えて、サブクラスタ決定処理は、同一のボロノイ領域における隆起間においてのみ行う。異なるサブクラスタに属するかもしれない隆起を一緒に統合してしまうことは回避すべきであるため、同一のボロノイ領域における隆起間についてのみサブクラスタ決定処理を行うことはとても重要である。このような目的を実現するために、本実施の形態では、S301～S305に示した処理を実行する。

[0098] S301～S305に示したサブクラスタ決定処理は、各ボロノイ領域の中心の隆起から開始する。S301～S305に示したサブクラスタ決定処理は繰り返し処理であり、隆起間に重複領域が存在し、かつ、ノード密度に関する上述した所定の条件を満たす場合に、ボロノイ領域の中心の隆起は、重複領域を有する隆起を統合する。なお、各繰り返し処理において、当初は重複領域を有していなかったいくつかの隆起について、最後の繰り返し処理において、重複領域を有することがある。S301～S305に示した処理は、異なる方向に向けて山を降りていく処理にとっても類似する処理である。

[0099] 図5は、E-S0INNによって生成された変動を有する分布と、情報処理装置1によって生成されてサブクラスタが統合された分布を概念的に示している。

本実施の形態に係るS201～S205及びS301～S305の処理によれば、明らかに、異なるサブクラスに混合クラスを分離するときの乱れを回避することができる。言い換えると、情報処理装置1は、E-SOINNよりも安定であると言える。

[0100] <3：新たな距離尺度の組み合わせの枠組み>

E-SOINNは、ユークリッド距離を使用して最近傍ノード（最も近い距離に位置するノード）を探索する。しかしながら、入力データが高次元である場合には、ユークリッド距離を使用すると距離が集中するように見られ、データ要素間での全ての距離がとても類似するようになってしまう（非特許文献：Verleysen, M., Francois, D., "The Concentration of Fractional Distances", IEEE Trans. on Knowledge and Data Engineering 19(7), 873-886 (2007).を参照。）。この現象は、“次元の呪い”と呼ばれている。次元の呪いを打倒するためにこれまでに多くの手法が提案されている。

[0101] 例えば、高次元において近似する類似探索問題を解決するために、局所性鋭敏型ハッシュ（LSH: Locality Sensitive Hashing）手法が広く用いられている（非特許文献：Aristides Gionis, Piotr Indyk, Rajeev Motwani, "Similarity Search in High Dimensions via Hashing", Proceedings of the 25th International Conference on Very Large Data Bases, p.518-529, September 07-10, 1999.や、非特許文献：M. Datar, N. Immorlica, P. Indyk, and V.S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions", in Proc. Symposium on Computational Geometry, 2004, pp.253-262.を参照。）。しかし、LSHは、メモリ消費量が大きすぎるという大きな制約があり（非特許文献：Xiaoguang Gu, Yongdong Zhang, Lei Zhang, Dongming Zhang and Jintao Li, "An improved method of locality sensitive hashing for indexing large-scale and high-dimensional features," in Press, Signal Processing, 2012.を参照。）、オンライン追加学習タスクにLSHを使用することができない。さらに、コンテンツベースの類似サーチ手法として、関連するフィードバックアルゴリズムが与えられて、最適な

ノルムを選択するという手法がある。しかしながら、この手法もまた、オンライン追加学習タスクには使用することができない（非特許文献：Verleysen, M., Francois, D., "The Concentration of Fractional Distances", IEE E Trans. on Knowledge and Data Engineering 19(7), 873-886 (2007).を参照。）。

[0102] 入力データの次元が低次元である場合、ユークリッド距離に基づく距離は有効である。入力データが高次元である場合に有効な距離尺度としては、例えば、コサイン距離、マンハッタン距離 (Manhattan distance)、フラクショナル距離 (fractional distance) などのいくつかが提案されている。しかし、オンライン追加学習環境においては、サンプルデータを事前に得ることができない。言い換えると、事前に入力ベクトルの次元数を分析し、どの距離尺度が有効であるかを決定することができない。このため、本実施の形態では、距離尺度の組み合わせの枠組みを提案し、高次元空間において有効な他の距離尺度とユークリッド距離を統合する。これによって、次元の呪いを克服する。この距離尺度の組み合わせの枠組みは、オンライン自己増殖型追加学習タスクに適したものである。

[0103] 本実施の形態に係る距離尺度の組み合わせの枠組みである距離算出式を、例えば、以下の式 (14) に示す。ここで、 $D(p, q)$ はノード p 及びノード q の間の距離を示す。 d は入力ベクトルの次元数を示し、 n は距離尺度の個数を示す。 η は、ユーザーによって予め適当な値が決定されて設定されるパラメータである。なお、 η は、実験を経て予め適当な値を決定し、ユーザーによって当該値を設定するものとしてもよいし、入力ベクトルの次元数を変数とする関数を情報処理装置 1 に設定し、情報処理装置 1 が、当該関数を用いて入力データの次元数に応じた適当な値を自ら算出して設定するようにしてもよい。

[0104] 式 (14) に示す距離算出式は、第 1 項部分と、第 2 項部分を含む。第 1 項部分は、ユークリッド距離に基づいてノード間の距離を算出する第 1 の距離算出式の一例である。第 2 項部分は、ユークリッド距離とは異なる他の距

離尺度に基づいてノード間の距離を算出する、複数の距離算出式の一例である。第1項部分及び第2項部分はそれぞれ重み付けがされており、当該重み付けは、パラメータ η 及び d に応じて設定されている。後述するように、入力ベクトルの次元数 d が増加するに従って、第1の距離算出式の重みを減少させるように設定されている。

[0105] $E U_{pq}$ は、ノード p 及びノード q の間のユークリッド距離に基づく距離を示し、式(15)により算出する。 $E U_{min}$ 及び $E U_{max}$ は、ネットワークにおける任意の2つのノード間の最小のユークリッド距離及び最大のユークリッド距離をそれぞれ示す。 $E U_{max}$ の値は減少せずに増加するだけであり、このことは、 $E U_{max}$ の値が、学習の過程においてそれまでに現れた最大のユークリッド距離であることを示す。また、学習フェーズが続く限り、 $E U_{min}$ の値は増加せずに減少するだけであり、このことは、 $E U_{min}$ の値が、学習の過程においてそれまでに現れた最小のユークリッド距離であることを示す。

[0106] D_{ipq} 、 D_{imin} 、 D_{imax} は、 i 番目の距離尺度に関するものである。 D_{ipq} は、ノード p 及びノード q の間の i 番目の距離尺度に基づく距離を示す。 D_{imin} 及び D_{imax} は、ネットワークにおける任意の2つのノード間の i 番目の距離尺度に基づく最小の距離値及び最大の距離値をそれぞれ示す。即ち、これらは $E U_{pq}$ 、 $E U_{min}$ 、 $E U_{max}$ とそれぞれ同一の意味合いを有する。

[0107] 各距離尺度に基づく距離は、互いに異なるスケールである。このため各距離尺度に基づく距離算出式を一つの距離算出式に統合するために、各距離尺度に基づく2つのノード間の最小距離値及び最大距離値を用いて、式(14)に示すようにして2つのノード間の距離を正規化する。

[0108] また、式(14)では、入力ベクトルの次元数が増加するに従って、式(14)に示すノード間距離算出式の値に占めるユークリッド距離に基づく距離算出式による値の比率を減少させ、これに対応するように、他の距離尺度に基づく距離算出式による値の比率を増加させる。

[0109] 従って、このようなノード間距離算出式に示す枠組みによれば、低次元空間におけるユークリッド距離の優位性を得られる上に、高次元空間における

学習タスクに対して情報処理装置 1 を適用させることも可能となる。さらに、この枠組みは入力ベクトルの事前の分析を必要としないために、オンライン追加の自己増殖型学習タスク、あるいは限られた入力ベクトルのサンプルによる実世界の学習タスクに適している。

[数14]

$$D(p, q) = \frac{1}{\eta^d} \frac{EU_{pq} - EU_{min}}{1 + EU_{max} - EU_{min}} + \sum_{i=1}^n \frac{1}{n} \left(1 - \frac{1}{\eta^d}\right) \frac{D_i^{pq} - D_i^{min}}{1 + D_i^{max} - D_i^{min}} \dots (14)$$

[数15]

$$EU_{pq} = \|W_p - W_q\| \dots (15)$$

[0110] 後述するように、本実施の形態に係る情報処理装置 1 に対する実験においては、他の距離尺度の一例としてコサイン距離 (cosine distance) を採用し、コサイン距離をユークリッド距離に統合する。コサイン距離の定義を以下の式に示す。ここで、 θ は、重みベクトル W_p 及び重みベクトル W_q の間の角度を示す。

[数16]

$$CO_{pq} = 1 - \cos \theta = 1 - \frac{W_p \cdot W_q}{\|W_p\| \|W_q\|} \dots (16)$$

[0111] 本実施の形態では、式 (16) を用いて、式 (14) に示した距離尺度の組み合わせを、以下に示すように再定義する。情報処理装置 1 は、例えば、式 (17) に示す新たな距離尺度を用いてノード p 及びノード q の間の距離を算出し、その結果を一時記憶部に格納する。

[数17]

$$D(p, q) = \frac{1}{\eta^d} \frac{EU_{pq} - EU_{min}}{1 + EU_{max} - EU_{min}} + \left(1 - \frac{1}{\eta^d}\right) \frac{CO_{pq} - CO_{min}}{1 + CO_{max} - CO_{min}}$$

・・・(17)

[0112] <4：類似度閾値の定義>

クラス間ノード挿入判定手段25は、一時記憶部に格納されたノード及びノードの類似度閾値について、注目するノードについて、隣接ノードが存在する場合には、隣接ノードのうち注目するノードからの距離が最大であるノードまでの距離を類似度閾値として算出し、その結果を一時記憶部に格納し、隣接ノードが存在しない場合には、注目するノードからの距離が最小であるノードまでの距離を類似度閾値として算出し、その結果を一時記憶部に格納する。具体的には、例えば以下のようにして注目するノードの類似度閾値を算出し、その結果を一時記憶部に格納する。なお、クラス間ノード挿入判定手段25は、ネットワークに初めて挿入され一時記憶部に格納されたノード*i*の類似度閾値 T_i については、予め定める十分大きな値を設定し、その結果を一時記憶部に格納する。

[0113] クラス間ノード挿入判定手段25は、一時記憶部に格納されたノードについて、ノード*i*が第1勝者ノード又は第2勝者ノードとなった場合に、ノード*i*が隣接ノードを有するか否かを判定し、その結果を一時記憶部に格納する。

[0114] クラス間ノード挿入判定手段25は、一時記憶部に格納された判定の結果、ノード*i*が隣接ノードを有する場合には、一時記憶部に格納された類似度閾値及びノードについて、類似度閾値 T_i を隣接ノードへの最大距離とし、その結果を一時記憶部に格納する。即ち、ノード*i*について、一時記憶部に格納される以下の式(18)に基づいて類似度閾値 T_i を算出し、その結果を一時記憶部に格納する。ここで、*j*は一時記憶部に格納されたノード*i*の隣接ノード集合 N_i に含まれるノードであり、 $D(i, j)$ は、式(17)を用い

て算出する、ノード i 及びノード j 間の距離である。

[数18]

$$T_i = \max_{j \in N_i} D(i, j) \quad \dots (18)$$

[0115] クラス間ノード挿入判定手段 25 は、一時記憶部に格納された判定の結果、ノード i が隣接ノードを有しない場合には、ノード i からノード i を除いた他の各ノードへの距離を算出し、算出された距離のうち最小の距離を類似度閾値 T_i とする。即ち、ノード i について、一時記憶部に格納される以下の式 (19) に基づいて類似度閾値 T_i を算出し、その結果を一時記憶部に格納する。

[数19]

$$T_i = \min_{j \in N_i \setminus \{i\}} D(i, j) \quad \dots (19)$$

[0116] <5：新たなノード密度の定義及びその算出処理>

ノード密度算出手段 23 は、注目するノードのノード密度を、隣接ノードからのノードの平均距離に基づいて算出する。隣接ノード j からのノード i の平均距離のベクトル $\vec{d}_i$ は、例えば、以下の式 (20) に示すようにして算出することができる。ここで、変数 $\vec{d}_i$ は、変数 d_i の上に記号「 $\rightarrow$ (ライトアロー)」が付いていることを示す。また、以下の説明において、記号「 $\rightarrow$ (ライトアロー)」は、ベクトルであることを示す。ここで、ベクトルの要素数は、使用する距離尺度の個数と一致する。本実施の形態では、後述するように、2つの距離尺度を使用するため、ベクトルの要素数は2である。

[数20]

$$\vec{d}_i = \frac{1}{m} \sum_{j=1}^m D(i, j) \quad \dots (20)$$

- [0117] 本実施の形態では、上述したように、新たな距離尺度の組み合わせの枠組みを提案し、ユークリッド距離に加えて他の距離尺度を使用するために、2つのノード間の距離を正規化する。このため、本実施の形態では、以下に説明するように、ノード密度の算出に際して上記式(20)をそのまま採用することに代えて、隣接ノード j からのノード i の平均距離のベクトル $d_{i \rightarrow}$ を以下の式(23)を使用して算出する。また、本実施の形態では、各距離尺度の正規化のために使用されるノード間の最小距離値及び最大距離値(EU_{max} 、 EU_{min} 、 $D_{i \rightarrow}^{max}$ 、 $D_{i \rightarrow}^{min}$)はネットワークに新たな入力ベクトルが入力されると変化するため、この点についても考慮する。
- [0118] 情報処理装置1では、新たな距離尺度の枠組みを採用する。このため、新たなノード i について隣接ノードからの平均距離を式(20)に基づいて算出する場合に、以前に挿入した他のノードの平均累積ポイント値を算出する際に使用した EU_{max} 又は $D_{i \rightarrow}^{max}$ と比べて、 EU_{max} 又は $D_{i \rightarrow}^{max}$ の値が十分に大きくなっている可能性がある(ただし、ここで EU_{min} 及び $D_{i \rightarrow}^{min}$ は変化しないものと仮定する。)。言い換えると、以前に挿入した他のノードの隣接ノードからの平均距離が算出され、新しいノードの隣接ノードからの平均距離が算出されるが、それぞれ異なったスケールにおいて正規化がなされる。
- [0119] そこで、本実施の形態では、情報処理装置1は、ノード密度の新たな定義及び算出処理を導入し、上述した問題に対応する。この基本的な考えは、各距離尺度に基づくノード間の最小距離値及び最大距離値(EU_{max} 、 EU_{min} 、 $D_{i \rightarrow}^{max}$ 、 $D_{i \rightarrow}^{min}$)のうちで少なくとも1つの値が変化した場合に、ネットワークに存在する全てのノードのノード密度の累積ポイント値を再び算出して更新することである。以下、具体的に説明する。
- [0120] まず、ノード密度の新たな定義を説明する。ノード i について、その隣接ノード j からのノード i の平均距離のベクトル $d_{i \rightarrow}$ を、以下に示すように新しく定義する。

[数21]

$$\vec{d}_i = \frac{1}{m} \sum_{j=1}^m \left(\frac{1}{\eta^d} \frac{EU_{ij} - EU_{\min}}{1 + EU_{\max} - EU_{\min}}, \frac{1}{n} \left(1 - \frac{1}{\eta^d} \right) \frac{D_1^j - D_1^{\min}}{1 + D_1^{\max} - D_1^{\min}}, \right. \\ \left. \dots, \frac{1}{n} \left(1 - \frac{1}{\eta^d} \right) \frac{D_n^j - D_n^{\min}}{1 + D_n^{\max} - D_n^{\min}} \right) \\ \dots (21)$$

[0121] すると、式(21)を変形することによって、以下の式(22)を得る。

[数22]

$$\vec{d}_i = \left(\frac{1}{\eta^d} \frac{1}{m} \sum_{j=1}^m \frac{EU_{ij} - EU_{\min}}{1 + EU_{\max} - EU_{\min}}, \frac{1}{n} \left(1 - \frac{1}{\eta^d} \right) \frac{1}{m} \sum_{j=1}^m \frac{D_1^j - D_1^{\min}}{1 + D_1^{\max} - D_1^{\min}}, \right. \\ \left. \dots, \frac{1}{n} \left(1 - \frac{1}{\eta^d} \right) \frac{1}{m} \sum_{j=1}^m \frac{D_n^j - D_n^{\min}}{1 + D_n^{\max} - D_n^{\min}} \right) \\ \dots (22)$$

[0122] 最後に、式(22)の表記を簡略化することによって、隣接ノードjからのノードiの平均距離のベクトル $\vec{d}_i$ を、以下の式に示すように表すことができる。ノード密度算出手段23は、一時記憶部に格納されたノード、各距離尺度に基づくノード間の最小距離値及び最大距離値、隣接ノードからのノードの平均距離のベクトルについて、各距離尺度に基づく注目するノードの隣接ノードからの距離と、各距離尺度に基づくノード間の最小距離値及び最大距離値と、に基づいて注目するノードの隣接ノードからの平均距離のベクトルを算出し、その結果を一時記憶部に格納する。ノード密度算出手段23は、例えば、一時記憶部に格納される以下の式(23)を用いて隣接ノードjからの注目するノードiの平均距離のベクトル $\vec{d}_i$ を算出し、その結果を一時記憶部に格納する。

[数23]

$$\vec{d}_i = \left(\frac{1}{\eta^d} \overline{NEU}_i, \frac{1}{n} \left(1 - \frac{1}{\eta^d} \right) \overline{ND}_1^i, \dots, \frac{1}{n} \left(1 - \frac{1}{\eta^d} \right) \overline{ND}_n^i \right) \dots (23)$$

[0123] 上記式（23）に示した隣接ノード j からの注目するノード i の平均距離のベクトル $\vec{d}_i$ における各要素は、0 より大きくかつ 1 より小さい。このため、ノード密度のポイント値のベクトル $\vec{p}_i$ の新たな定義を、以下の式（24）に示すように定める。ノード密度算出手段 23 は、一時記憶部に格納されるノード、隣接ノードからのノードの平均距離のベクトル、ノード密度のポイント値のベクトルについて、注目するノードが第 1 勝者である場合にのみ隣接ノードからのノードの平均距離のベクトルに応じた値をノード密度のポイント値のベクトルとして算出し、その結果を一時記憶部に格納する。ノード密度算出手段 23 は、例えば、一時記憶部に格納される以下の式（24）に示す算出処理を実行し、ノード密度のポイント値のベクトル $\vec{p}_i$ を算出し、その結果を一時記憶部に格納する。

[数24]

$$\vec{p}_i = \begin{cases} \vec{1} - \vec{d}_i & \text{if node } i \text{ is winner} \\ 0 & \text{if node } i \text{ is not a winner} \end{cases} \dots (24)$$

[0124] 式（24）に示した”ノード密度のポイント値のベクトル $\vec{p}_i$ ”の定義に関して、隣接ノードからのノード i の平均距離が大きく、かつ、その領域におけるノードの個数が少ない場合、ノード密度のポイント値の定義は、E-S0INN の定義とは異なったものになる。情報処理装置 1 では、このような場合には、低い”ノード密度のポイント値のベクトル $\vec{p}_i$ ”を、ノード i に与える（低いとは、ノード密度のポイント値のベクトル $\vec{p}_i$ の要素の合計値が小さいことを意味する。）。他方で、隣接ノードからのノード i の平均距離が小さい

場合には、その領域におけるノードの個数が多く、その領域におけるノード密度が高いことを意味する。このため、情報処理装置 1 は、高い”ノード密度のポイント値のベクトル $\vec{p}_i$ ”を、ノード i に与える（高いとは、ノード密度のポイント値のベクトル $\vec{p}_i$ の要素の合計値が大きいことを意味する。）。

[0125] 情報処理装置 1 は、入力ベクトルが 1 つ入力されてそれに対する 1 回の学習の繰り返しにおいて、式 (24) に示したようにノード i が勝者ノードである場合にのみ、ノード i についての”ノード密度のポイント値のベクトル $\vec{p}_i$ ”のみを算出する。この繰り返しにおいて、勝者ノードでない他のノードの”ノード密度のポイント値のベクトル $\vec{p}_i$ ”は算出せず、その値は全て 0 である。情報処理装置 1 は、ノード密度のポイント値のベクトル $\vec{p}_i$ の定義を E-SOINN とは異なるものにし、式 (24) が線形関数であることから、各距離尺度に基づくノード間の最小距離値及び最大距離値 ($E U_{\max}$ 、 $E U_{\min}$ 、 $D_{i \max}$ 、 $D_{i \min}$) のうちで少なくとも 1 つの値が変化した場合に、ノード密度を容易に再び算出できるようにする。

[0126] ノード密度算出手段 25 は、一時記憶部に格納されたノード、ノード密度のポイント値のベクトルについて、第 1 勝者ノード i のノード密度のポイント値のベクトル $\vec{p}_i$ を、 λ 及び m に関して累積して合計することによって、第 1 勝者ノード i のノード密度の累積ポイント値のベクトル $\vec{s}_i$ を算出し、その結果を一時記憶部に格納する。ノード密度算出手段 25 は、例えば、一時記憶部に格納される以下の式 (25) に示す算出処理を実行し、その結果を一時記憶部に格納する。ここで、連続して与えられる入力ベクトルの入力総数が $L T$ となるまでの学習区間を、1 つの学習区間とする。さらに、この学習区間を、入力ベクトルの入力総数が所定の単位数 (λ) となるごとに複数の単位学習区間に分割する。この複数の学習区間の個数を m とする ($m = L T / \lambda$)。

[数25]

$$\vec{s}_i = \sum_{j=1}^m \left(\sum_{k=1}^{\lambda} \vec{p}_i \right) \dots (25)$$

[0127] そして、ノード密度算出手段25は、一時記憶部に格納されたノード、ノード密度の累積ポイント値のベクトルについて、所定の関数 f に対してノード i のノード密度の累積ポイント値のベクトル $\vec{s}_i$ を入力し、単位入力数 N によって除算して算出し、その結果をノード i のノード密度 h_i として一時記憶部に格納する。本実施の形態では、単位入力数 N あたりの平均値 h_i を、ノード i のノード密度として定義する。従って、ノード密度算出手段25は、例えば、一時記憶部に格納される以下の式(26)に示す算出処理を実行し、ノード密度を算出し、その結果を一時記憶部に格納する。ここで、 $f(\vec{X})$ は、ベクトル $\vec{X}$ の要素の合計値を算出する関数である。単位入力数 N は、 m 個の単位学習区間のうちで、ノード密度の累積ポイント値のベクトル $\vec{s}_i$ の要素が0より大きい場合の単位学習区間の回数として算出する。なお、 N の定義はE-S0INNと同一である(特許文献1の段落0067を参照。)

[数26]

$$h_i = \frac{1}{N} f(\vec{s}_i) \dots (26)$$

[0128] 次に、ノード密度の更新処理について説明する。ノード密度更新判定手段22は、一時記憶部に格納されたノード、各距離尺度に基づくノード間の最小距離値及び最大距離値について、各距離尺度に基づくノード間の最小距離値及び最大距離値(EU_{max} 、 EU_{min} 、 D_i^{max} 、 D_i^{min})のうちで少なくとも1つの値が変化したか否かを確認し、少なくとも1つの値が変化した場合にはノード密度を更新すると判定し、その結果を一時記憶部に格納する。

[0129] そして、ノード密度算出手段25は、一時記憶部に格納された判定の結果、ノード密度を更新する場合には、一時記憶部に格納されたノード、ノード密度の累積ポイント値のベクトル、ノードの学習時間、各距離尺度に基づくノード間の最小距離値及び最大距離値について、ノード密度の累積ポイント値のベクトル、ノードの学習時間、各距離尺度に基づくノード間の最小距離値及び最大距離値に基づいて、ネットワークに存在する全てのノードについ

て、ノード i のノード密度の累積ポイント値のベクトル $\vec{s}_i$ を以下に示すようにして再び算出して更新し、更新したノード i のノード密度の累積ポイント値のベクトル $\vec{s}_i$ を用いてノード i のノード密度 h_i を再び算出し、その結果を一時記憶部に格納する。ノード密度算出手段 25 は、例えば、一時記憶部に格納される以下の式 (27) ~ (30) に示す算出処理を実行することによって、ネットワークに存在するノード i のノード密度の累積ポイント値のベクトル $\vec{s}_i$ を算出し、算出したノード i のノード密度の累積ポイント値のベクトル $\vec{s}_i$ を式 (26) に代入して、ノード i のノード密度 h_i を再び算出し、その結果を一時記憶部に格納する。このようにして、ノード密度算出手段 25 は、ノード i のノード密度の累積ポイント値のベクトル $\vec{s}_i$ 及びノード i のノード密度 h_i を再び算出して更新する。

[数27]

$$\vec{s}_i = \vec{k} \cdot (\vec{s}_i - \vec{M}_i) - M_i \vec{b} + \vec{M}_i \dots (27)$$

[0130] 式 (27) において、 M_i はノード i の学習時間を示す。ノード i の学習時間のベクトル $\vec{M}_i$ を、以下の式 (28) に示す。

[数28]

$$\vec{M}_i = (M_i, M_i, \dots, M_i) \dots (28)$$

[0131] ノード密度算出手段 25 は、例えば、一時記憶部に格納される以下の式 (29) 及び式 (30) に示す算出処理を実行することによって、式 (27) におけるベクトル $\vec{k}$ 及び $\vec{b}$ を算出し、その結果を一時記憶部に格納する。ここで、 $EU'_{\max}$ 、 $EU'_{\min}$ 、 $(D_i^{\max})'$ 、 $(D_i^{\min})'$ は、各距離尺度に基づくノード間の最小距離値及び最大距離値についての、変化後の新たな値である。他方で、 $EU_{\max}$ 、 $EU_{\min}$ 、 $D_i^{\max}$ 、 $D_i^{\min}$ は、変化する以前の値である。変化する以前の値とは、学習処理過程においてネットワークに

おけるノード間の距離が変化した結果、 $EU_{\max}$ 、 $EU_{\min}$ 、 $D_i^{\max}$ 、 $D_i^{\min}$ の値が変化した場合に、変化以前の値を指し、 $EU_{\max}$ 、 $EU_{\min}$ 、 $D_i^{\max}$ 、 $D_i^{\min}$ をそのまま用いて記述する。一方で、変化後の新しい値とは、 $EU'_{\max}$ 、 $EU'_{\min}$ 、 $(D_i^{\max})'$ 、 $(D_i^{\min})'$ の値が変化した場合に、変化後の値を指し、以前の値と区別するために、 $EU'_{\max}$ 、 $EU'_{\min}$ 、 $(D_i^{\max})'$ 、 $(D_i^{\min})'$ を用いて示す。

[数29]

$$\vec{k} = \left(\frac{1 + EU_{\max} - EU_{\min}}{1 + EU'_{\max} - EU'_{\min}}, \frac{1 + D_1^{\max} - D_1^{\min}}{1 + (D_1^{\max})' - (D_1^{\min})'}, \dots, \frac{1 + D_n^{\max} - D_n^{\min}}{1 + (D_n^{\max})' - (D_n^{\min})'} \right) \quad \dots (29)$$

[数30]

$$\vec{b} = \left(\frac{1}{\eta^d} \frac{EU_{\min} - EU'_{\min}}{1 + EU'_{\max} - EU'_{\min}}, \frac{1}{n} \left(1 - \frac{1}{\eta^d} \right) \frac{D_1^{\min} - (D_1^{\min})'}{1 + (D_1^{\max})' - (D_1^{\min})'}, \dots, \frac{1}{n} \left(1 - \frac{1}{\eta^d} \right) \frac{D_n^{\min} - (D_n^{\min})'}{1 + (D_n^{\max})' - (D_n^{\min})'} \right) \quad \dots (30)$$

[0132] < 6 : 辺接続判定処理及び辺接続処理（勝者ノード接続処理） >

E-S0INNでは、勝者ノード間の辺接続判定処理及び辺接続処理（特許文献1の図5のS606、S607の処理を指す。具体的には、明細書の段落0072～0079に記載の処理を指す。）に関して、パラメータ α を使用する。このパラメータ α は、所定の閾値関数を用いて定義されている（特許文献1の段落0078を参照。）。この所定の閾値関数は、2.0、3.0、0.0、0.5、1.0などのいくつかの整数を必要としており、実際上の応用処理において良好な学習結果を得るためには、これらの整数値を調整する必要がある。しかしながら、ユーザーにとって、そのような調整タスクを行うことは容易ではない。そこで、情報処理装置1では、以下に示す滑らかな関数（soft function）を提案し、区分的関数（piecewise function）である

前記所定の閾値関数を置き換える。

[0133] 辺接続判定手段 27 は、一時記憶部に格納されたクラス間ノード挿入判定手段 25 による判定結果に応じて、一時記憶部に格納されたノード、ノードのサブクラスタラベル、ノードの密度について、第 1 勝者ノード及び第 2 勝者ノードが属するサブクラスタラベルと、第 1 勝者ノード及び第 2 勝者ノードについてのノード密度の条件とに基づいて、第 1 勝者ノード及び第 2 勝者ノード間に辺を接続するか否かを判定し、その結果を一時記憶部に格納する。

[0134] 辺接続手段 28 は、一時記憶部に格納された判定結果に応じて、第 1 勝者ノード及び第 2 勝者ノード間の辺の生成及び削除処理を実行し、その結果を一時記憶部に格納する。

[0135] 辺接続判定手段 27 は、例えば以下のようにして、辺を接続するか否かを判定し、その結果を一時記憶部に格納する。辺接続手段 28 は、例えば以下のようにして、辺の生成及び削除処理を実行し、その結果を一時記憶部に格納する。

[0136] <辺接続判定処理及び勝者ノード接続処理：S401～S405>

S401：辺接続判定手段 27 は、一時記憶部に格納されたノード、ノードのサブクラスタラベル（隆起ラベル）について、ノードのサブクラスタラベル（隆起ラベル）に基づいて、第 1 勝者ノード及び第 2 勝者ノードが属するサブクラスタをそれぞれ判定し、その結果を一時記憶部に格納する。

[0137] S402：一時記憶部に格納された判定の結果、第 1 勝者ノード及び第 2 勝者ノードの少なくとも 1 つがいずれのサブクラスタにも属していない場合、又は、第 1 勝者ノード及び第 2 勝者ノードが同一のサブクラスタに属している場合には、辺接続手段 28 は、一時記憶部に格納されたノード及びノード間の辺について、第 1 勝者ノード及び第 2 勝者ノード間に辺を生成することによりノード間を接続し、その結果を一時記憶部に格納する。

[0138] S403：一時記憶部に格納された S401 における判定の結果、第 1 勝者ノード及び第 2 勝者ノードが互いに異なるサブクラスタに属す場合には（例えば、第 1 勝者ノードがサブクラスタ A に属し、かつ、第 2 勝者ノードが

サブクラスタ B に属する場合)、辺接続判定手段 27 は、一時記憶部に格納されたノード、ノード密度について、第 1 勝者ノード及び第 2 勝者ノードのノード密度に関して、第 1 勝者ノードを含むサブクラスタの平均ノード密度を基準とする第 1 勝者ノードについてのノード密度の条件、又は、第 2 勝者ノードを含むサブクラスタの平均ノード密度を基準とする第 2 勝者ノードについてのノード密度の条件のうちの少なくとも 1 つを満たすか否かを判定し、その結果を一時記憶部に格納する。

[0139] S 4 0 4 : 一時記憶部に格納された S 4 0 3 における判定の結果、辺を接続する必要があると判定した場合には、辺接続手段 28 は、一時記憶部に格納されたノード及びノード間の辺について、第 1 勝者ノード及び第 2 勝者ノード間に辺を生成しノード間を接続する。これによって、サブクラスタ A 及び B が統合される。

[0140] S 4 0 5 : 一時記憶部に格納された S 4 0 3 における判定の結果、辺を接続する必要がないと判定した場合には、一時記憶部に格納されたノード及びノード間の辺について、第 1 勝者ノード及び第 2 勝者ノード間を辺によって接続しない。なお、ノード間が辺によって既に接続されていた場合には、辺接続手段 28 は、一時記憶部に格納されたノード及びノード間の辺について、一時記憶部に格納された第 1 勝者ノード及び第 2 勝者ノード間の辺を削除し、その結果を一時記憶部に格納する。

[0141] ここで、S 4 0 3 における辺接続判定手段 27 による判定処理について詳細に説明する。

辺接続判定手段 27 は、例えば、以下の式 (31) 又は式 (32) の少なくとも 1 つを満足する場合、第 1 勝者ノード及び第 2 勝者ノード間に辺を接続するものと判定し、そうでない場合、第 1 勝者ノード及び第 2 勝者ノード間に辺を接続しないと判定し、その結果を一時記憶部に格納する。

[数 31]

$$\min(h_{winner}, h_{secondwinner}) > (\theta^{\frac{h_A}{mean_A}} - 1) mean_A \dots (31)$$

[数32]

$$\min(h_{winner}, h_{secondwinner}) > (\theta^{\frac{h_B}{mean_B}} - 1) mean_B \dots (32)$$

[0142] 式(31)及び式(32)において、 h_{winner} は第1勝者ノードのノード密度を示し、 $h_{secondwinner}$ は第2勝者ノードのノード密度を示す。 $\min(h_{winner}, h_{secondwinner})$ は、第1勝者ノードのノード密度 h_{winner} 及び第2勝者ノードのノード密度 $h_{secondwinner}$ のうちの最小のノード密度を示す。 h_A は、サブクラスタAに含まれるノードのうちで最大のノード密度を有するノードのノード密度を示し、 $mean_A$ は、サブクラスタAに含まれる全てのノードから算出するサブクラスタAの平均ノード密度を示す。 h_B は、サブクラスタBに含まれるノードのうちで最大のノード密度を有するノードのノード密度を示し、 $mean_B$ は、サブクラスタBに含まれる全てのノードから算出するサブクラスタBの平均ノード密度を示す。 θ は、式(12)において用いたパラメータと同一のパラメータである。

[0143] 式(31)に示す第1勝者ノードについてのノード密度の条件は、第1勝者ノードのノード密度及び第2勝者ノードのノード密度のうちの最小のノード密度が、第1勝者ノードを含むサブクラスタAの平均ノード密度 $mean_A$ を基準として、サブクラスタAの平均ノード密度 $mean_A$ に対する最大ノード密度 h_A の比率に応じて算出する閾値よりも大きいかな否かを判定する条件である。また、式(32)に示す第2勝者ノードについてのノード密度の条件は、第1勝者ノードのノード密度及び第2勝者ノードのノード密度のうちの最小のノード密度が、第2勝者ノードを含むサブクラスタBの平均ノード密度 $mean_B$ を基準として、サブクラスタBの平均ノード密度 $mean_B$ に対する最大ノード密度 h_B の比率に応じて算出する閾値よりも大きいかな否かを判定する条件である。

[0144] <全体処理フロー：S501～S519>

続いて、本実施形態に係る情報処理装置における全体処理フローについて

、図22のフローチャートを用いて説明する。図22は、本実施形態に係る情報処理装置による学習処理の処理概要を示すフローチャートである。

[0145] S501：入力情報取得手段21は、2つの入力ベクトルをランダムに取得し、ノード集合Aをそれらに対応する2つのノードのみを含む集合として初期化し、その結果を一時記憶部に格納する。また、入力情報取得手段21は、辺集合 $C \subset A \times A$ を空集合として初期化し、その結果を一時記憶部に格納する。

S502：入力情報取得手段21は、新しい入力ベクトル $\varepsilon \in R^n$ を入力し、その結果を一時記憶部に格納する。

[0146] S503：ノード密度更新判定手段22は、一時記憶部に格納されたノード、各距離尺度に基づくノード間の最小距離値及び最大距離値について、各距離尺度に基づくノード間の最小距離値及び最大距離値（ EU_{max} 、 EU_{min} 、 D_i^{max} 、 D_i^{min} ）のうちで少なくとも1つの値が変化したか否かを確認し、少なくとも1つの値が変化した場合にはノード密度を更新すると判定し、その結果を一時記憶部に格納する。なお、各距離尺度に基づくノード間の最小距離値及び最大距離値（ EU_{max} 、 EU_{min} 、 D_i^{max} 、 D_i^{min} ）は、学習開始時点において初期値として0が予め設定される。

[0147] S504：ノード密度算出手段23は、一時記憶部に格納された判定の結果、ノード密度を更新する場合には、一時記憶部に格納されたノード、ノード密度の累積ポイント値のベクトル、ノードの学習時間、各距離尺度に基づくノード間の最小距離値及び最大距離値について、ノード密度の累積ポイント値のベクトル、ノードの学習時間、各距離尺度に基づくノード間の最小距離値及び最大距離値に基づいて、ノード集合Aに含まれるノード $i \in A$ のノード密度の累積ポイント値のベクトル s_i を再び算出して更新し、更新したノード i のノード密度の累積ポイント値のベクトル s_i を用いてノード i のノード密度 h_i を再び算出し、その結果を一時記憶部に格納する。ノード密度算出手段23は、例えば、一時記憶部に格納される上述した式（27）～（30）及び式（26）に示す算出処理を実行することによって、ノード i の

ノード密度の累積ポイント値のベクトル s_i 及びノード i のノード密度 h_i を再び算出して更新する。

[0148] なお、一時記憶部に格納された判定の結果、ノード密度を更新しない場合には、S504におけるノード密度算出手段23による処理は行わずに、情報処理装置1は、S505へと処理を進める。

[0149] S505：勝者ノード探索手段24は、一時記憶部に格納された入力ベクトル及びノードについて、入力ベクトル ε に最も距離が近い第1勝者ノード a_1 及び2番目に近い第2勝者ノード a_2 を探索し、その結果を一時記憶部に格納する。勝者ノード探索手段24は、探索処理として、例えば、 $a_1 = \min_{a \in A} D(\varepsilon, a)$ 、 $a_2 = \min_{a \in A \setminus \{a_1\}} D(\varepsilon, a)$ に示す処理を実行し、その結果を一時記憶部に格納する。ここで、勝者ノード探索手段24は、上述した式(17)に示した新たな距離尺度を用いて入力ベクトル ε 及びノード a の間の距離 $D(\varepsilon, a)$ を算出し、その結果を一時記憶部に格納する。

[0150] S506：クラス間ノード挿入判定手段25は、一時記憶部に格納された入力ベクトル、ノード、ノードの類似度閾値について、入力ベクトル ε と第1勝者ノード a_1 間の距離が第1勝者ノード a_1 の類似度閾値 T_{a_1} より大きいかな否か、及び、入力ベクトル ε と第2勝者ノード a_2 間の距離が第2勝者ノード a_2 の類似度閾値 T_{a_2} より大きいかな否かを判定し、その結果を一時記憶部に格納する。ここで、クラス間ノード挿入判定手段25は、上述した式(18)又は式(19)を用いて第1勝者ノード a_1 の類似度閾値 T_{a_1} 及び第2勝者ノード a_2 の類似度閾値 T_{a_2} を算出し、その結果を一時記憶部に格納する。さらに、クラス間ノード挿入判定手段25は、入力ベクトル ε と第1勝者ノード a_1 間の距離が第1勝者ノード a_1 の類似度閾値 T_{a_1} より大きい、又は、入力ベクトル ε と第2勝者ノード a_2 間の距離が第2勝者ノード a_2 の類似度閾値 T_{a_2} より大きい場合には、クラス間ノード挿入を実行すると判定し、そうでない場合には、クラス間ノード挿入を実行しないと判定し、その結果を一時記憶部に格納する。

[0151] S507：一時記憶部に格納されたS506における判定の結果、クラス

間ノード挿入を実行する場合には、クラス間ノード挿入手段26は、一時記憶部に格納された入力ベクトル及びノードについて、入力ベクトル ε はネットワークへの新たなノードとなるため、入力ベクトル ε の成分と同一の成分を重みベクトルとして有するクラス間挿入ノードを生成し、生成したクラス間挿入ノードをノード集合Aに追加し、その結果を一時記憶部に格納する。そして、情報処理装置1は、S502へと処理を進めて、再び入力ベクトルの入力を行う。

[0152] S508：一方、一時記憶部に格納されたS506における判定の結果、クラス間ノード挿入を実行しない場合には、辺接続判定手段27は、一時記憶部に格納されたノード、ノードのサブクラスタラベルについて、ノードのサブクラスタラベルに基づいて、第1勝者ノード a_1 及び第2勝者ノード a_2 が属するサブクラスタをそれぞれ判定し、その結果を一時記憶部に格納する。辺接続判定手段27は、例えば、上述したS401に示した辺接続判定処理を行う。

[0153] また、辺接続判定手段27は、一時記憶部に格納されたS508における判定の結果、第1勝者ノード及び第2勝者ノードが互いに異なるサブクラスタに属する場合には、一時記憶部に格納されたノード、ノードの密度について、第1勝者ノード a_1 及び第2勝者ノード a_2 のノード密度に関して、第1勝者ノード a_1 を含むサブクラスタの平均ノード密度を基準とする第1勝者ノードについてのノード密度の条件、又は、第2勝者ノード a_2 を含むサブクラスタの平均ノード密度を基準とする第2勝者ノードについてのノード密度の条件のうちの少なくとも1つを満たすか否かを判定し、その結果を一時記憶部に格納する。例えば、上述したS403に示した辺接続判定処理を行う。

[0154] S509：一時記憶部に格納されたS508における判定の結果、第1勝者ノード及び第2勝者ノードの少なくとも1つがいずれのサブクラスタにも属していない、又は、同一のサブクラスタに属する場合には、辺接続手段28は、一時記憶部に格納されたノード及びノード間の辺について、第1勝者ノード a_1 及び第2勝者ノード a_2 間に辺を接続し、その結果を一時記憶部に

格納する。辺接続手段28は、例えば、上述したS402に示した勝者ノード接続処理を行う。そして、辺接続手段28は、一時記憶部に格納された辺及び辺の年齢について、新しく生成された辺、及び、既にノード間に辺が生成されていた場合にはその辺について、辺の年齢を0に設定しその結果を一時記憶部に格納する。

[0155] また、辺接続手段28は、一時記憶部に格納されたS508における判定の結果、上記の第1勝者ノード及び第2勝者ノードについてのノード密度の条件のうちの少なくとも1つを満たす場合には、一時記憶部に格納されたノード、ノード間の辺について、第1勝者ノード a_1 及び第2勝者ノード a_2 間に辺を接続し、その結果を一時記憶部に格納する。辺接続手段28は、例えば、上述したS404に示した勝者ノード接続処理を行う。

[0156] 辺接続手段28は、一時記憶部に格納されたS508における判定の結果、上記の第1勝者ノード及び第2勝者ノードについてのノード密度の条件のうちの少なくとも1つを満たさない場合には、一時記憶部に格納されたノード、ノード間の辺について、第1勝者ノード及び第2勝者ノード間に辺を接続しない（第1勝者ノード及び第2勝者ノード間に辺が存在する場合には辺を削除する）、その結果を一時記憶部に格納する。辺接続手段28は、例えば、上述したS405に示した勝者ノード接続処理を行う。

[0157] S510：ノード密度算出手段23は、一時記憶部に格納されたノード、各距離尺度に基づくノード間の最小距離値及び最大距離値、隣接ノードからのノードの平均距離のベクトル、ノードのノード密度のポイント値のベクトル、ノード密度の累積ポイント値のベクトル、ノード密度について、第1勝者ノード a_1 をノード i として、各距離尺度に基づく隣接ノードからのノード i の距離と、各距離尺度に基づくノード間の最小距離値及び最大距離値と、に基づいてノード i についての隣接ノードからの平均距離のベクトル $d_{i\rightarrow}$ を算出し、当該算出した隣接ノードからの平均距離のベクトル $d_{i\rightarrow}$ に基づいて第1勝者ノード a_1 のノード密度のポイント値のベクトル $p_{i\rightarrow}$ を算出し、当該算出した第1勝者ノード a_1 のノード密度のポイント値のベクトル $p_{i\rightarrow}$ に基づ

いてノード密度の累積ポイント値のベクトル s_i を算出し、当該算出した第1勝者ノード a_1 のノード密度の累積ポイント値のベクトル s_i に基づいて第1勝者ノード a_1 のノード密度 h_i を算出し、その結果を一時記憶部に格納する。ノード密度算出手段23は、例えば、一時記憶部に格納される上述した式(24)～(26)に示す算出処理を実行することによって、ノード i のノード密度の累積ポイント値のベクトル s_i 及びノード i のノード密度 h_i を算出する。

[0158] S511：勝者ノード学習時間算出手段29は、一時記憶部に格納された第1勝者ノード a_1 の学習時間 M_{a_1} を所定の値増加し、その結果を一時記憶部に格納する。勝者ノード学習時間算出手段29は、例えば、 $M_{a_1}(t+1) = M_{a_1}(t) + 1$ という処理を実行することによって、第1勝者ノード a_1 の学習時間 M_{a_1} を1増加し、その結果を一時記憶部に格納する。

[0159] S512：重みベクトル更新手段30は、一時記憶部に格納されたノード及びノードの重みベクトルについて、第1勝者ノード a_1 及びその隣接ノードの重みベクトルをそれぞれ入力ベクトル ε に更に近づけるように更新し、その結果を一時記憶部に格納する。重みベクトル更新手段30は、例えば以下の式(33)及び式(34)を用いて、第1勝者ノード a_1 の重みベクトル W_{a_1} についての更新量 ΔW_{a_1} 及び第1勝者ノード a_1 の隣接ノード j の重みベクトル W_{s_1} についての更新量 ΔW_j を算出し、更新量 ΔW_{a_1} を第1勝者ノード a_1 の重みベクトル W_{a_1} に加算し、更新量 ΔW_j を隣接ノード j の重みベクトル W_{s_1} に加算して、この結果を一時記憶部に格納する。式(33)及び式(34)において、 M_{s_1} は、第1勝者ノード a_1 の学習時間を示す。なお、ここでの重みベクトルの変化方法はE-S0INNと同様である（特許文献1の段落0063を参照。）。

[数33]

$$\Delta W_{a_1} = \frac{1}{M_{a_1}} (\varepsilon - W_{a_1}) \dots (33)$$

[数34]

$$\Delta W_j = \frac{1}{100M_{a_1}} (\varepsilon - W_j) \dots (34)$$

[0160] S 5 1 3 : 老齡辺削除手段 3 1 は、一時記憶部に格納されたノード、ノード間の辺、辺の年齢について、第 1 勝者ノード a_1 と直接的に接続される全ての辺の年齢を所定の値増加し、その結果を一時記憶部に格納する。老齡辺削除手段 3 1 は、例えば、第 1 勝者ノード a_1 と直接的に接続される全ての辺の年齢を 1 増加し、その結果を一時記憶部に格納する。老齡辺削除手段 3 1 は、一時記憶部に格納された辺について、予め設定され一時記憶部に格納された所定の閾値 age_{max} を超えた年齢を有する辺を削除し、その結果を一時記憶部に格納する。ここで、パラメータ age_{max} は実験により予め適当な値を算出し一時記憶部に格納される。

[0161] S 5 1 4 : 入力数判定手段 3 2 は、一時記憶部に格納された与えられた入力ベクトル ε の総数について、入力された入力ベクトル ε の総数が予め設定され一時記憶部に格納された所定の単位数 (λ) の倍数であるか否かを判定し、その結果を一時記憶部に格納する。一時記憶部に格納された判定の結果、入力ベクトルの総数が所定の単位数でない場合には S 5 0 2 へと戻り、次の入力ベクトル ε を処理する。入力ベクトル ε の総数が所定の単位数となった場合には以下の S 5 1 5 以降に示す処理を実行する。なお、パラメータ λ は実験により予め適当な値を算出し一時記憶部に格納される。

[0162] なお、S 5 1 5 ~ S 5 1 7 に示す処理は、情報処理装置 1 に対して入力された入力ベクトルの総数が所定の単位数となるごとに実行される。即ち、入力ベクトルの総数が所定の数 (λ) となるまでの単位学習区間を一周期とすると、S 5 1 5 ~ S 5 1 7 に示す処理は、入力ベクトルの総数に応じた所定の周期ごとに定期的に行われる。

[0163] S 5 1 5 : 負荷平衡化手段 3 3 は、一時記憶部に格納されたノード、ノード間の辺、ノードのサブクラスタラベル、ノードの重みベクトル、ノードの

学習時間、ノードの密度について、学習時間の値が大きくかつ偏ったノードをノードの学習時間に基づいて検出し、当該検出したノードとその隣接ノードの間に新たなノードを挿入してノード集合Aに追加する。当該検出したノード及びその隣接ノードについてのノードの学習時間をそれぞれ減少し、その結果を一時記憶部に格納する。負荷平衡化手段33は、例えば、上述したS101～S107に示したノード間の負荷平衡化処理を実行し、その結果を一時記憶部に格納する。これによって、負荷平衡化手段33は、負荷が不均衡な状態にあるノードについてその負荷を平衡化させる。

[0164] S516：サブクラスタ決定手段34は、一時記憶部に格納されたノード、ノード間の辺、ノード密度、ノードのサブクラスタラベルについて、ノード集合Aに含まれるノードから局所的に最大のノード密度を有するノードを頂点として探索し、当該探索した頂点に対してそれぞれ異なるサブクラスタラベル（隆起ラベル）を付与し、当該サブクラスタラベル（隆起ラベル）が付与されなかった全てのノードに対してノード密度が最大の隣接ノードと同一のサブクラスタラベル（隆起ラベル）を付与し、その結果を一時記憶部に格納する。また、サブクラスタ決定手段34は、一時記憶部に格納されたノード、ノード間の辺、ノード密度について、頂点のうちでノード密度が所定の閾値よりも大きな頂点を基準としてボロノイ領域を生成し、その結果を一時記憶部に格納する。さらに、サブクラスタ決定手段34は、一時記憶部に格納されたノード、ノード間の辺、ノード密度、ボロノイ領域について、生成したボロノイ領域において、基準とした頂点を含むサブクラスタと基準とした頂点とは異なる他の頂点を含むサブクラスタとが重複領域を有し、かつ、当該重複領域に位置するノードの平均ノード密度の条件を満たす場合に、基準とした頂点を含むサブクラスタのサブクラスタラベルを、他の頂点を含むサブクラスタのサブクラスタラベルとして付与し、その結果を一時記憶部に格納する。サブクラスタ決定手段34は、例えば、上述したS201～S205、S301～S305に示したサブクラスタ決定処理を実行し、その結果を一時記憶部に格納する。

[0165] S 5 1 7 : ノイズノード削除手段 3 5 は、一時記憶部に格納されたノード集合 A に含まれる全てのノード a について、ノイズノードと見なしたノードを削除し、その結果を一時記憶部に格納する。ノイズノード削除手段 3 5 は、一時記憶部に格納されたノード、ノード間の辺、隣接ノードの個数、ノード密度について、例えば、以下の S 6 0 1 ~ S 6 0 4 に示す処理を実行し、注目するノード a の隣接ノードの個数及びノード密度に基づいて、注目するノードを削除し、その結果を一時記憶部に格納する。

[0166] S 6 0 1 : ノイズノード削除手段 3 5 は、一時記憶部に格納されたノード、ノード間の辺、隣接ノードの個数について、注目するノード a の隣接ノードの個数を算出し、その結果を一時記憶部に格納する。そして、ノイズノード削除手段 3 5 は、一時記憶部に格納された隣接ノードの個数に応じて、以下の S 6 0 2 ~ S 6 0 4 のいずれかの処理を実行する。なお、以下の処理において使用されるパラメータ c_1 及び c_2 は、予め適当な値が設定され一時記憶部に格納される。

[0167] S 6 0 2 : 一時記憶部に格納された隣接ノード数が 2 の場合、ノイズノード削除手段 3 5 は、一時記憶部に格納される例えば式 (3 5) が成立するか否かを判定し、その結果を一時記憶部に格納する。ここで、 h_a は、ノード a のノード密度を示し、 N_A は、ノード集合 A に含まれるノードの個数を示す。一時記憶部に格納された判定結果について式 (3 5) が成立する場合には、ノイズノード削除手段 3 5 は、一時記憶部に格納されたノードについて、該当するノード a を削除し、その結果を一時記憶部に格納する。

[数35]

$$h_a < c_1 \sum_{j=1}^{N_A} \frac{h_j}{N_A} \cdots (35)$$

[0168] S 6 0 3 : 一時記憶部に格納された隣接ノード数が 1 の場合、ノイズノード削除手段 3 5 は、ノード a のノード密度 h_a について、一時記憶部に格納される例えば式 (3 6) が成立するか否かを判定し、その結果を一時記憶部に

格納する。一時記憶部に格納された判定結果について式（36）が成立する場合には、ノイズノード削除手段35は、一時記憶部に格納されたノードについて、該当するノードaを削除し、その結果を一時記憶部に格納する。

[数36]

$$h_a < c_2 \sum_{j=1}^{N_A} \frac{h_j}{N_A} \cdots (36)$$

[0169] S604：一時記憶部に格納された隣接ノード数が0の場合（隣接ノードを有しない場合）、ノイズノード削除手段35は、一時記憶部に格納されたノードについて、該当するノードaを削除し、その結果を一時記憶部に格納する。

[0170] S518：学習終了判定手段36は、情報処理装置1による学習処理を終了するか否かを判定する。例えば、学習終了判定手段36は、一時記憶部に格納された与えられた入力ベクトル ε の総数について、与えられた入力ベクトル ε の総数が予め設定され一時記憶部に格納された所定の数LTの倍数であるか否かを判定し、その結果を一時記憶部に格納する。一時記憶部に格納された判定の結果、入力ベクトルの総数がLTの倍数でない場合にはS502へと戻り、次の入力ベクトル ε を処理する。一方、入力ベクトル ε の総数がLTの倍数となった場合には以下の処理を実行する。なお、学習終了判定の方法はこれに限定されず、ユーザーによって学習終了の指示が与えられるものとしてもよい。

[0171] S519：クラス決定手段37は、一時記憶部に格納されたノード、ノード間の辺、ノードのクラスについて、ノード間に生成された辺に基づいて、各ノードが属するクラスを決定し、その結果を一時記憶部に格納する。そして、出力情報表示手段38は、一時記憶部に格納されたノード及びノードのクラスについて、ノードの属するクラスのクラス数、及び各クラスのプロトタイプベクトルを出力する。以上の処理を終了した後、学習を停止する。

[0172] クラス決定手段37は、一時記憶部に格納されたノード、ノード間の辺、ノードのクラスについて、例えば以下のS701～S704に示す処理を実行して各ノードが属するクラスを決定し、その結果を一時記憶部に格納する。

[0173] S701：一時記憶部に格納されたノード及びノードのクラスについて、すべてのノードをどのクラスにも属していない状態にし、その結果を一時記憶部に格納する。

S702：一時記憶部に格納されたノード及びノードのクラスについて、いずれのクラスにも属していないノードからノード*i*をランダムに選択して新しいクラスラベルを付与し、その結果を一時記憶部に格納する。

S703：一時記憶部に格納されたノード、ノード間の辺、ノードのクラスについて、ノード*i*と「パス」によって接続しているノードをすべて探索し、当該探索したノードにノード*i*と同じクラスラベルを付与し、その結果を一時記憶部に格納する。

S704：一時記憶部に格納されたノード及びノードのクラスについて、いずれのクラスにも属していないノードが存在する場合にはS702へと進み、全てのノードに対してクラスラベルを付与し終えるまでの間、処理を続ける。

[0174] S703において、任意の2つのノード*a*及びノード*b*とが「パス」によって接続されるとは、ノード*a*及びノード*b*間において、いくつかの辺を通して2つのノードが接続されることを示す。即ち、ノード集合*A*に含まれるノード*a*、ノード*b*、ノード*x_i* (*i* = 1, 2, ..., *n*) に対して、ノード*a*及びノード*x₁*間の辺を示す(*a*, *x₁*)、ノード*x₁*及びノード*x₂*間の辺を示す(*x₁*, *x₂*)、...、ノード*x_n*及びノード*b*間の辺を示す(*x_n*, *b*)という辺の連続が存在する場合に、ノード*a*及びノード*b*間とがパスによって接続されると呼ぶ。

[0175] <実験結果>

続いて、本実施の形態に係る情報処理装置1の有効性を確認するため、3

つの人工データセット及び3つの実世界データセットを用いて実験を行う。情報処理装置1 (LBS0INN) をE-S0INNと比較するために、いくつかの実験では同一のデータセットを選択した。また、同一の人工データセットI (図1) を情報処理装置1 (LBS0INN) 及びE-S0INNに適用してノードの学習時間を比較し、さらに、カーネル密度推定 (KDE : kernel density estimation) を使用して、ノード間の接続構造に関するノードの密度を推定する。

[0176] 図7は、E-S0INN及び情報処理装置1 (LBS0INN) におけるノードの学習時間を示す。図において、縦軸はノードの学習時間を示し、横軸はノードの個数を示す。E-S0INNによるノードの個数は、ネットワークのいくつかの領域において十分ではなく、それらの領域におけるノードの学習時間は長くかつ偏っている。一方で、E-S0INNとは異なり、情報処理装置1 (LBS0INN) におけるノードの学習時間の分布は平衡していることが分かる。

[0177] 図8は人工データセットIのノードの密度分布を示す。図9は、E-S0INNにおけるノードの密度の推定結果を示し、図10は、情報処理装置1 (LBS0INN) におけるノードの密度の推定結果を示す。図8、9、10において、x軸及びy軸はノードが配置される2次元空間を示し、z軸はノード密度を示す。明らかに、E-S0INNは、入力パターンに対する良好な精度を得ていないことが分かる。しかし、E-S0INNでは、ノード密度を使用してクラス間の重複領域を探索するために、入力パターンの接続構造を正確に表現することはとても重要である。これに対して、図10に示したように、情報処理装置1 (LBS0INN) では、入力パターンの接続構造を正確に記述することができるため、E-S0INNに比べてより優れた安定性能を有する。

[0178] 図11に、人工データセットIIを示す。人工データセットIIでは、ノードの密度に関して高密度の重複領域を有する3つのガウス分布を含む。人工データセットIIを使用して、定常環境におけるE-S0INN及び情報処理装置1 (LBS0INN) の安定性についての実験を行う。実験では、データセットから入力パターンをランダムに選択して、入力パターンに対して10%のノイズを追加する。パラメータの設定に関して、E-S0INNでは、 $\lambda = 200$ 、age

$_{max} = 50$ 、 $c_1 = 0.001$ 、 $c_2 = 1.0$ に設定する。情報処理装置1 (LBS0INN) では、 λ 、 age_{max} 、 c_1 、 c_2 についてはE-S0INNと同じ値を設定し、 $\gamma = 1.3$ 、 $\eta = 1.001$ 、 $\theta = 1.3$ に設定する。

[0179] 図12(A)～(J)は、定常環境において時系列にネットワークを学習したときの、人工データセット11に対する学習結果を示す。図12(A)～(E)は、100000回、200000回、300000回、400000回、500000回の繰り返しにおけるE-S0INNの学習結果をそれぞれ示す。図12(F)～(J)は、100000回、200000回、300000回、400000回、500000回の繰り返しにおける情報処理装置1 (LBS0INN) の学習結果をそれぞれ示す。明らかに、E-S0INNは、異なるクラスが常に一緒に接続されてしまうために、ノードの接続構造が乱れておりかつ不安定である。

[0180] 情報処理装置1 (LBS0INN) は、上述したサブクラスタ決定手法を使用して、ネットワークに存在する過度に局所的に最大となるノードの発生を回避する。これによって、情報処理装置1 (LBS0INN) は、混合クラスをサブクラスタに分割してしまうことによるノードの接続構造の乱れを回避する。人工データセット11に対する実験は、E-S0INNと比較して情報処理装置1 (LBS0INN) がより優れた安定性を有し、かつ、ノード密度が高密度の重複領域を正確に分離できることを示している。後述するように、AT&T画像データセットを使用して、非定常的な学習環境下において、実世界のデータセットによってまた、E-S0INNと比較して情報処理装置1 (LBS0INN) がより優れた安定性を有することを確認する。

[0181] 図13に、E-S0INNの評価において使用された、他の人工データセットを示す。この人工データセットは、2つのガウス分布、2つのリング形状分布、正弦関数分布を含んでいる。入力分布は、入力サンプル数の合計の10%に達するランダムノイズを含んでいる。パラメータの設定に関して、情報処理装置1 (LBS0INN) は、 $\lambda = 100$ 、 $age_{max} = 100$ 、 $c_1 = 0.001$ 、 $c_2 = 1.0$ 、 $\gamma = 1.3$ 、 $\eta = 1.001$ 、 $\theta = 1.3$ に設定する。図14

及び図 15 は、定常環境下及び非定常環境下での情報処理装置 1 (LBS0INN) によるクラスタリング結果をそれぞれ描写する。図 14 及び図 15 の類似性から、情報処理装置 1 (LBS0INN) は入力データの入力順序に依存していないこと、即ち、E-S0INNと比較してよりも優れた安定性を有することを示している。

[0182] 実世界での実験の一つとして、情報処理装置 1 (LBS0INN) をE-S0INNと比較するために、E-S0INNにおいて使用されたデータセットと同一のA T & T画像データセットを選択する。この画像データセットは、40の異なるクラスと、クラスごとの10の異なる画像を含んでいる。これらのクラスは異なる年齢及び人種に関するものである。いくつかのクラスに対して、異なるタイミングで画像が取得されている。顔の表現（開いた／閉じた目、微笑む／微笑んでいない）及び顔の詳細（メガネあり／メガネなし）にはバリエーションがある。全ての画像は、クラスを有する黒色の均一な背景の下で、直立して正面の位置（ある程度の側面の動きに対する耐性を有する。）から撮られている。このデータセットから10クラスを選択し（図16）、最近傍補間法（nearest-neighbor interpolation）を使用して、データセットにおける画像を 92×112 から 23×28 へとサンプリングし直す。そして、ガウス法（Gaussian method）を使用することによって、 $Width = 4$ 、 $\sigma = 2$ であるガウシアンを用いて画像を平滑化して、特徴ベクトルを得る（図17）。

[0183] 定常及び非定常環境下におけるE-S0INN及び情報処理装置 1 (LBS0INN) の性能を実験する。LBS0INNのパラメータは、 $\lambda = 25$ 、 $age_{max} = 25$ 、 $c_1 = 0.0$ 、 $c_2 = 1.0$ 、 $\gamma = 1.3$ 、 $\eta = 1.001$ 、 $\theta = 1.3$ に設定する。学習を経て、ネットワークからのプロトタイプベクトルを得ることができる。これらのプロトタイプベクトルを使用して、元々の顔画像及び算出される認識率を分類する。情報処理装置 1 (LBS0INN) は、定常環境下では、96.3%の認識率であり、非定常環境下では、96.5%の認識率である。E-S0INNは、定常環境下では、90.3%の認識率であり、非定常環境下では

、96%の認識率である。即ち、E-S0INNと比較して、情報処理装置1（LBS0INN）は、E-S0INNよりも十分に高い認識率を有している。定常環境及び非定常環境における共通性はまた、情報処理装置1（LBS0INN）が入力データの入力順序に依存せず、E-S0INNと比較してより優れた安定性を有することを示している。

[0184] E-S0INN及び情報処理装置1（LBS0INN）の安定性を比較するため、E-S0INNと同一の処理を情報処理装置1（LBS0INN）によって実行する。即ち、（1）E-S0INN及び情報処理装置1（LBS0INN）の両方に対して1000回の学習を実行し、（2）クラスの個数の頻度を記録する。なお、クラスの個数の頻度とは、例えば、1000回の実験を行い、情報処理装置1（LBS0INN）の学習結果としてクラスの個数を10とした出力回数が600回である、又は、クラスの個数を9とした出力回数が90回であるときに、この600回や90回の数値を示す。

[0185] 図18（A）、（B）は、E-S0INNおよび情報処理装置1（LBS0INN）によるクラスの個数の分布をそれぞれ示す。図において、縦軸はクラスの個数の頻度を示し、横軸はクラスの個数を示す。情報処理装置1（LBS0INN）に対するクラスの個数の分布はE-S0INNのものと比べて十分に大きい。そして、クラスの個数が10個となる辺りにおいて、情報処理装置1（LBS0INN）の分布はE-S0INNのものと比べて十分に大きい。さらに、定常環境下における情報処理装置1（LBS0INN）に対するクラスの個数の分布は、非定常環境下におけるクラスの分布にとっても類似している。上述した全ての実験結果は、E-S0INNと比較して情報処理装置1（LBS0INN）がより優れた安定性を有することを示している。

[0186] 次に、2つのテキストデータセット（WebKbデータセット及びR8データセット）を選択して、高次元空間におけるE-S0INN及び情報処理装置1（LBS0INN）の性能を実験する。WebKbは、1997年1月における様々な大学のコンピュータサイエンス部門から、CMUテキスト学習グループのワールドワイド知識ベースプロジェクトによって収集されたWWWページを含

んでいる。その8282ページは、生徒、学部、スタッフ、部門、コース、プロジェクトやその他のカテゴリに、人手によって分類されている。本実験では、生徒、学部及びコースのカテゴリのみを選択する。R8データセットは、1つのトピック及びそのクラス（少なくとも1つの学習及び1つの実験例を有する。）を備えた文書を単に考慮する。WebKb及びR8の両方ともに、フィルタリング処理（非特許文献：Khaled M. Hammouda and Mohamed S. Kamel, "Efficient Phrase-Based Document Indexing for Web Document Clustering," IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 10, pp. 1279-1296, 2004.を参照。）と、stop wordsの削除(removing)、刈り取り (pruning) 及び取り出し (stemming) 処理（非特許文献：R. Baeza-Yates and B. Ribeiro-Neto, "Modern Information Retrieval," Addison-Wesley, 1999.を参照。）を事前に行う。

[0187] 図19は、異なる各クラスに対して異なるパーセントを入力したときの精度を示す。パラメータの設定に関して、情報処理装置1 (LBS0INN) は、 $\lambda = 25$ 、 $age_{max} = 25$ 、 $c_1 = 0.0$ 、 $c_2 = 1.0$ 、 $\gamma = 1.3$ 、 $\eta = 1.001$ 、 $\theta = 1.3$ に設定する。また、比較のために、線形関数を使用したサポートベクトルマシーン (SVM) の性能についても示す。

[0188] WebKbデータセットに対して、情報処理装置1 (LBS0INN) 及びSVMの精度は、それぞれ81.21%及び84.21%に達する。そして、R8データセットに対しては、情報処理装置1 (LBS0INN) 及びSVMの精度は、89.71%及び93.34%に達する。情報処理装置1 (LBS0INN) の精度は、SVMの精度ほどは良くない。しかしながら、情報処理装置1 (LBS0INN) は、オンライン教師なし学習のアプローチに属する。また違いとして、情報処理装置1 (LBS0INN) は、限られたデータセットを用いて満足できる精度を得ることができる。この特徴は、情報処理装置1 (LBS0INN) を、実世界におけるタスクにより適したものにする。E-S0INNは高次元タスクを扱うことができないために、E-S0INNの性能はとても悪い。この実験は、高次元空間において情報処理装置1 (LBS0INN) が十分に実行可能であることを示している。

[0189] 以上に、本実施の形態に係る情報処理装置 1 の構成及び動作、さらに、情報処理装置 1 に対して実験を行い、その効果を説明した。クラスタリングは最も重要なデータ分析手法の一つである。本実施の形態では、オンラインでの教師なし学習に属する負荷平衡型自己増殖型ニューラルネットワーク (LBSOINN) を提案し、これは、高密度の重複クラスを分離することができる。情報処理装置 1 (LBSOINN) は、E-SOINNと比較して優れた安定性能を有しており、かつ、E-SOINNが有する全ての利点をも備えている。人口データセット及び実データセットにおける実験により、情報処理装置 1 (LBSOINN) は、ラベル付けされていないノイズ付き知識をクラスタリングする場合に正確な学習結果を得ることができ、また、高次元空間においても効率的に実行可能なことを確認した。このことは、実世界でのオンライン学習タスクに適していることを示している。

[0190] なお、本発明は上記実施の形態に限られたものではなく、趣旨を逸脱しない範囲で適宜変更することが可能である。例えば、上述した実施の形態では、ノードの負荷平衡化処理において、各サブクラスタにおいて最大のノードの学習時間を有するノードのみを探索する場合を例に説明したが、本発明はこれに限定されず、各サブクラスタにおいて相対的に大きなノードの学習時間を有する複数のノードを探索し、当該探索した複数のノードに関して、以降の負荷平衡化処理を実行するものとしてもよい。

[0191] また、例えば、上述した実施の形態では、ノードの負荷平衡化処理において、最大のノードの学習時間を有するノードを探索し、当該探索したノードの隣接ノードのうちで最大の学習時間を有する隣接ノードを決定する場合を例に説明したが、本発明はこれに限定されず、他の隣接ノード（例えば、当該探索したノードの隣接ノードのうちで 2 番目に大きな学習時間を有する隣接ノード）を決定し、当該決定した隣接ノードに関して、以降の負荷平衡化処理を実行するものとしてもよい。

[0192] また、例えば、上述した実施の形態では、距離尺度に関して、ユークリッド距離にコサイン距離を組合せる場合を例に説明したが、本発明はこれに限

定されず、他の距離尺度（例えば、コサイン距離、マンハッタン距離、フラクショナル距離）を組合せるものとしてもよい。さらに、高次元空間における有効な距離尺度に限定されず、学習しようとする問題に応じた他の距離尺度を組合せるものとしてもよい。

[0193] 上記で説明した情報処理方法は、ASIC (Application Specific Integrated Circuit) を含む半導体処理装置を用いて実現されてもよい。また、これらの処理は、少なくとも1つのプロセッサ (e.g. マイクロプロセッサ、MPU、DSP (Digital Signal Processor)) を含むコンピュータシステムにプログラムを実行させることによって実現されてもよい。具体的には、これらの送信信号処理又は受信信号処理に関するアルゴリズムをコンピュータシステムに行わせるための命令群を含む1又は複数のプログラムを作成し、当該プログラムをコンピュータに供給すればよい。

[0194] これらのプログラムは、様々なタイプの非一時的なコンピュータ可読媒体 (non-transitory computer readable medium) を用いて格納され、コンピュータに供給することができる。非一時的なコンピュータ可読媒体は、様々なタイプの実体のある記録媒体 (tangible storage medium) を含む。非一時的なコンピュータ可読媒体の例は、磁気記録媒体（例えばフレキシブルディスク、磁気テープ、ハードディスクドライブ）、光磁気記録媒体（例えば光磁気ディスク）、CD-ROM (Read Only Memory)、CD-R、CD-R/W、半導体メモリ（例えば、マスクROM、PROM (Programmable ROM)、EPROM (Erasable PROM)、フラッシュROM、RAM (random access memory)）を含む。また、プログラムは、様々なタイプの一時的なコンピュータ可読媒体 (transitory computer readable medium) によってコンピュータに供給されてもよい。一時的なコンピュータ可読媒体の例は、電気信号、光信号、及び電磁波を含む。一時的なコンピュータ可読媒体は、電線及び光ファイバ等の有線通信路、又は無線通信路を介して、プログラムをコンピュータに供給できる。

[0195] この出願は、2013年2月22日に提出された日本出願特願2013-

0 3 3 1 7 6を基礎とする優先権を主張し、その開示の全てをここに取り込む。

符号の説明

[0196]	1	情報処理装置
	1 0	コンピュータ
	1 1	C P U
	1 2	R O M
	1 3	R A M
	1 4	バス
	1 5	入出力インターフェイス
	1 6	入力部
	1 7	出力部
	1 8	記憶部
	1 9	通信部
	2 0	ドライブ
	2 0 1	磁気ディスク
	2 0 2	光ディスク
	2 0 3	フレキシブルディスク
	2 0 4	半導体メモリ
	2 1	入力情報取得手段
	2 2	ノード密度更新判定手段
	2 3	ノード密度算出手段
	2 4	勝者ノード探索手段
	2 5	クラス間ノード挿入判定手段
	2 6	クラス間ノード挿入手段
	2 7	辺接続判定手段
	2 8	辺接続手段
	2 9	勝者ノード学習時間算出手段

- 3 0 重みベクトル更新手段
- 3 1 老齡辺削除手段
- 3 2 入力数判定手段
- 3 3 負荷平衡化手段
- 3 4 サブクラスタ決定手段
- 3 5 ノイズノード削除手段
- 3 6 学習終了判定手段
- 3 7 クラス決定手段
- 3 8 出力情報表示手段

請求の範囲

- [請求項1] 多次元ベクトルで記述されるノード及び当該ノード間を接続する辺が配置されるネットワーク構造を有し、
- 任意のクラスに属する入力ベクトルを順次入力して、当該入力ベクトルの入力分布構造を学習する情報処理装置において、
- 入力される前記入力ベクトルに最も近い距離に位置するノード及び2番目に近い距離に位置するノードを第1勝者ノード及び第2勝者ノードとしてそれぞれ探索する勝者ノード探索手段と、
- 前記入力ベクトルと前記第1勝者ノード間の距離及び前記入力ベクトルと前記第2勝者ノード間の距離に基づいて、クラス間ノード挿入を実行するか否かを判定するクラス間ノード挿入判定手段と、
- 前記クラス間ノード挿入判定手段による判定の結果、前記クラス間ノード挿入を実行する場合に、前記入力ベクトルの成分と同一の成分を重みベクトルとして有するクラス間挿入ノードを生成し、当該生成したクラス間挿入ノードを前記ネットワークに挿入するクラス間ノード挿入手段と、
- 前記クラス間ノード挿入判定手段による判定の結果、前記クラス間ノード挿入を実行しない場合に、前記第1勝者ノード及び前記第2勝者ノード間に辺を接続したときに、前記第1勝者ノードについてノードの学習時間を所定の値増加する勝者ノード学習時間算出手段と、
- 前記クラス間ノード挿入判定手段による判定の結果、前記クラス間ノード挿入を実行しない場合に、注目するノードについて当該注目するノードと辺によって直接的に接続されるノードを隣接ノードとして、前記第1勝者ノードの重みベクトル及び前記第1勝者ノードの隣接ノードの重みベクトルをそれぞれ前記入力ベクトルに更に近づけるように更新する重みベクトル更新手段と、
- 前記入力ベクトルの総数に応じた所定の周期ごとに、前記ノードの学習時間の値が相対的に大きくかつ偏っているノードを検出し、当該

検出したノード及びその隣接ノードの重みベクトルに基づく重みベクトルを有する新たなノードを生成して前記ネットワークに挿入し、前記検出したノードの学習時間及び前記検出したノードの隣接ノードの学習時間をそれぞれ減少し、前記検出したノード及びその隣接ノード間の辺を削除し、前記新たに挿入したノード及び前記検出したノード間と、前記新たに挿入したノード及び前記検出したノードのその隣接ノード間と、に辺をそれぞれ接続する負荷平衡化手段と、を備える、ことを特徴とする情報処理装置。

[請求項2]

前記クラス間ノード挿入判定手段による判定の結果、前記クラス間ノード挿入を実行しない場合に、付与されるノードのサブクラスタレベルに基づいて、前記第1勝者ノード及び前記第2勝者ノードが属するサブクラスタをそれぞれ判定し、当該判定の結果、前記第1勝者ノード及び前記第2勝者ノードが互いに異なるサブクラスタに属する場合には、前記第1勝者ノード及び前記第2勝者ノードのノード密度に関して、前記第1勝者ノードを含むサブクラスタの平均ノード密度を基準とする前記第1勝者ノードについてのノード密度の条件、又は、前記第2勝者ノードを含むサブクラスタの平均ノード密度を基準とする前記第2勝者ノードについてのノード密度の条件のうちの少なくとも1つを満たすか否かを更に判定する辺接続判定手段と、

(1) 前記辺接続判定手段による判定の結果、前記第1勝者ノード及び前記第2勝者ノードの少なくとも1つがいずれのサブクラスタに属していない、又は、同一のサブクラスタに属する場合には、前記第1勝者ノード及び前記第2勝者ノード間に辺を接続し、若しくは、(2) 前記辺接続判定手段による判定の結果、前記第1勝者ノード及び前記第2勝者ノードについてのノード密度の条件のうちの少なくとも1つを満たす場合には、前記第1勝者ノード及び前記第2勝者ノード間に辺を接続し、前記第1勝者ノード及び前記第2勝者ノードについてのノード密度の条件のうちの少なくとも1つを満たさない場合には

、前記第1勝者ノード及び前記第2勝者ノード間に辺を接続しない辺接続手段と、

注目するノード及び当該注目するノードの隣接ノードからの平均距離に基づいて、当該注目するノードのノード密度を算出するノード密度算出手段と、

局所的に最大のノード密度を有するノードを頂点として、当該頂点に対してそれぞれ異なるサブクラスタラベルを付与し、当該サブクラスタラベルが付与されなかった全てのノードに対してノード密度が最大の隣接ノードと同一のサブクラスタラベルを付与し、前記頂点のうちでノード密度が所定の閾値よりも大きな頂点を基準としてボロノイ領域を生成し、当該生成したボロノイ領域において、前記基準とした頂点を含むサブクラスタと前記基準とした頂点とは異なる他の頂点を含むサブクラスタとが重複領域を有し、かつ、当該重複領域に位置するノードの平均ノード密度の条件を満たす場合に、前記基準とした頂点を含むサブクラスタのサブクラスタラベルを、前記他の頂点を含むサブクラスタのサブクラスタラベルとして付与するサブクラスタ決定手段と、を更に備える、

請求項1に記載の情報処理装置。

[請求項3]

ユークリッド距離に基づいてノード間の距離を算出する重み付きの第1の距離算出式と、前記ユークリッド距離とは異なる他の距離尺度に基づいてノード間の距離を算出する重み付きの第2の距離算出式と、を含むノード間距離算出式を更に備え、

前記ネットワークにおける任意の2つのノード間の前記ユークリッド距離に基づく最小距離値及び最大距離値を用いて前記第1の距離算出式を正規化し、

前記ネットワークにおける任意の2つのノード間の前記他の距離尺度に基づく最小距離値及び最大距離値を用いて前記第2の距離算出式を正規化し、

前記入力ベクトルの次元数が増加するに従って、前記第 1 の距離算出式の重みを減少させ、

前記ノード間距離算出式を用いて、前記入力ベクトル及び前記ノード間の距離、又は、前記ネットワークにおける任意の 2 つのノード間の距離を算出する、

請求項 1 又は 2 に記載の情報処理装置。

[請求項 4] 前記第 2 の距離算出式は、コサイン距離に基づいてノード間の距離を算出する、

請求項 3 に記載の情報処理装置。

[請求項 5] 前記ユークリッド距離に基づく前記 2 つのノード間の最小距離値及び最大距離値と、前記他の距離尺度に基づく前記 2 つのノード間の最小距離値及び最大距離値と、のうちで少なくとも 1 つの値が変化したか否かを確認し、少なくとも 1 つの値が変化した場合にはノード密度を更新すると判定するノード密度更新判定手段と、を更に備え、

前記ノード密度算出手段は、

(1) 前記第 1 の距離算出式及び前記第 2 の距離算出式をそれぞれ用いて前記第 1 勝者ノードの隣接ノードからの平均距離のベクトルの各成分を算出し、(2) 当該算出した前記第 1 勝者ノードの前記隣接ノードからの平均距離のベクトルに基づいて前記第 1 勝者ノードのノード密度のポイント値のベクトルを算出し、(3) 当該算出した前記第 1 勝者ノードの前記ノード密度のポイント値のベクトルに基づいて前記第 1 勝者ノードのノード密度の累積ポイント値のベクトルを算出し、(4) 当該算出した前記第 1 勝者ノードの前記ノード密度の累積ポイント値のベクトルに基づいて前記第 1 勝者ノードのノード密度を算出し、(5) 前記ノード密度更新判定手段による判定の結果、ノード密度を更新する場合には、変化前後における前記ユークリッド距離に基づく前記 2 つのノード間の最小距離値及び最大距離値と、変化前後における前記他の距離尺度に基づく前記 2 つのノード間の最小距離値

及び最大距離値と、に基づいて、前記ネットワークに存在する全てのノードについて、注目するノードの前記ノード密度の累積ポイント値のベクトルを更新し、（６）当該更新した前記注目するノードの前記ノード密度の累積ポイント値のベクトルに基づいて前記注目するノードの前記ノード密度を再び算出する、
請求項３又は４に記載の情報処理装置。

[請求項６] 多次元ベクトルで記述されるノード及び当該ノード間を接続する辺が配置されるネットワーク構造を有し、

 任意のクラスに属する入力ベクトルを順次入力して、当該入力ベクトルの入力分布構造を学習する情報処理装置における情報処理方法であって、

 勝者ノード探索手段が、入力される前記入力ベクトルに最も近い距離に位置するノード及び２番目に近い距離に位置するノードを第１勝者ノード及び第２勝者ノードとしてそれぞれ探索するステップと、

 クラス間ノード挿入判定手段が、前記入力ベクトルと前記第１勝者ノード間の距離及び前記入力ベクトルと前記第２勝者ノード間の距離に基づいて、クラス間ノード挿入を実行するか否かを判定するステップと、

 クラス間ノード挿入手段が、前記クラス間ノード挿入判定手段による判定の結果、前記クラス間ノード挿入を実行する場合に、前記入力ベクトルの成分と同一の成分を重みベクトルとして有するクラス間挿入ノードを生成し、当該生成したクラス間挿入ノードを前記ネットワークに挿入するステップと、

 勝者ノード学習時間算出手段が、前記クラス間ノード挿入判定手段による判定の結果、前記クラス間ノード挿入を実行しない場合に、前記第１勝者ノード及び前記第２勝者ノード間に辺を接続したときに、前記第１勝者ノードについてノードの学習時間を所定の値増加するステップと、

重みベクトル更新手段が、前記クラス間ノード挿入判定手段による判定の結果、前記クラス間ノード挿入を実行しない場合に、注目するノードについて当該注目するノードと辺によって直接的に接続されるノードを隣接ノードとして、前記第1勝者ノードの重みベクトル及び前記第1勝者ノードの隣接ノードの重みベクトルをそれぞれ前記入力ベクトルに更に近づけるように更新するステップと、

負荷平衡化手段が、前記入力ベクトルの総数に応じた所定の周期ごとに、前記ノードの学習時間の値が相対的に大きくかつ偏っているノードを検出し、当該検出したノード及びその隣接ノードの重みベクトルに基づく重みベクトルを有する新たなノードを生成して前記ネットワークに挿入し、前記検出したノードの学習時間及び前記検出したノードの隣接ノードの学習時間をそれぞれ減少し、前記検出したノード及びその隣接ノード間の辺を削除し、前記新たに挿入したノード及び前記検出したノード間と、前記新たに挿入したノード及び前記検出したノードのその隣接ノード間と、に辺をそれぞれ接続するステップと、
、を実行する、
ことを特徴とする情報処理方法。

[請求項7] 辺接続判定手段が、前記クラス間ノード挿入判定手段による判定の結果、前記クラス間ノード挿入を実行しない場合に、付与されるノードのサブクラスタラベルに基づいて、前記第1勝者ノード及び前記第2勝者ノードが属するサブクラスタをそれぞれ判定し、当該判定の結果、前記第1勝者ノード及び前記第2勝者ノードが互いに異なるサブクラスタに属する場合には、前記第1勝者ノード及び前記第2勝者ノードのノード密度に関して、前記第1勝者ノードを含むサブクラスタの平均ノード密度を基準とする前記第1勝者ノードについてのノード密度の条件、又は、前記第2勝者ノードを含むサブクラスタの平均ノード密度を基準とする前記第2勝者ノードについてのノード密度の条件のうちの少なくとも1つを満たすか否かを更に判定するステップと

、

辺接続手段が、（１）前記辺接続判定手段による判定の結果、前記第１勝者ノード及び前記第２勝者ノードの少なくとも１つがいずれのサブクラスタに属していない、又は、同一のサブクラスタに属する場合には、前記第１勝者ノード及び前記第２勝者ノード間に辺を接続し、若しくは、（２）前記辺接続判定手段による判定の結果、前記第１勝者ノード及び前記第２勝者ノードについてのノード密度の条件のうちの少なくとも１つを満たす場合には、前記第１勝者ノード及び前記第２勝者ノード間に辺を接続し、前記第１勝者ノード及び前記第２勝者ノードについてのノード密度の条件のうちの少なくとも１つを満たさない場合には、前記第１勝者ノード及び前記第２勝者ノード間に辺を接続しないステップと、

ノード密度算出手段が、注目するノード及び当該注目するノードの隣接ノードからの平均距離に基づいて、当該注目するノードのノード密度を算出するステップと、

サブクラスタ決定手段が、局所的に最大のノード密度を有するノードを頂点として、当該頂点に対してそれぞれ異なるサブクラスタラベルを付与し、当該サブクラスタラベルが付与されなかった全てのノードに対してノード密度が最大の隣接ノードと同一のサブクラスタラベルを付与し、前記頂点のうちでノード密度が所定の閾値よりも大きな頂点を基準としてボロノイ領域を生成し、当該生成したボロノイ領域において、前記基準とした頂点を含むサブクラスタと前記基準とした頂点とは異なる他の頂点を含むサブクラスタとが重複領域を有し、かつ、当該重複領域に位置するノードの平均ノード密度の条件を満たす場合に、前記基準とした頂点を含むサブクラスタのサブクラスタラベルを、前記他の頂点を含むサブクラスタのサブクラスタラベルとして付与するステップと、を更に実行する、
請求項６に記載の情報処理方法。

[請求項8] 前記情報処理装置は、ユークリッド距離に基づいてノード間の距離を算出する重み付きの第1の距離算出式と、前記ユークリッド距離とは異なる他の距離尺度に基づいてノード間の距離を算出する重み付きの第2の距離算出式と、を含むノード間距離算出式を更に備え、

前記ネットワークにおける任意の2つのノード間の前記ユークリッド距離に基づく最小距離値及び最大距離値を用いて前記第1の距離算出式を正規化し、

前記ネットワークにおける任意の2つのノード間の前記他の距離尺度に基づく最小距離値及び最大距離値を用いて前記第2の距離算出式を正規化し、

前記入力ベクトルの次元数が増加するに従って、前記第1の距離算出式の重みを減少させ、

前記ノード間距離算出式を用いて、前記入力ベクトル及び前記ノード間の距離、又は、前記ネットワークにおける任意の2つのノード間の距離を算出する、

請求項6又は7に記載の情報処理方法。

[請求項9] 前記第2の距離算出式は、コサイン距離に基づいてノード間の距離を算出する、

請求項8に記載の情報処理方法。

[請求項10] ノード密度更新判定手段が、前記ユークリッド距離に基づく前記2つのノード間の最小距離値及び最大距離値と、前記他の距離尺度に基づく前記2つのノード間の最小距離値及び最大距離値と、のうちで少なくとも1つの値が変化したか否かを確認し、少なくとも1つの値が変化した場合にはノード密度を更新すると判定するステップを更に実行し、

前記ノード密度算出手段は、

(1) 前記第1の距離算出式及び前記第2の距離算出式をそれぞれ用いて前記第1勝者ノードの隣接ノードからの平均距離のベクトルの各

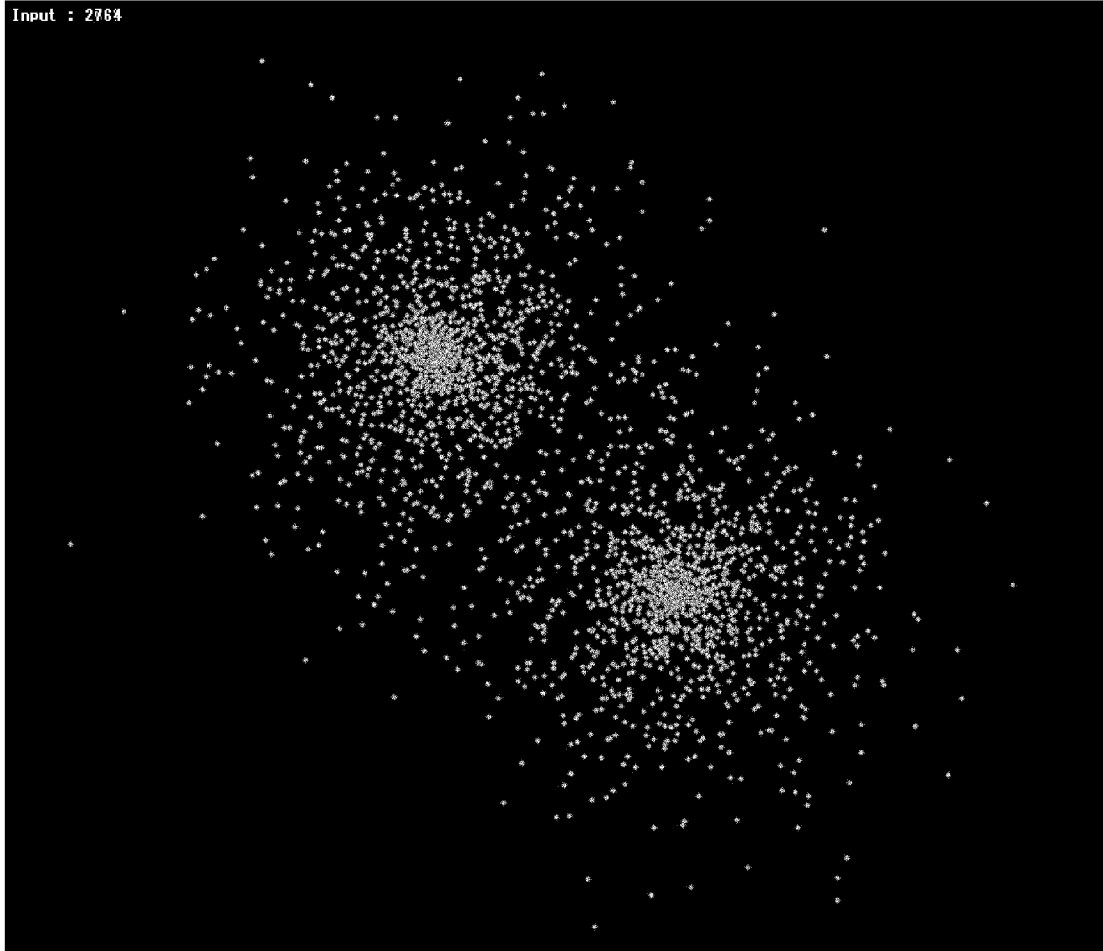
成分を算出し、（２）当該算出した前記第１勝者ノードの前記隣接ノードからの平均距離のベクトルに基づいて前記第１勝者ノードのノード密度のポイント値のベクトルを算出し、（３）当該算出した前記第１勝者ノードの前記ノード密度のポイント値のベクトルに基づいて前記第１勝者ノードのノード密度の累積ポイント値のベクトルを算出し、（４）当該算出した前記第１勝者ノードの前記ノード密度の累積ポイント値のベクトルに基づいて前記第１勝者ノードのノード密度を算出し、（５）前記ノード密度更新判定手段による判定の結果、ノード密度を更新する場合には、変化前後における前記ユークリッド距離に基づく前記２つのノード間の最小距離値及び最大距離値と、変化前後における前記他の距離尺度に基づく前記２つのノード間の最小距離値及び最大距離値と、に基づいて、前記ネットワークに存在する全てのノードについて、注目するノードの前記ノード密度の累積ポイント値のベクトルを更新し、（６）当該更新した前記注目するノードの前記ノード密度の累積ポイント値のベクトルに基づいて前記注目するノードの前記ノード密度を再び算出する、

請求項８又は９に記載の情報処理方法。

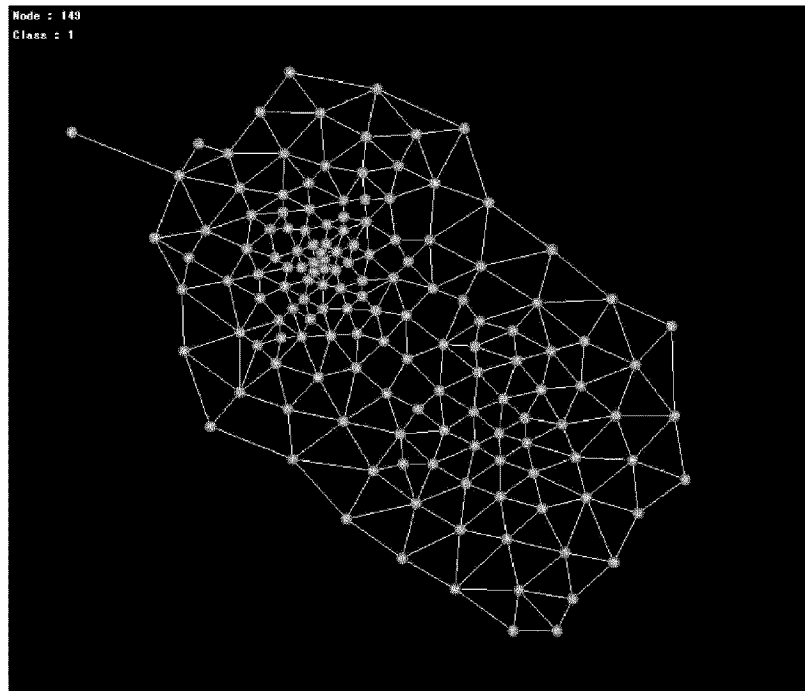
[請求項11] 請求項６乃至１０いずれか１項に記載の情報処理をコンピュータに実行させることを特徴とする非一時的なコンピュータ可読媒体。

[図1]

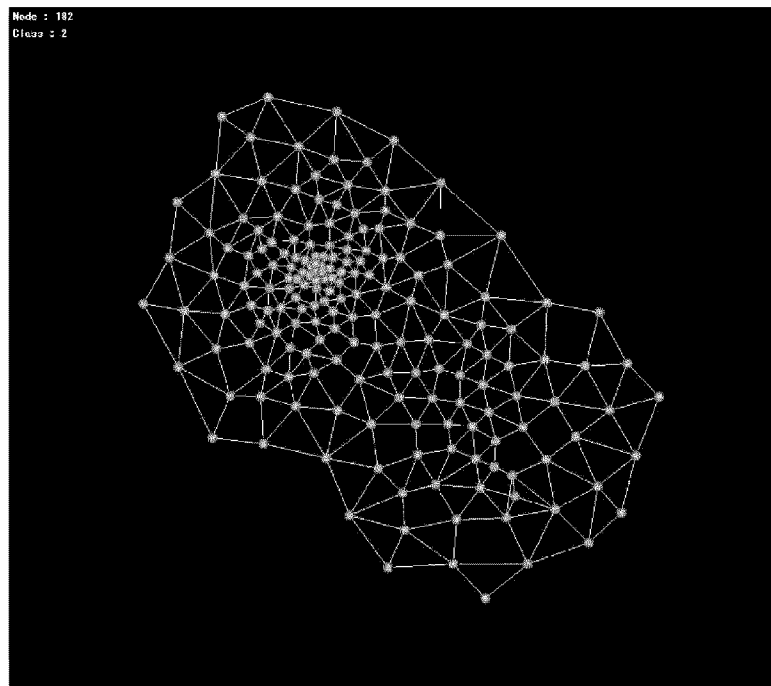
Input : 2764



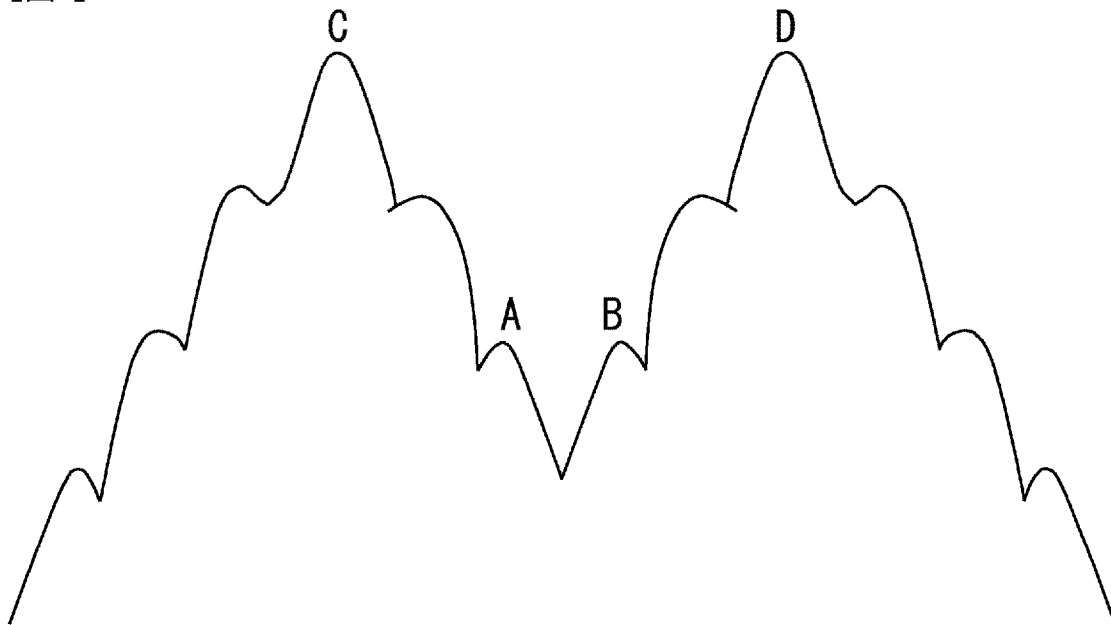
[図2A]



[図2B]

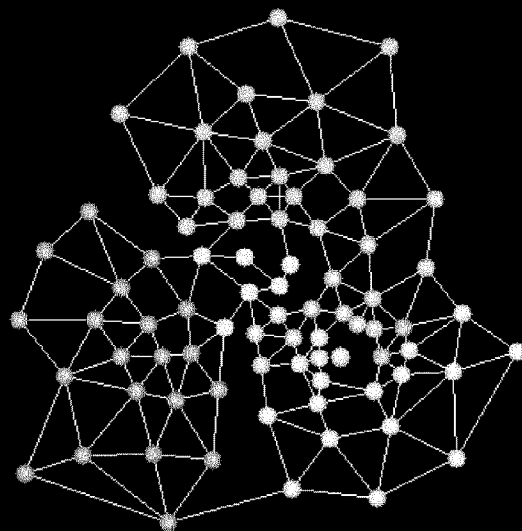


[図3]

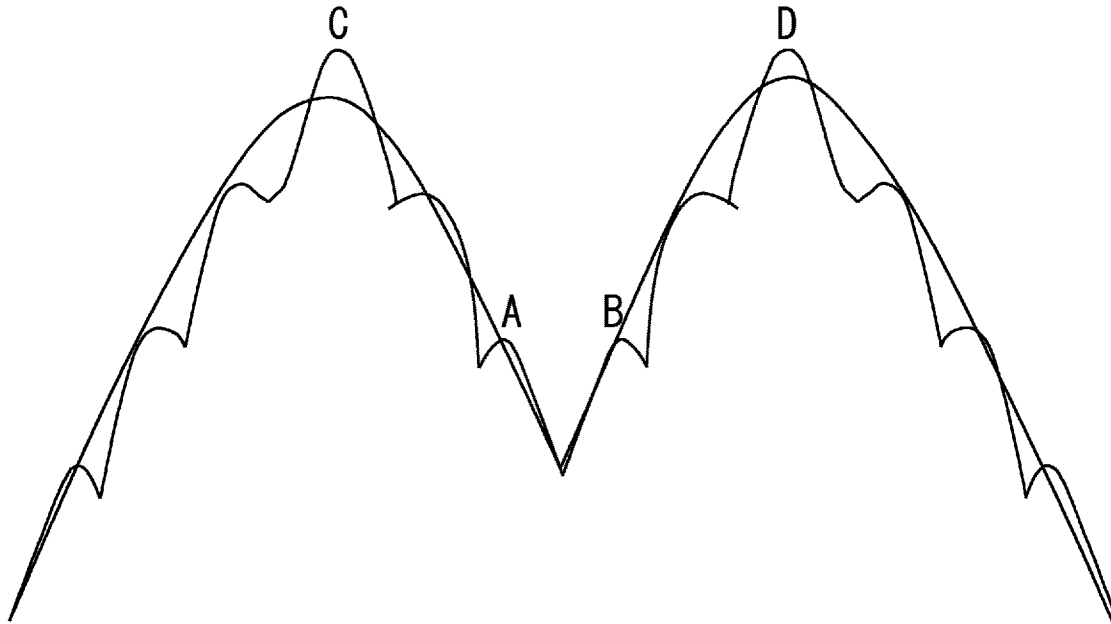


[図4]

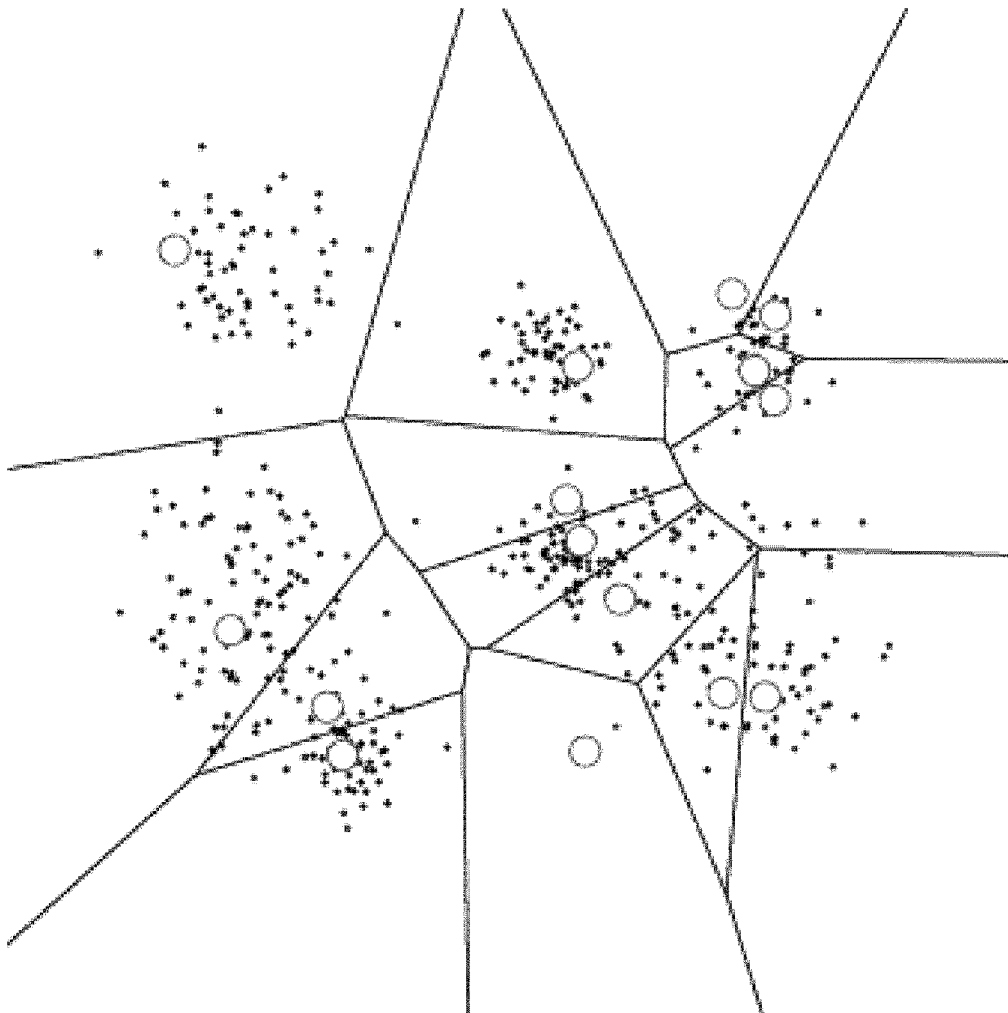
Node : 81
Class : 2

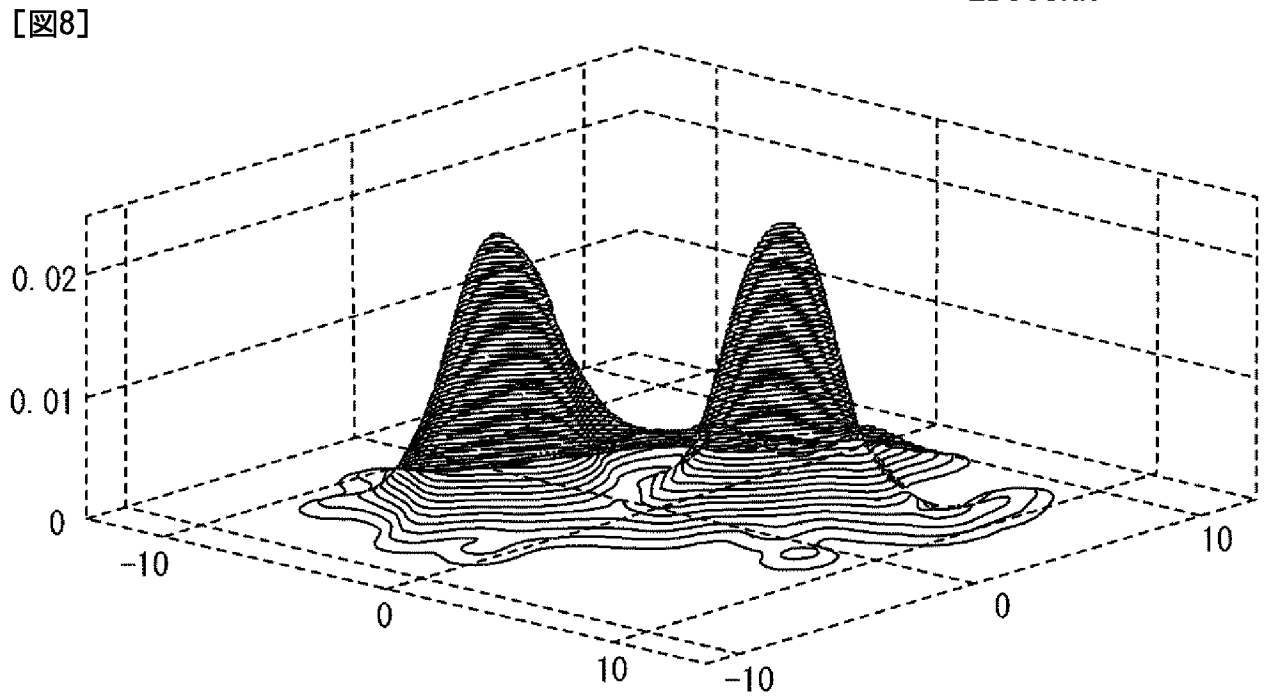
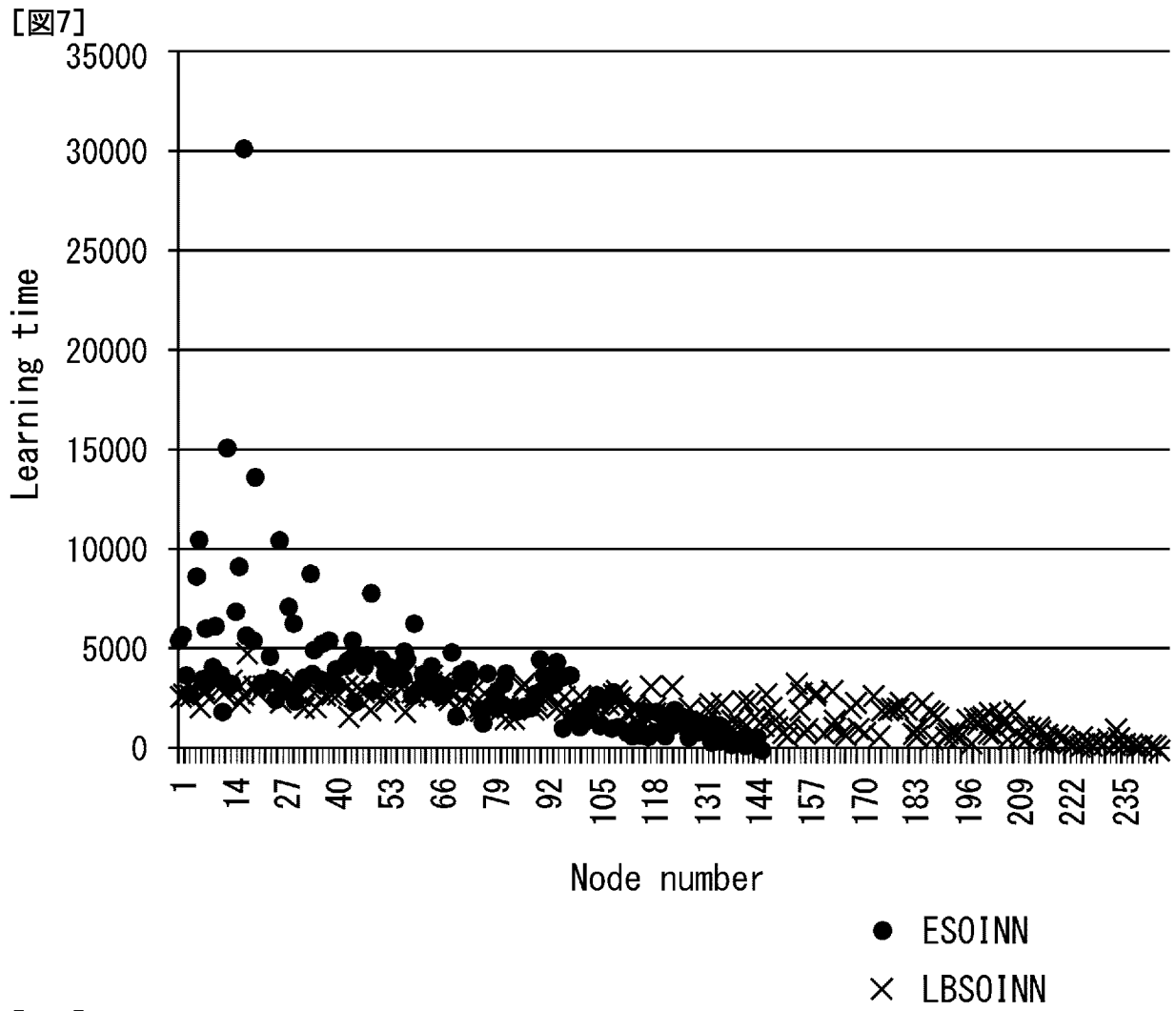


[図5]

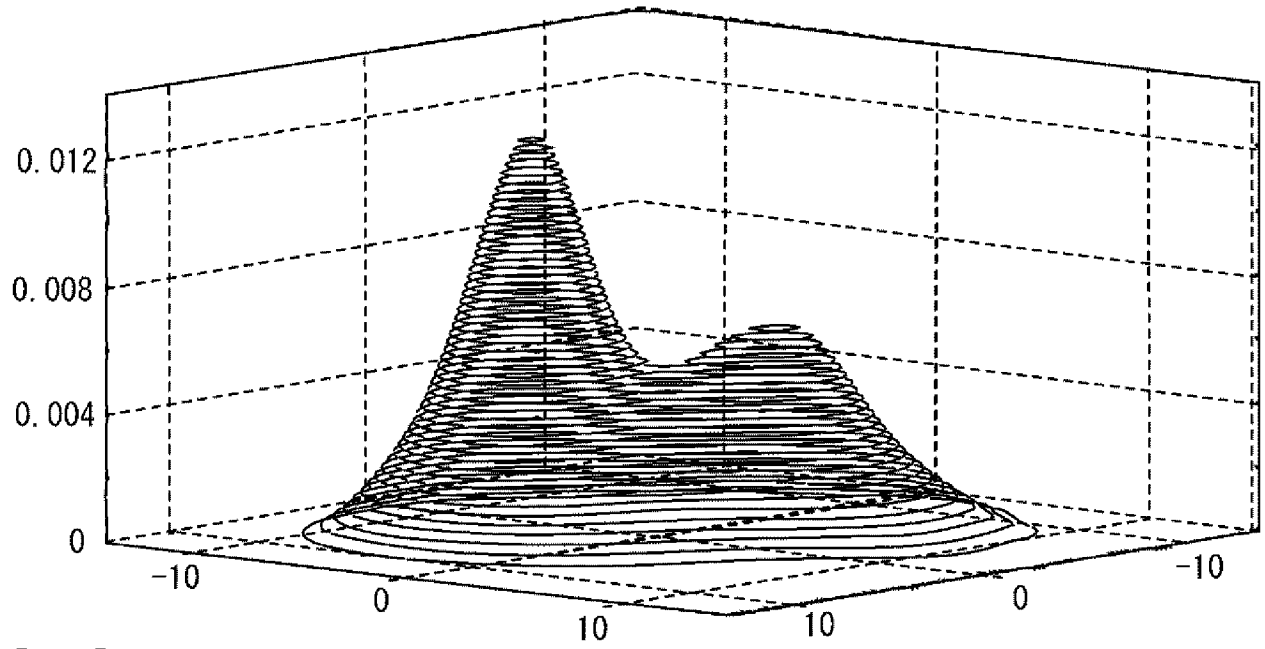


[図6]

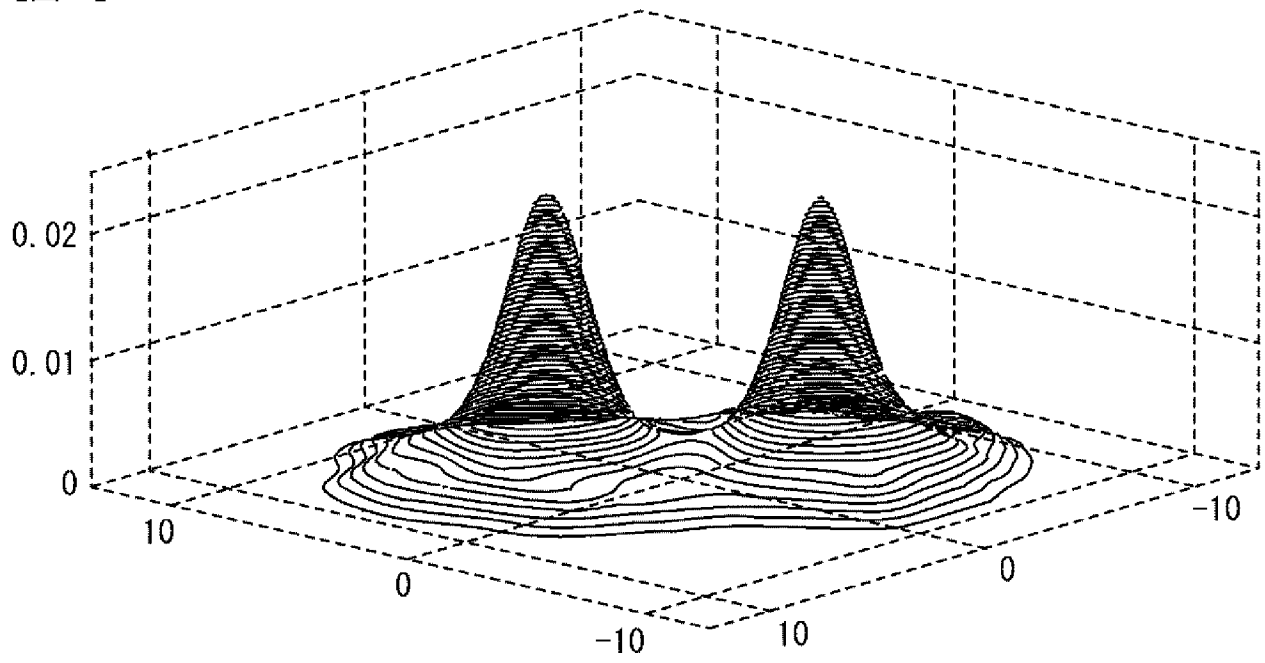




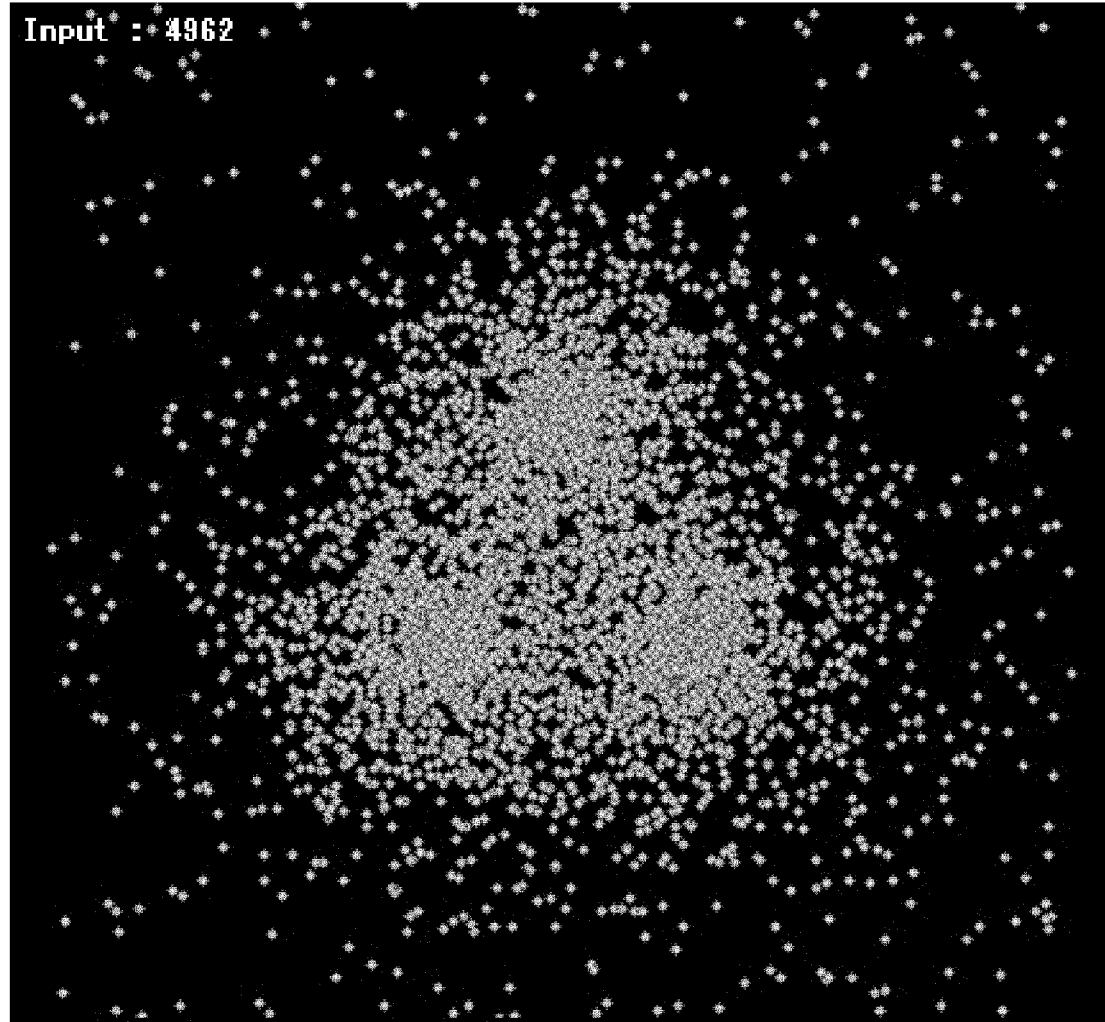
[図9]



[図10]

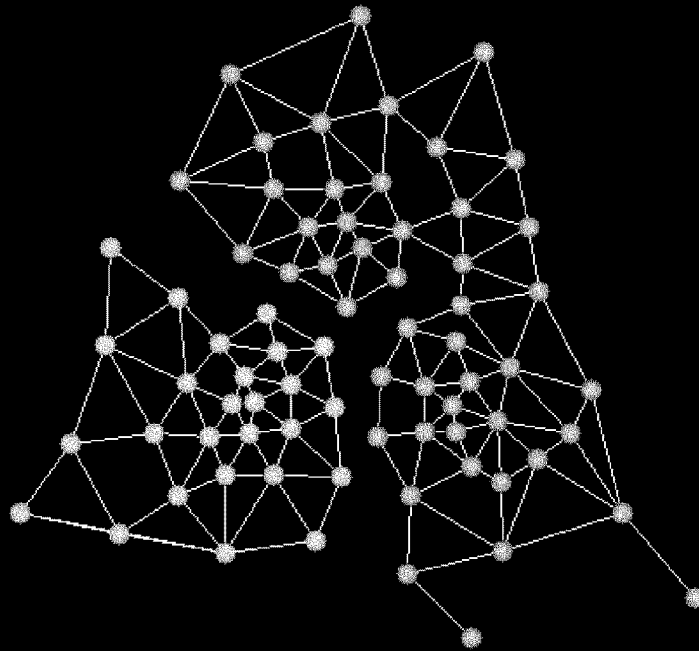


[図11]



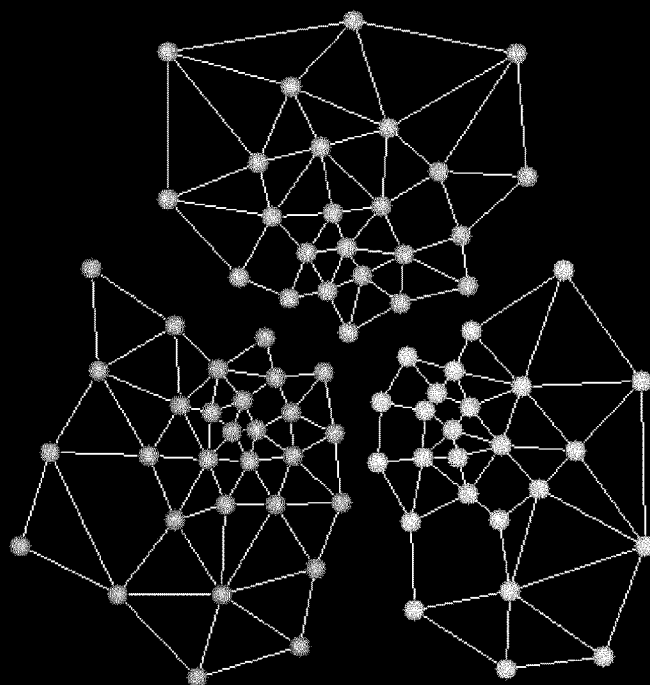
[図12A]

Node : 74
Class : 3



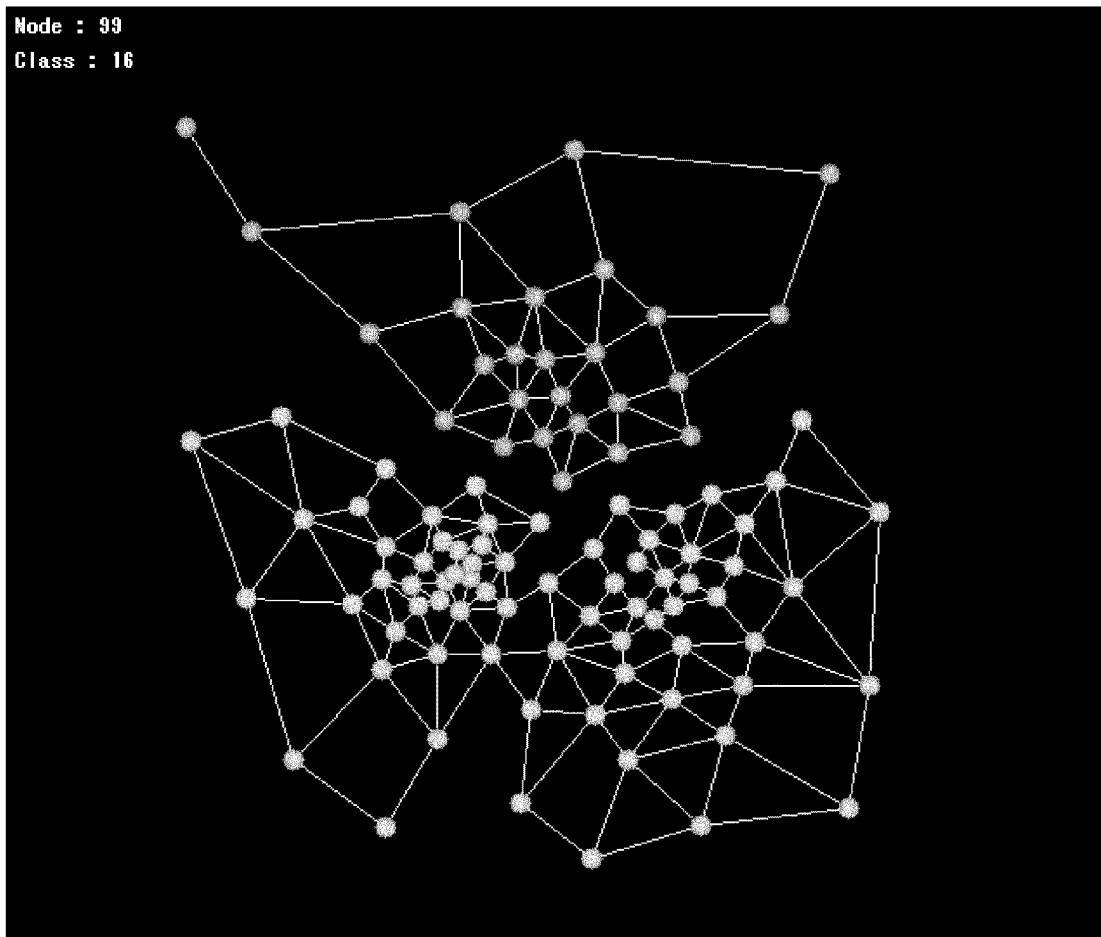
[図12B]

Node : 78
Class : 7



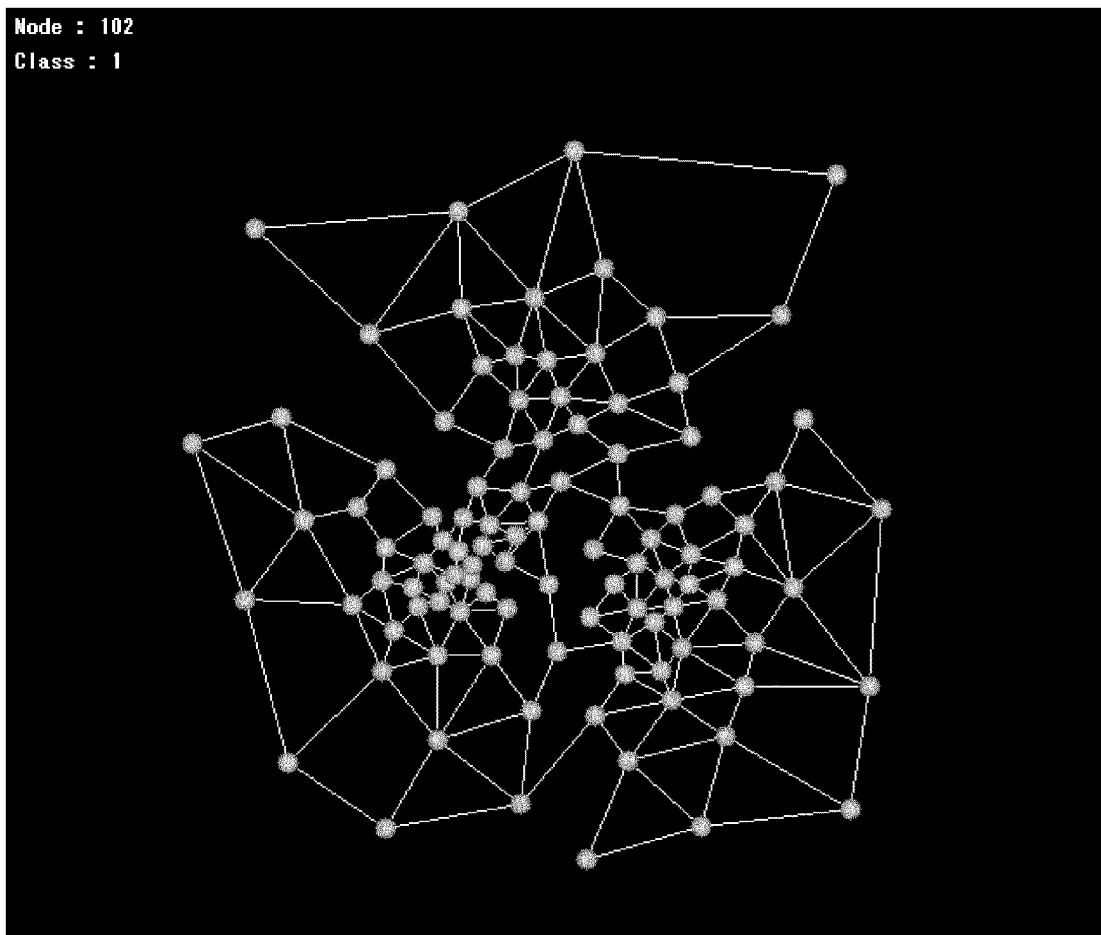
[図12C]

Node : 99
Class : 16



[図12D]

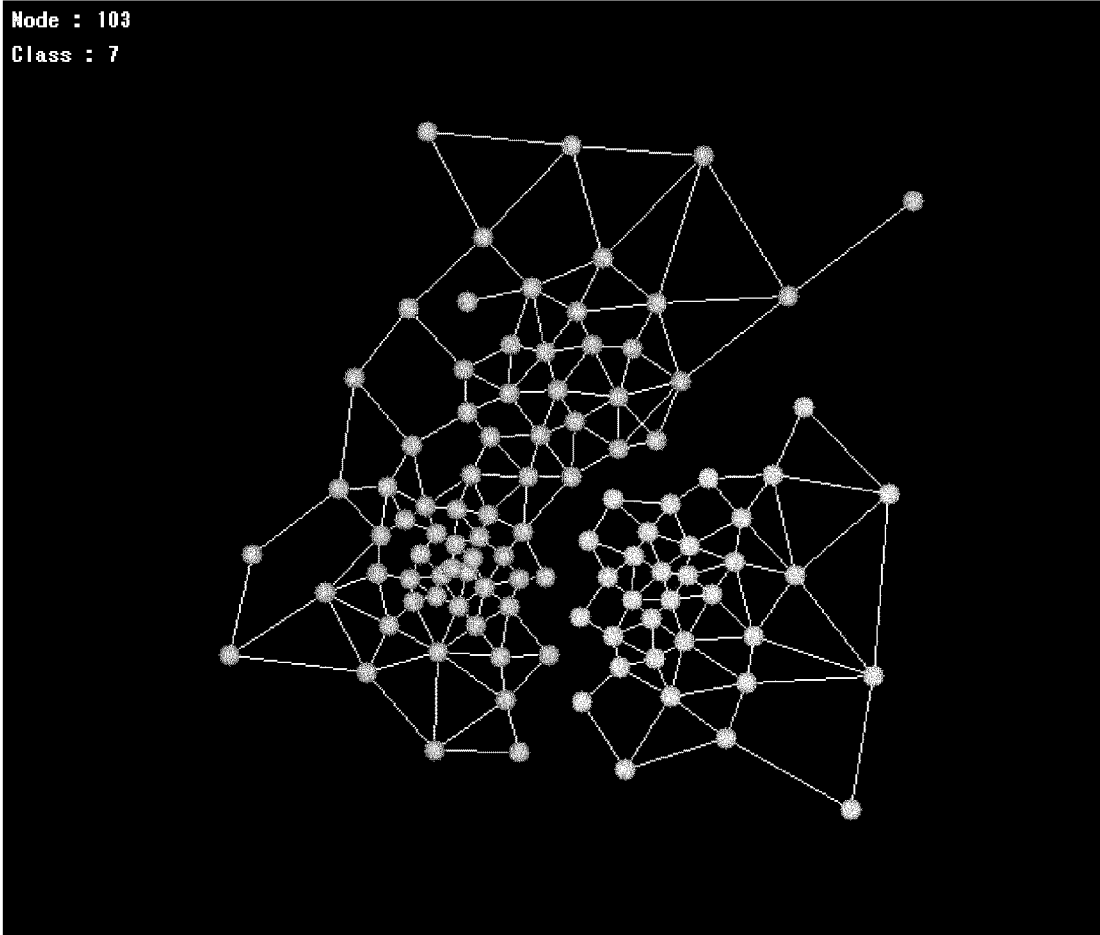
Node : 102
Class : 1



[図12E]

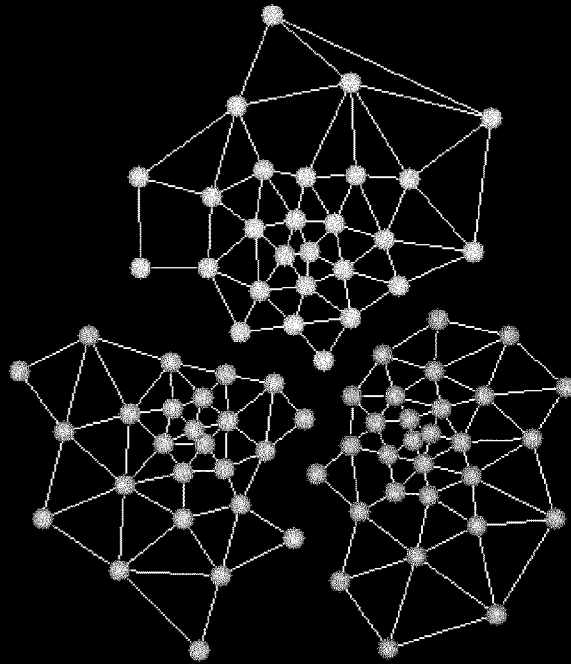
Node : 103

Class : 7



[図12F]

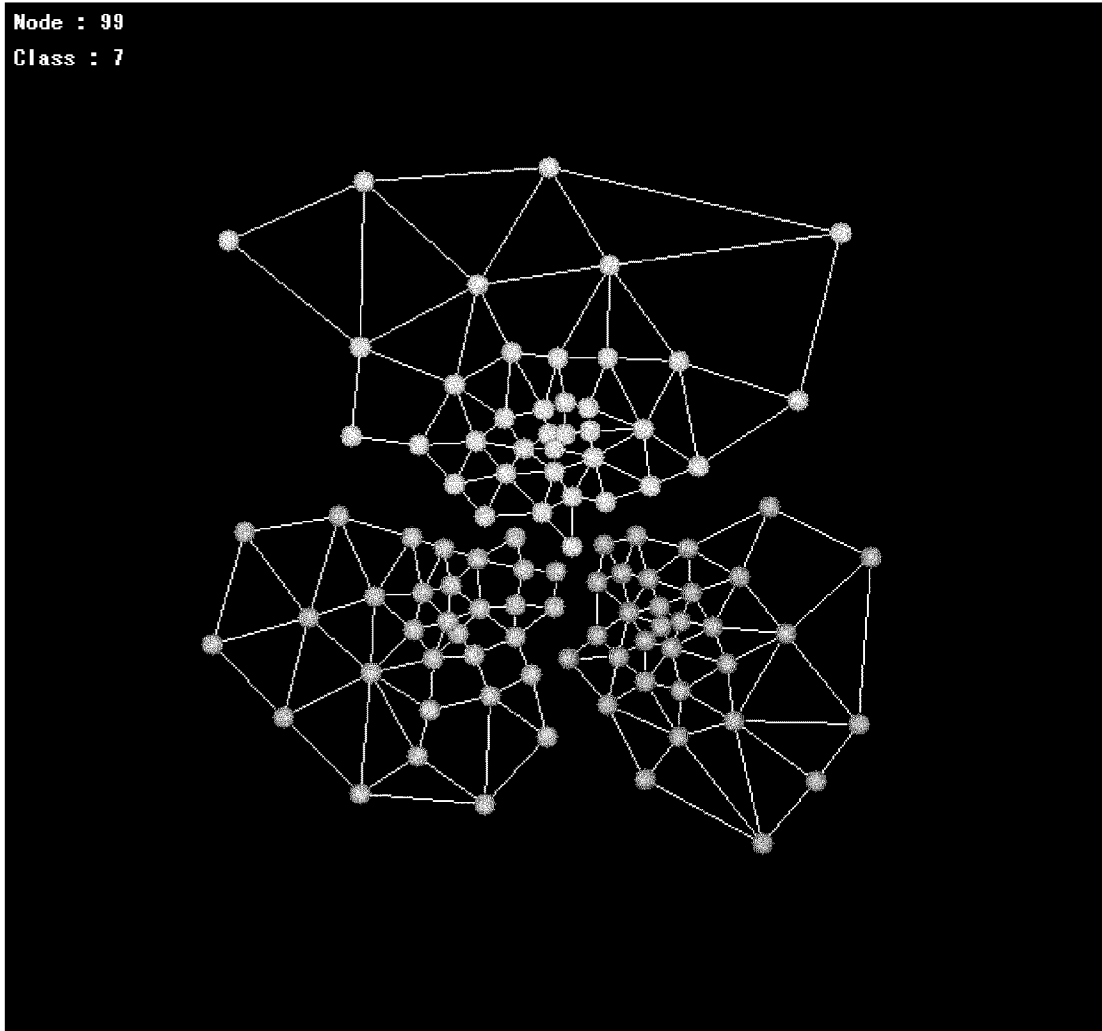
Node : 81
Class : 13



[図12G]

Node : 99

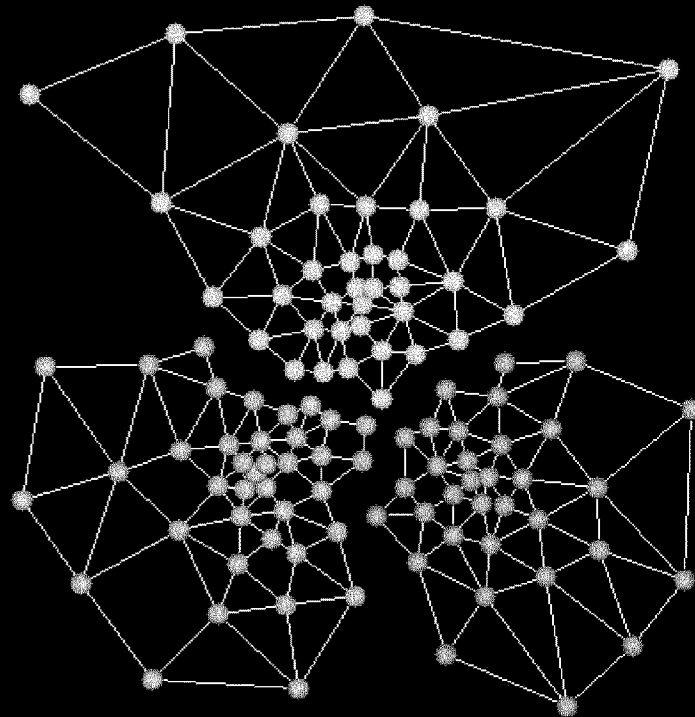
Class : 7



[図12H]

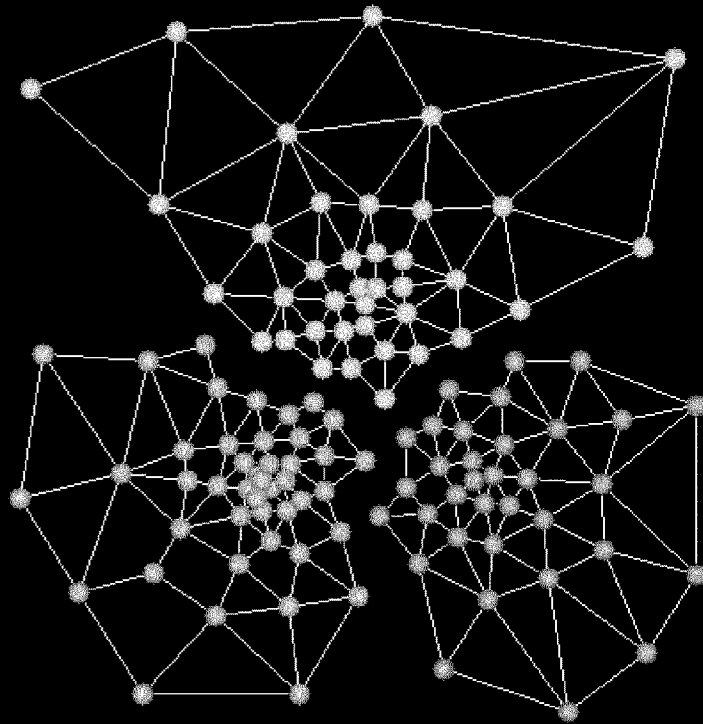
Node : 109

Class : 11



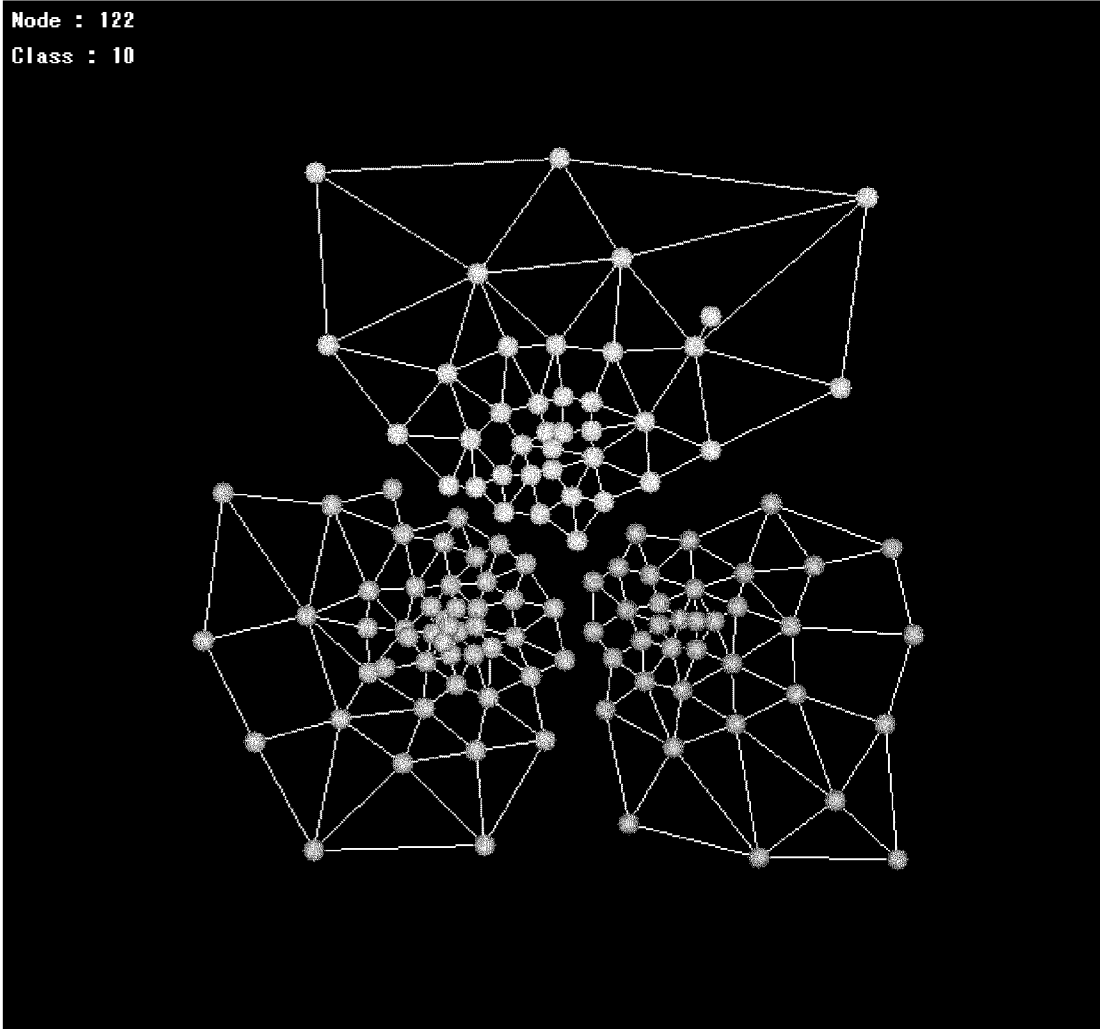
[図12I]

Node : 116
Class : 5

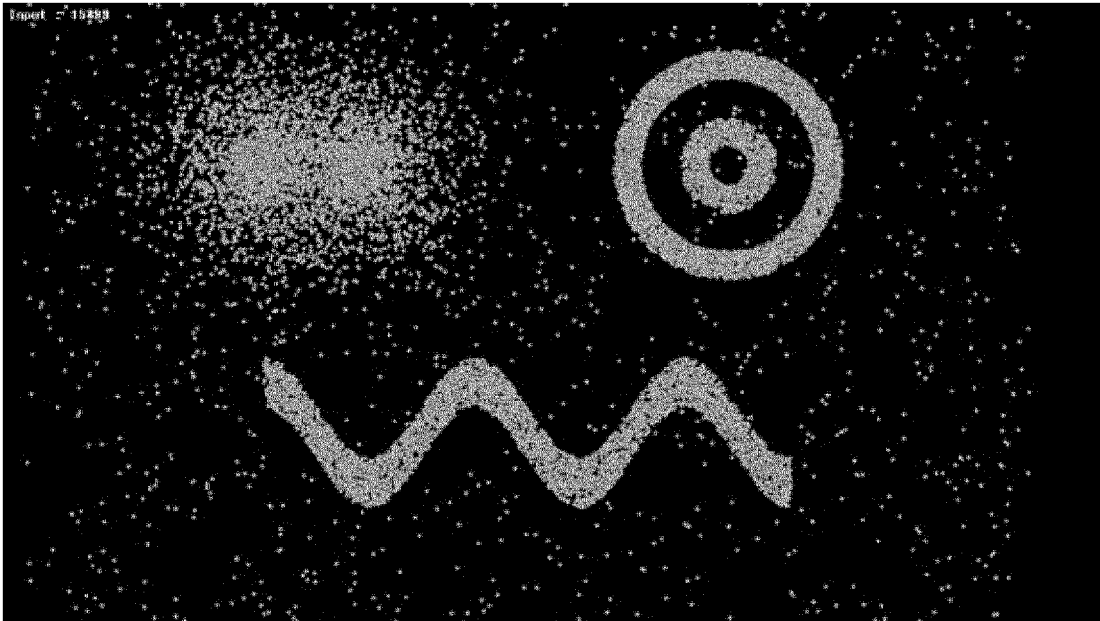


[図12J]

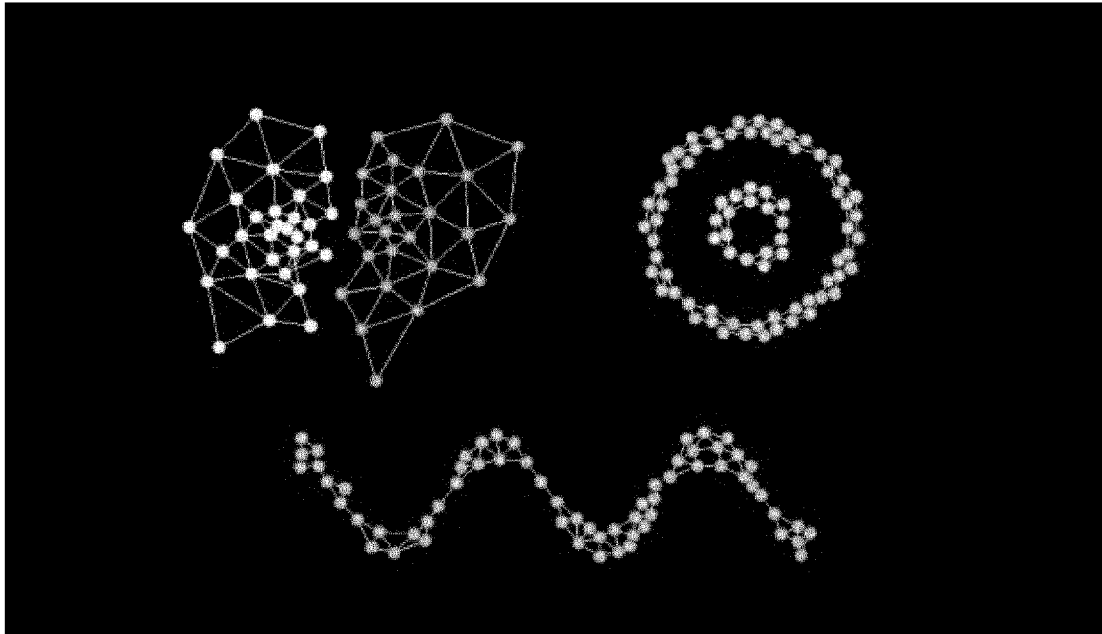
Node : 122
Class : 10



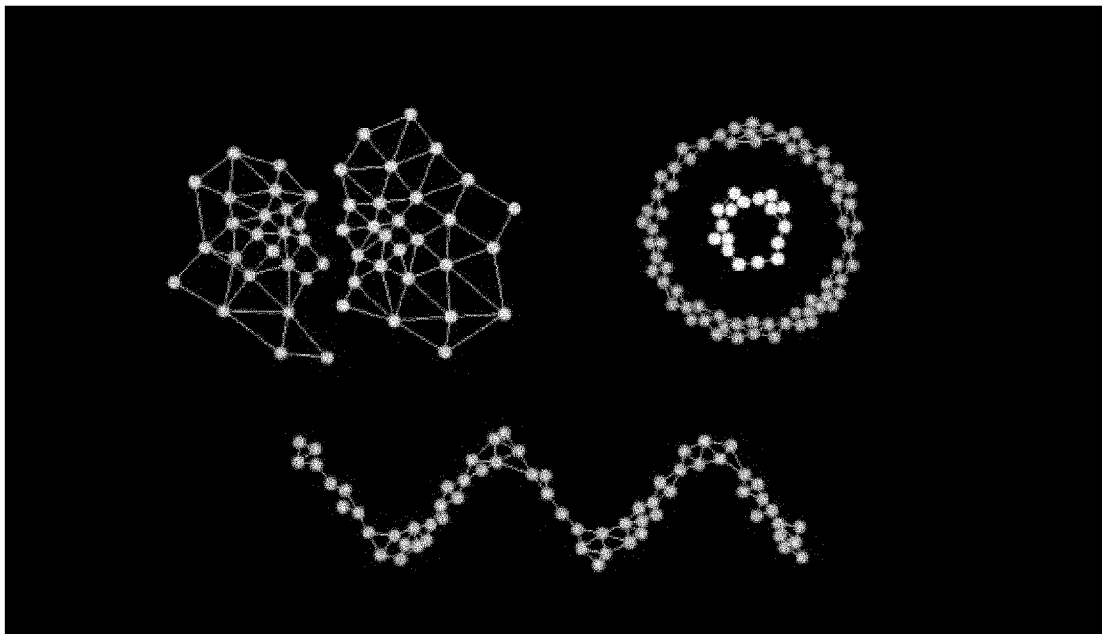
[図13]



[図14]



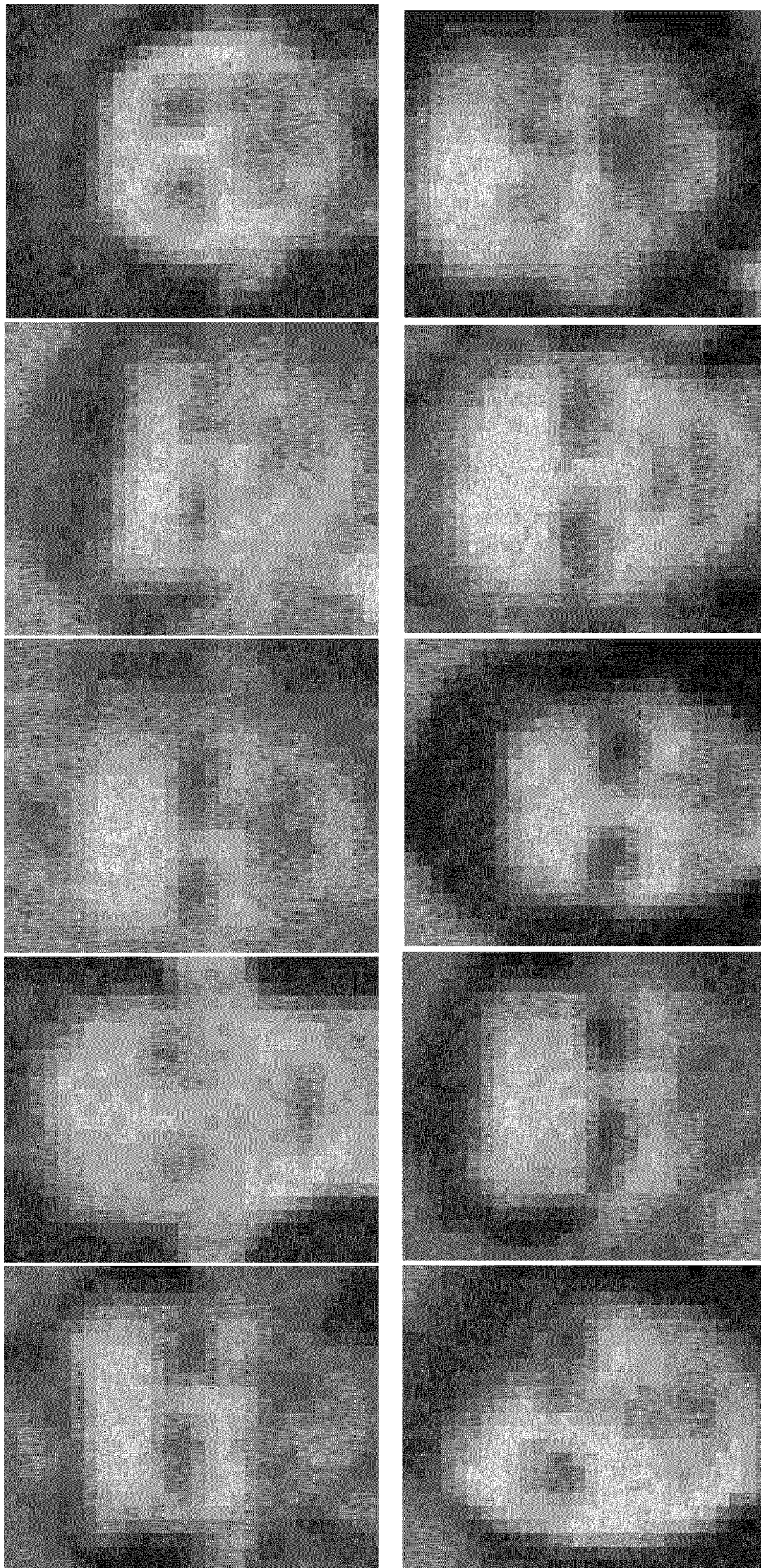
[図15]

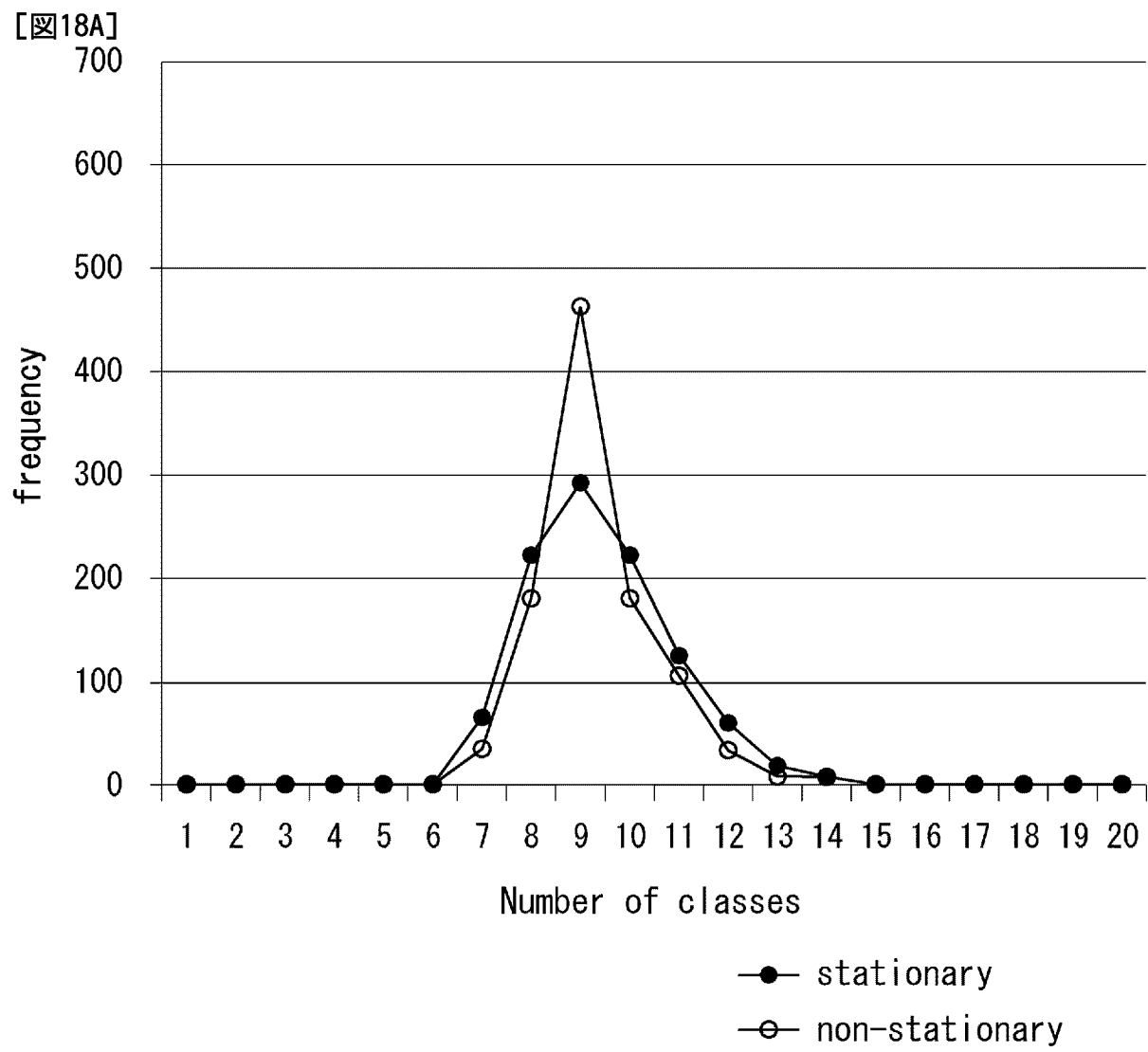


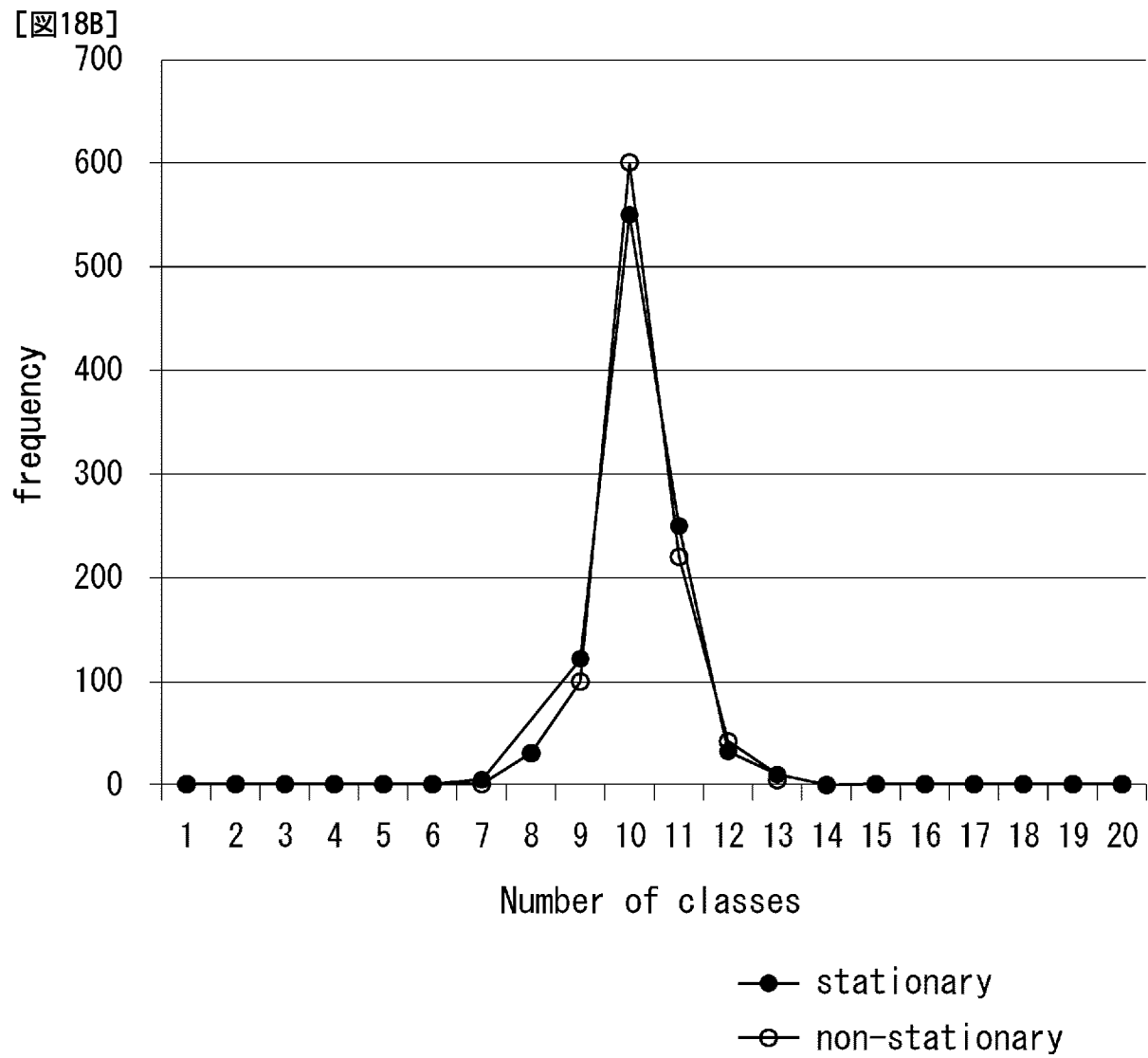
[図16]



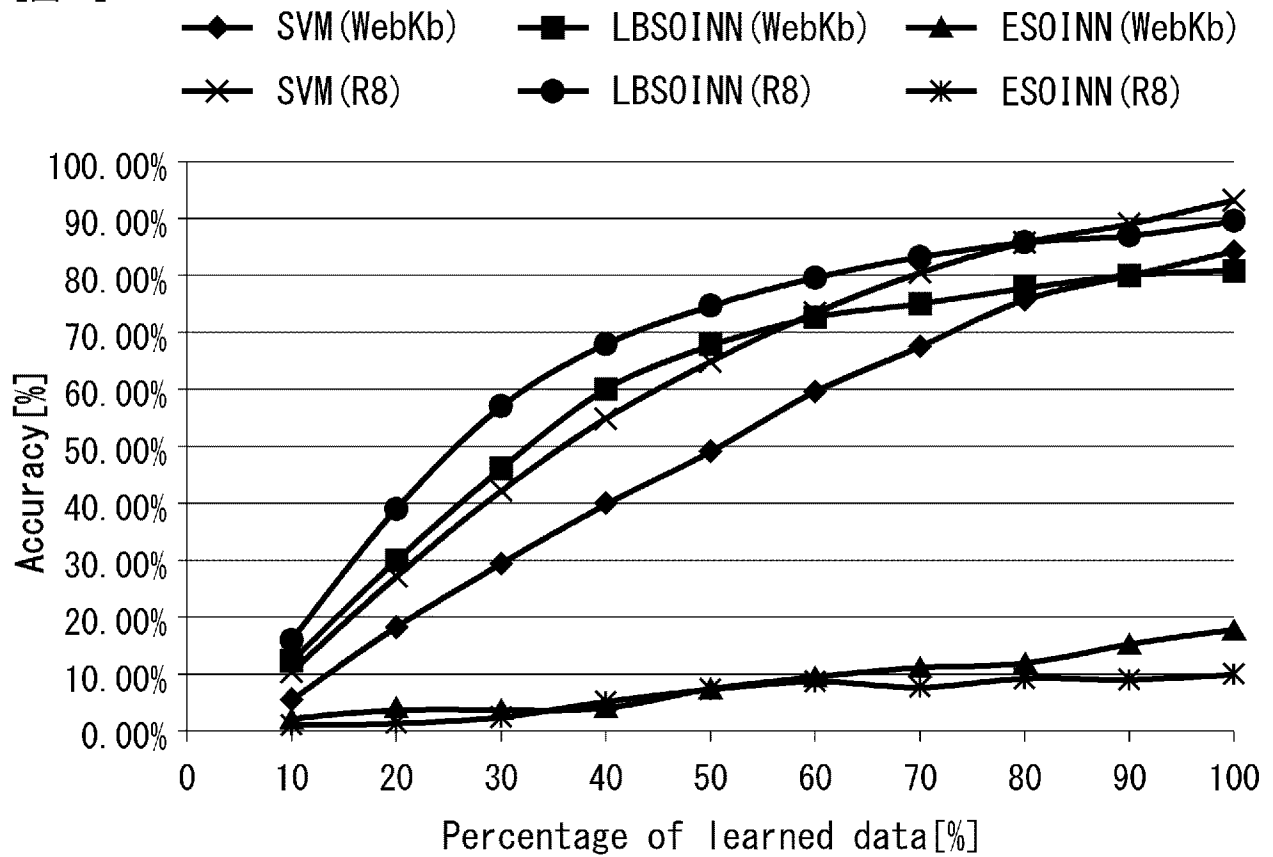
[図17]



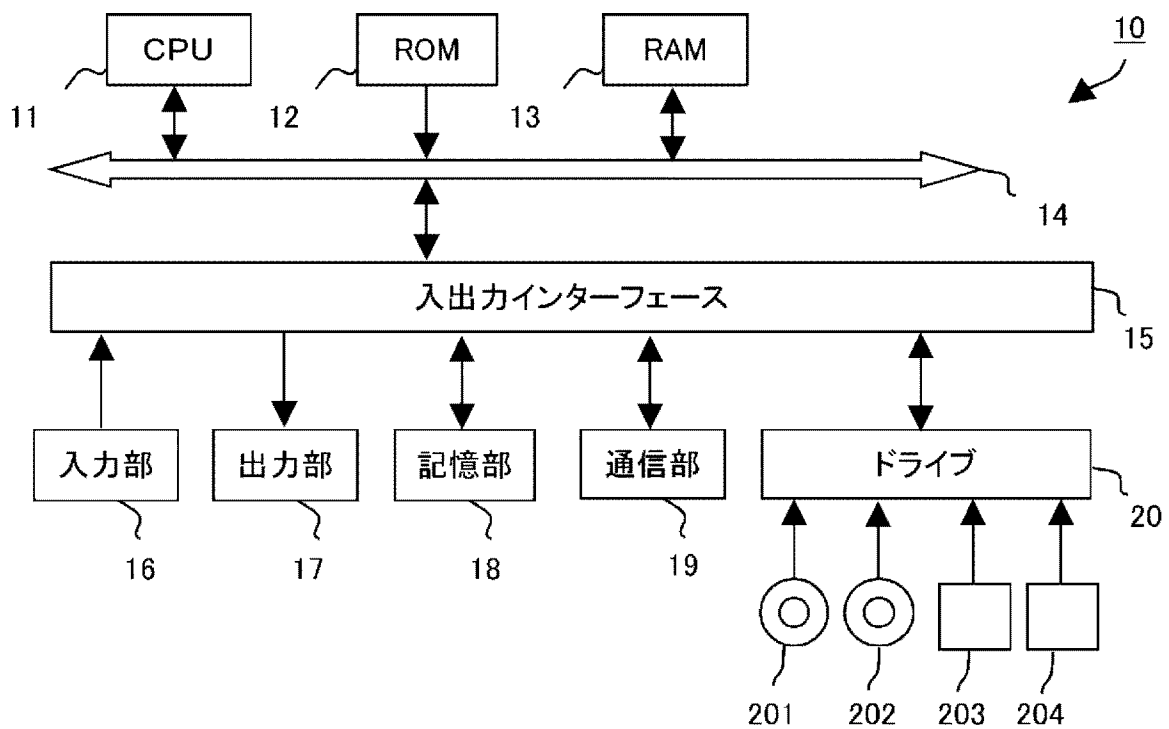




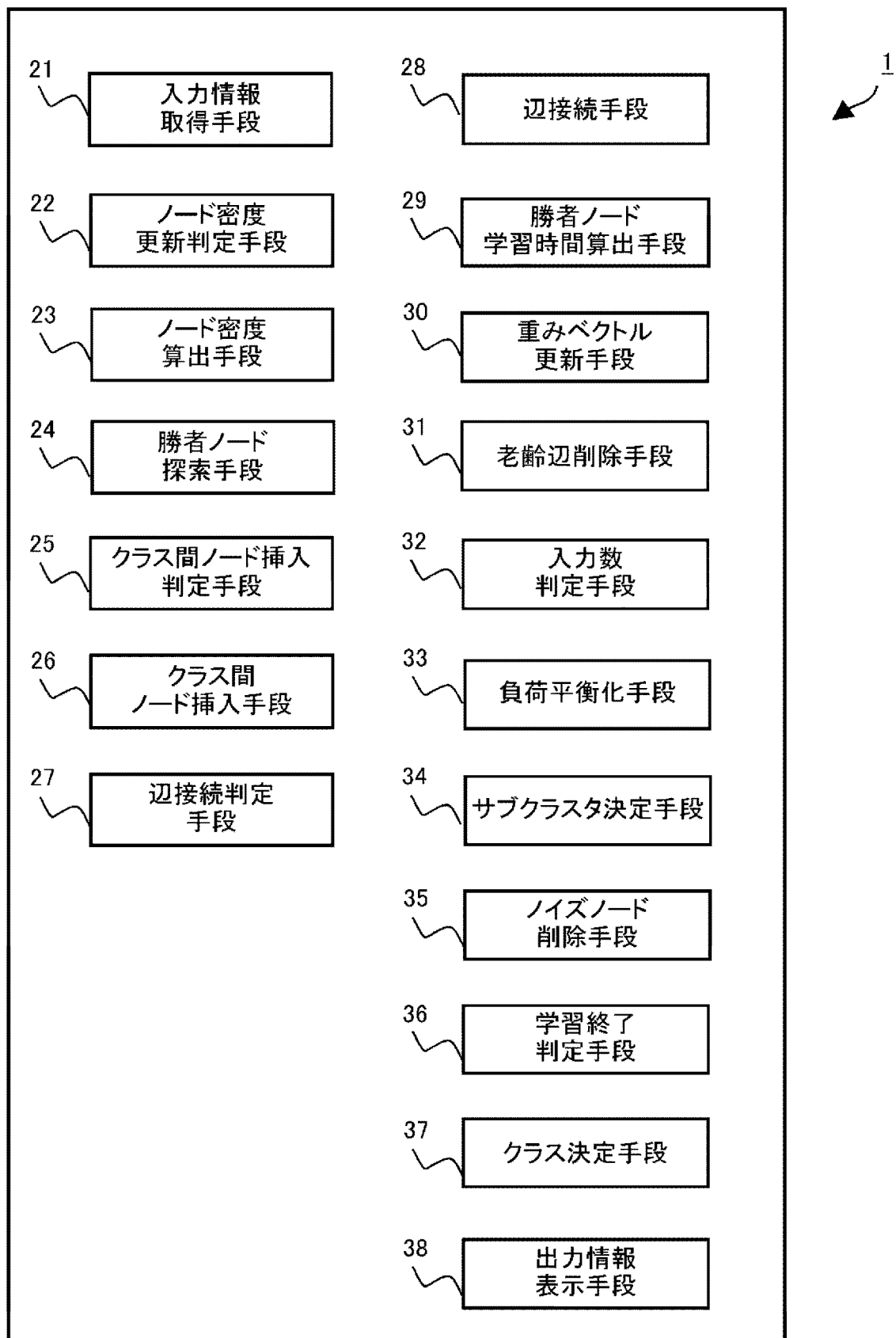
[図19]



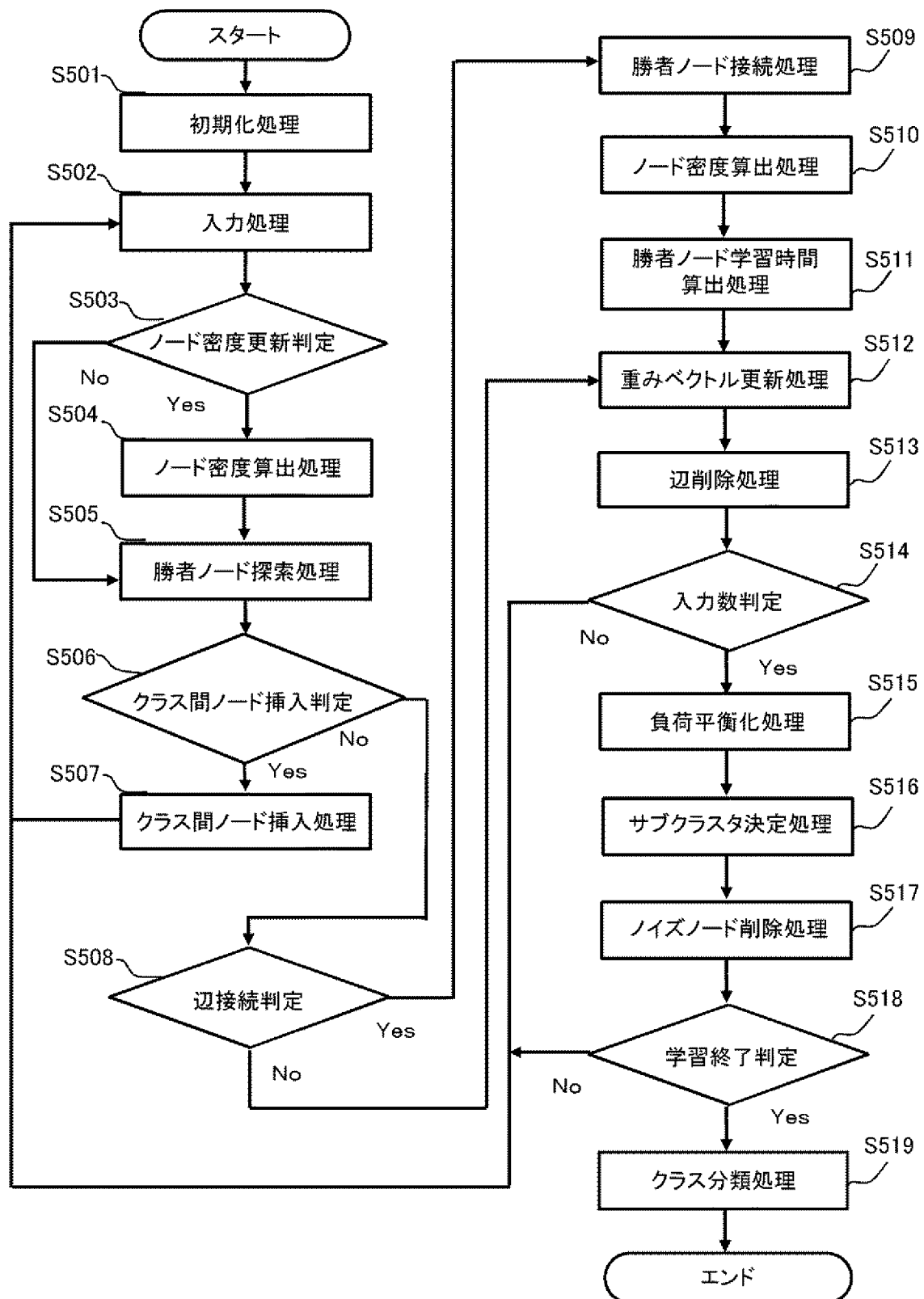
[図20]



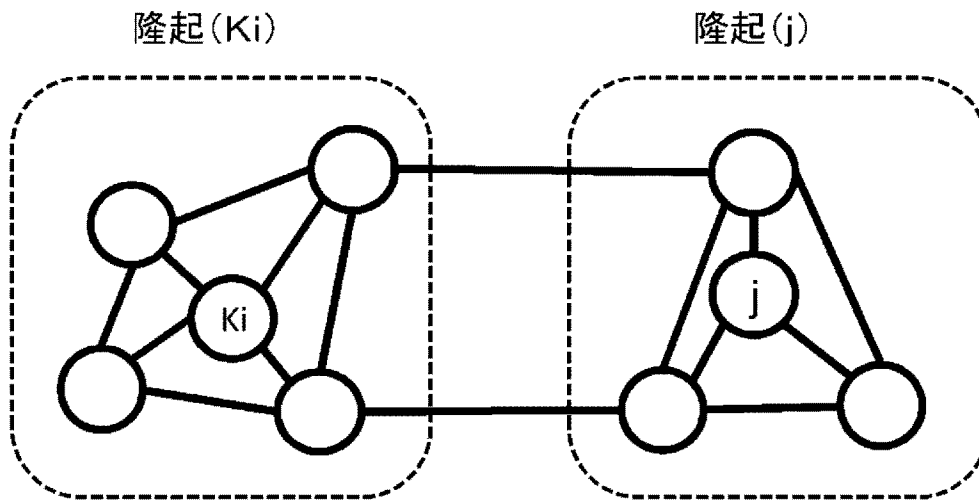
[図21]



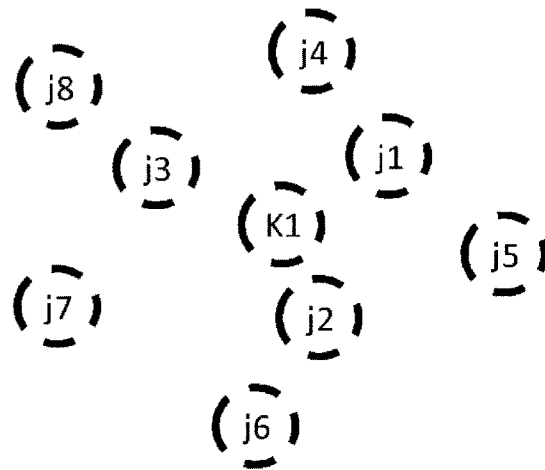
[図22]



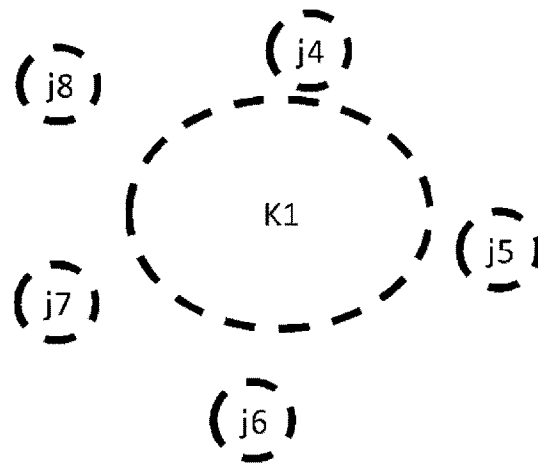
[図23]



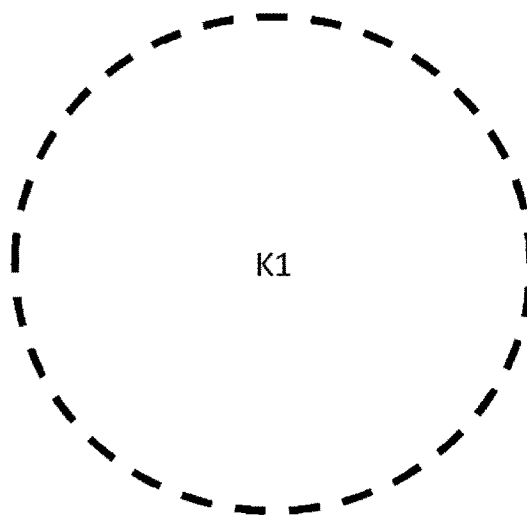
[図24A]



[図24B]



[図24C]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2014/000390

A. CLASSIFICATION OF SUBJECT MATTER

G06N3/08(2006.01)i, G06N3/04(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06N3/08, G06N3/04

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho	1922-1996	Jitsuyo Shinan Toroku Koho	1996-2014
Kokai Jitsuyo Shinan Koho	1971-2014	Toroku Jitsuyo Shinan Koho	1994-2014

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 2013-25398 A (Tokyo Institute of Technology), 04 February 2013 (04.02.2013), entire text; the whole (Family: none)	1-11
A	JP 2009-74995 A (The University of Electro-Communications), 09 April 2009 (09.04.2009), entire text; all drawings (Family: none)	1-11
A	Kazuhiro TOKUNAGA, "Verification of the growth algorithm as the base of the intellectual growth system of robot", IEICE Technical Report, 11 March 2010 (11.03.2010), vol.109, no.461, pages 125 to 130	1-11

☒ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search
17 April, 2014 (17.04.14)

Date of mailing of the international search report
28 April, 2014 (28.04.14)

Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2014/000390

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	Youki KAMIYA, An Online Semi-Supervised Clustering Algorithm Based on a Self-organizing Incremental Neural Network, Proceedings of International Joint Conference on Neural Networks, 2007.08.17, pp.1061-1066	1-11

A. 発明の属する分野の分類（国際特許分類（I P C））

Int.Cl. G06N3/08(2006.01)i, G06N3/04(2006.01)i

B. 調査を行った分野

調査を行った最小限資料（国際特許分類（I P C））

Int.Cl. G06N3/08, G06N3/04

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	1922-1996年
日本国公開実用新案公報	1971-2014年
日本国実用新案登録公報	1996-2014年
日本国登録実用新案公報	1994-2014年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
A	JP 2013-25398 A（国立大学法人東京工業大学）2013.02.04, 全文、全部（ファミリーなし）	1-11
A	JP 2009-74995 A（国立大学法人電気通信大学）2009.04.09, 全文、全図（ファミリーなし）	1-11
A	徳永 憲洋, ロボットの知能発達システムの基盤となる成長型アルゴリズムの検証, 電子情報通信学会技術研究報告, 2010.03.11, 第109巻／第461号, 第125-130頁	1-11

☒ C欄の続きにも文献が列挙されている。☐ パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」特に関連のある文献ではなく、一般的技術水準を示すもの

「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの

「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）

「O」口頭による開示、使用、展示等に言及する文献

「P」国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの

「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの

「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの

「&」同一パテントファミリー文献

国際調査を完了した日

17.04.2014

国際調査報告の発送日

28.04.2014

国際調査機関の名称及びあて先

日本国特許庁（I S A/J P）

郵便番号100-8915

東京都千代田区霞が関三丁目4番3号

特許庁審査官（権限のある職員）

塚田 肇

5 B

3652

電話番号 03-3581-1101 内線 3545

C (続き) . 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
A	Youki KAMIYA, An Online Semi-Supervised Clustering Algorithm Based on a Self-organizing Incremental Neural Network, Proceedings of International Joint Conference on Neural Networks, 2007.08.17, pp.1061-1066	1 - 1 1