(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3974511号
(P3974511)
(45)発行日　平成19年9月12日(2007.9.12)　(24)登録日　平成19年6月22日(2007.6.22)

(51)Int.Cl.　F I
G06F 17/30 (2006.01)　G06F 17/30 210D
G06F 17/30 350C
G06F 17/30 414A

請求項の数 11　外国語出願　(全 36 頁)

(21)出願番号　特願2002-368276(P2002-368276)
(22)出願日　平成14年12月19日(2002.12.19)
(65)公開番号　特開2004-199472(P2004-199472A)
(43)公開日　平成16年7月15日(2004.7.15)
審査請求日　平成15年8月22日(2003.8.22)

(73)特許権者　390009531
インターナショナル・ビジネス・マシーンズ・コーポレーション
INTERNATIONAL BUSINESS MASCHINES CORPORATION
アメリカ合衆国10504　ニューヨーク州　アーモンク　ニュー　オーチャード　ロード
(74)代理人　100086243
弁理士　坂口　博
(74)代理人　100091568
弁理士　市位　嘉宏
(74)代理人　100108501
弁理士　上野　剛史

最終頁に続く

(54)【発明の名称】情報検索のためのデータ構造を生成するコンピュータ・システム、そのための方法、情報検索のためのデータ構造を生成するコンピュータ実行可能なプログラム、情報検索のためのデータ構造

(57)【特許請求の範囲】

【請求項１】

所定のキーワード・リストから生成され、かつドキュメントに対して与えられる階層構造のノードを構成するドキュメント－キーワード・ベクトルとしてデータベースに格納されたドキュメントの情報検索のためのデータ構造を生成するコンピュータ・システムであって、前記コンピュータ・システムは、

前記ドキュメント－キーワード・ベクトルについてランダムに決定されたトップ・ノードに対するノード間距離により連結される階層構造を生成させる部分と前記ノード間にノード間の距離に関連して検索されたノードと同一の階層レベルまたはそれ以上の階層レベルのノードを含むパッチを生成するパッチ規定部とを含んで構成され、前記トップ・ノードからの距離により規定された階層構造を使用して決定され、かつ距離関数により定義される類似性を有するノード・グループを生成するための近傍パッチ生成部と、

前記パッチに含まれる前記ノードが複数の前記パッチにわたって含まれる割合を規定するパッチ間コンフィデンス値を計算することにより前記パッチ間の類似性を計算し、前記類似性に対応して、前記パッチからクラスターの要素を選択して前記ドキュメント－キーワード・ベクトルについてのクラスター・データを生成するためのクラスター見積もり部とを含む

コンピュータ・システム。

【請求項２】

前記コンピュータ・システムは、前記パッチ間のパッチ間コンフィデンス値およびパッ

チ内コンフィデンス値を決定するコンフィデンス決定部を含む、請求項１に記載のコンピュータ・システム。
【請求項３】
前記クラスター見積もり部は、前記パッチ内コンフィデンス値についての設定値により前記クラスター・データのサイズを増加または減少させる、請求項２に記載のコンピュータ・システム。
【請求項４】
所定のキーワード・リストから生成され、かつドキュメントに対して与えられる階層構造のノードを構成するドキュメント－キーワード・ベクトルとしてデータベースに格納されたドキュメントの情報検索のためのデータ構造をコンピュータに生成させる方法であって、前記方法は、前記コンピュータが、
前記ドキュメント－キーワード・ベクトルについてランダムに決定されたトップ・ノードに対するノード間距離により連結される階層構造を生成し、格納領域に階層データを格納するステップと、
前記ノード間にノード間の距離尺度に関連して検索されたノードと同一の階層レベルまたはそれ以上の階層レベルのノードを含むパッチを生成するパッチ規定部とを含んで構成され、前記トップ・ノードからの距離により規定された階層構造を使用して決定され、かつ距離関数により定義される類似性を有する近傍パッチを生成し、前記パッチを格納領域に格納するステップと、
前記階層データおよび前記パッチとを読み込んで、前記パッチに含まれる前記ノードが複数の前記パッチにわたって含まれる割合を規定するパッチ間コンフィデンス値を計算することにより前記パッチ間の類似性を計算し、前記類似性に対応して前記パッチを参照可能に格納領域に対応するリストとして格納するステップと、
前記パッチの要素を選択して前記ドキュメント－キーワード・ベクトルについてのクラスター・データを生成するステップと
を実行する、方法。
【請求項５】
さらに、前記パッチ内コンフィデンス値についての設定値により前記クラスター・データのサイズを増加または減少するステップを含む、請求項４に記載の方法。
【請求項６】
請求項４または５のいずれかに記載の方法をコンピュータが実行するためのコンピュータ実行可能なプログラム。
【請求項７】
請求項６に記載のプログラムを格納する、コンピュータ可読な記録媒体。
【請求項８】
所定のキーワード・リストから生成され、かつドキュメントに対して与えられる階層構造のノードを構成するドキュメント－キーワード・ベクトルとしてデータベースに格納されたドキュメントの情報検索のための情報検索システムであって、前記情報検索システムは、
請求項１～３のいずれか１項に記載のコンピュータ・システムと、
前記コンピュータ・システムにより見積もられたクラスター・データをディスプレイ手段に表示させるグラフィカル・ユーザ・インタフェイスと
を含む情報検索システム。
【請求項９】
前記情報検索システムは、さらに、クエリーを受信すると共に情報検索のためのデータを抽出してクエリー・ベクトルを生成するユーザ・クエリー受信部と、前記ドキュメント－キーワード・ベクトルと前記クエリー・ベクトルとの間の類似性を算出する情報検索部と
を含む、請求項８に記載の情報検索システム。
【請求項１０】

前記クラスター・データは、ユーザ入力クエリーに関連して検索された前記ドキュメント－キーワード・ベクトルを使用して見積もられる、請求項９に記載の情報検索システム。

【請求項１１】

ユーザ入力クエリーに応答してディスプレイ・デバイス上に見積もられたクラスターをグラフィカルに表示するためのグラフィカル・ユーザ・インタフェイス・システムであって、前記グラフィカル・ユーザ・インタフェイス・システムは、

ドキュメントを格納するデータベースと、

請求項１～３のいずれか１項に記載のコンピュータ・システムと、

前記コンピュータ・システムにより見積もられた前記クラスターと共に、前記クラスターの間のコンフィデンス関係とクラスター・サイズとを含む前記クラスターの階層情報とを画面上に表示するためのディスプレイと

を含むグラフィカル・ユーザ・インタフェイス・システム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、大規模データベースからの情報検索に関し、より詳細には、情報検索のためのデータ構造を生成するためのコンピュータ・システム、そのための方法、情報検索のためのデータ構造を生成するコンピュータ実行可能なプログラム、情報検索のためのデータ構造を生成するコンピュータ実行可能なプログラムを記憶したコンピュータ可読な記憶媒体、情報検索システム、およびグラフィカル・ユーザ・インタフェイス・システムに関する。

【０００２】

【従来の技術】

近年における情報処理システムは、例えば、ニュース・データ、顧客情報、特許情報および株式市場データと言った大量のデータを取り扱うことがますます期待されている。上述したデータベースのユーザは、所望するデータの迅速、効果的、かつ高精度な検索を行うことがますます困難になってきている。このため、大規模データベースからの適時的で精度の良いドキュメントの安価な検出を行うことは、多くの種類の事業に対してきわめて価値ある情報を提供することになろう。加えて、ユーザは、時として検出されたデータに関連する例えば、データベース中におけるクラスター情報およびクラスター間にわたる相関関係などについて、さらなる情報を得ることを希望する場合もある。

【０００３】

クラスターを検出するための典型的な方法は、データ要素間の類似の尺度に依存し、類似検索に基づく上述した方法がこれまで提案されており、これらについては以下に挙げることができる。

【０００４】

類似検索（また、曖昧検索としても知られる）は、データベース内のアイテムが与えられたクエリーに対してどの程度類似するかを探索するものである。類似（または、非類似）は、典型的にはある種の実数値または整数値の距離である“尺度”を使用した距離(dist)、すなわち、

【０００５】

(1) dist(p, q) ≧ 0 for all p, q (non-negativity);
(2) dist(p, q) = dist(q, p) for all p, q (symmetry);
(3) dist(p, q) = 0 if and only if p = q;
(4) dist(p, q) + dist(q, r) ≧ dist(p, r) （すべてのp, q, r に対して）
(三角不等式)

を使用することにより、モデリングされる。

【０００６】

上述した距離関数が存在する対象のセットは、いずれも距離空間として参照することがで

きる。クエリー時において、距離評価の数を低減することが可能なデータ構造は、インデックスとして知られている。類似クエリーのための多くの方法が提案されている。距離空間における類似クエリーとしては、下記のような概ね２通りが知られている。
（Ａ）k－近接クエリー：クエリー要素qと正の正数kとを与え、qに対して近い方からk番目のデータベース要素を報告する。
（Ｂ）レンジ・クエリー：クエリー要素qと、距離rとを与え、dist(p, q)≦rとなるアイテムqを報告するというものである。
【０００７】
大規模データベースについて言えば、類似クエリーを、クエリー要素からデータベース要素のすべてについて正確に算出することはきわめてコスト的に高いものとなる。データベース要素間のすべての距離を算出し、格納する従来の方法は、データベース要素の数の２乗に比例する時間と記憶空間とを必要とするので、コスト的に高すぎる（すなわち、計算時間とメモリ使用量とに対して２乗のオーダーとなる）。より現実的には、２次以下の記憶空間および前処理時間を使用して時間に対して略直線的なクエリー処理を可能とする検索構造を構築することが目的となりうる。
【０００８】
Ａ．ベクトル空間モデルの概説
近年における情報検索方法は、多くの場合、ベクトル空間モデルを使用してデータベースのドキュメントを表示させる。このようなベクトル空間モデルにおいては、考察しているデータベース内の各ドキュメントはベクトルを伴い、ベクトルの各座標軸は、ドキュメントのキーワード、すなわちアトリビュートを示す。ベクトル空間モデルの詳細については、別の文献に与えられている（Gerald Salton, The SMART Retrieval System - Experiments in Automatic Document Processing, Prentice-Hall, Englewood Cliffs, NJ, USA, 1971）。
【０００９】
Ｂ．類似検索構造の概説
過去３０年にわたり、類似クエリーを取り扱うための多くの種類の構造が提案されてきている。これらのうちの主要なものは、空間インデックスであり、このインデックスは、d個の実数値のアトリビュート・ベクトルとして対象セットがモデルされていることを必要とするものである。他のものは、“距離”インデックスであり、距離インデックスは、距離尺度の存在以外にはデータベース要素の性質に対してまったく仮定をすることがなく、したがって空間検索構造よりもより広い適用性がある。多次元ベクトル空間および距離空間のための検索構造についての最近の総説については、Gaedeら (Volker Gaede and Oliver Gunther, Multidimensional Access Methods, ACM Computing Surveys, 30, 2, 1998, pp. 170-231.), and Chavez ら (Edgar Chavez, Gonzalo Navarro, Ricardo Baeza-Yates and Jose L. Marroquin, Searching in metric spaces, ACM Computing Surveys 33, 3, 2001, pp. 273-321.)を参照されたい。
【００１０】
類似検索は、距離データまたはベクトル・データであるにせよ、“次元の呪い”として参照される効果により、多くの場合に制限を受ける。近年における最近傍クエリーまたは範囲クエリーの計算の一般的な問題は、低いフラクタル分布、本質的な次元の低さ、または分布の他の特性など、空間的な特性を有するデータの分布に基づかない限り、正確な技術でさえもがデータベース全体の連続的な検索に対して実質的な改善をもたらすものではない、ということが証明されている。
【００１１】
データの次元および次元の呪いについてのさらなる情報については、例えば、Chavezら（前掲）, Pagelら (Bernd-Uwe Pagel, Flip Korn and Christos Faloutsos, Deflating the dimensionality curse using multiple fractal dimensions, Proc. 16th International Conference on Data Engineering (ICDE 2000), San Diego, USA, IEEE CS Press, 2000, pp. 589-598.), Pestov (Vladimir Pestov, On the geometry of similarity sear

ch: dimensionality curse and concentration of measure, Information Processing Letters, 73, 2000, pp. 47-51.), and Weberら (Roger Weber, Hans-J. Schek and Stephen Blott, A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces, Proc. 24th VLDB Conference, New York, USA, 1998, pp. 194-205)を参照されたい。

【００１２】

Ｃ．近似的な類似検索の概説

次元の呪いを回避する試みとして、研究者は、計算の高速化を得ることを希望し、類似クエリーを幾分か犠牲にすることを考案した。これらの技術の詳細については他の文献、例えばIndykら (P. Indyk and R. Motwani, Approximate nearest neighbors: towards removing the curse of dimensionality, Proc. 30th ACM Symposium on Theory of Computing, Dallas, 1998, pp. 604-613.), and Ferhatosmanogluら (Hakan Ferhatosmanoglu, Ertem Tuncel, Divyakant Agrawal and Amr El Abbadi, Approximate nearest neighbor searching in multimedia databases, Proc. 17th International Conference on Data Engineering (ICDE), Heidelberg, Germany, IEEE CS Press, 2001, pp. 503-514.); 距離空間については、Ciacciaら (Paolo Ciaccia and Marco Patella, PAC nearest neighbor queries: approximate and controlled search in high-dimensional and metric spaces, Proc. 16th International Conference on Data Engineering (ICDE 2000), San Diego, USA, 2000, pp. 244-255; Paolo Ciaccia, Marco Patella and Pavel Zezula, M-tree: an efficient access method for similarity search in metric spaces, Proc. 23rd VLDB Conference, Athens, Greece, 1997, pp. 426-435.) and Zezulaら (Pavel Zezula, Pasquale Savino, Giuseppe Amato and Fausto Rabitti, Approximate similarity retrieval with M-trees, The VLDB Journal, 7, 1998, pp. 275-293.)を参照されたい。しかしながら、これらの方法は、すべて実際上の有効性に制限を与えるという不都合をもたらすものである。あるものは、データの分布について非現実的な仮定を行ない、他のものは、精度と速度との間のトレード・オフに対して効果的な管理を行うことができないものである。

【００１３】

Ｄ．空間近似サンプル階層(SASH)

Houleら(Michael E. Houle, SASH: a spatial approximation sample hierarchy for similarity search, IBM Tokyo Research Laboratory Research Report RT-0446, 18 pages, February 18, 2002)およびHoule, KobayashiおよびAono (Japanese Patent Application No. 2002-037842)による大規模な多次元データ・セットのための近似的な類似検索構造は、精度－速度トレード・オフに対して著しく良好な制御を可能とするものであり、これを空間近似サンプル階層(SASH)として参照する。SASHは、距離尺度の条件を満足する類似関数を必要とするものの、データの性質に対してはそれ以外の仮定を加えない。各データ要素には、構造内における固有の位置が与えられ、２つの要素間のそれぞれの連結は、それらが密接な関係にあることを示す。階層構造の各レベルは、要素のランダム・サンプルとして構成され、また各レベルのサンプル・サイズは、直上のレベルの概ね２倍となるように構成されている。この構造は、所与の要素vに最も近い複数の要素をvに対して最も類似するものとして組織化される。具体的には、vに対応するノードは、その上のレベルの近傍点のセットへと結合され、またvが近傍点として選択される下層のアイテムのセットへと関連づけられる。

【００１４】

Ｅ．クラスター化技術の概説

用語「クラスター化」は、類似基準にしたがい、ラベル付けされていないデータのいずれかのグループ分けを意味する。従来のクラスター化方法は、概ね分割または階層化に分類することができる。階層化技術は、データのグループ（クラスター）の包含関係を示す木構造を生成し、木の経路がデータ・セット全体に対応する。分割技術は、典型的には関連性のないクラスターの、固定された数の中からデータ点の分布における分類誤差を大局的

に最小にすることに基づくものである。これらの最近の概説（Jain, Murty and Flynn （A. K. Jain, M. N. Murty and P. J. Flynn, Data clustering: a review, ACM Computing Surveys 31, 3, 1999, pp. 264-323.））においては、分割クラスター化手法は、階層化よりもコスト的に高くはならないものの、また柔軟性が大きく劣るとしている。簡略化され、迅速性（時間と複雑さとの間の線形関係が見出される。）と、実装化の容易性とはあるものの、周知の分割アルゴリズムであるK-meansとその変形法でさえも、概ね大規模なデータ・セットに対して良好に機能しない。分割アルゴリズムは、等質的な（近似された）クラスターの生成には向くものの、不規則な形質のクラスターを見つけるには充分に適するとは言えない。

【００１５】

Ｆ．階層凝集クラスター化

階層凝集クラスター化では、各データ点をまず考察し、分離されたクラスターを構築する。クラスターの対は、その後、すべてのデータ点が１つのクラスターに帰属されるまで連続的に統合される。各ステップで生成されたより大きなクラスターは、統合されたd個のサブクラスターの要素を含んでおり、そしてクラスター階層を与える包含関係とされる。統合する対の選択は、クラスター間距離の所定の基準を最小化させるようにして行われる。

【００１６】

Ｇ．近傍共有方法

単純な距離に基づく凝集クラスタリング方法の基準の１つは、より高い密度のクラスターを形成するように方向付けを行うものである。領域内のデータの良好に随伴する低密度のグループは、多くの対の距離が統合しきい値よりも低い場合には、まったく発見されないというリスクを負う。凝集クラスター化のために、より専用化された（より高コストの）、データ要素間の近接性を考慮した距離尺度が提案されている。Jarvisらは、R. A. Jarvis and E. A. Patrick, Clustering using a similarity measure based on shared nearest neighbors, IEEE Transactions on Computers C-22, 11, Nov. 1973, pp. 1025-1034.において、任意の類似基準distと、固定した整数のパラメータk＞r＞0とで統合基準を規定しており、最近傍点の少なくとも所定数が同一のクラスターにより共有される場合には、２つのデータ要素は、それら自体が同一のクラスターであるとされる。クラスターを統合するか否か、の決定は、データ・セットの局所的な密度には依存せず、実質的な仕方で近傍を共有する互いの要素の対が存在するか否かに依存することとなる。

【００１７】

JarvisとPatrickの方法（前掲）は、凝集的なものであり、また随伴の連鎖を介した不適正なクラスターを生成してしまう単一リンク法(single-link method)に類似する。より最近に変形例が提案されており、生成されるクラスターの質を変更する試みがなされている。例えば、Guhaら(S. Guha, R. Rastogi and K. Shim, ROCK: a robust clustering algorithm for categorical attributes, Information Systems 25, 5, 2000, pp. 345-366.); by Ertozら(Levent Ertoz, Michael Steinbach and Vipin Kumar, Finding topics in collections of documents: a shared nearest neighbor approach, University of Minnesota Army HPC Research Center Preprint 2001-040, 8 pages, 2001.); Ertozら(Levent Ertoz, Michael Steinbach and Vipin Kumar, A new shared nearest neighbor clustering algorithm and its applications, Proc. Workshop on Clustering High Dimensional Data and its Applications (in conjunction with 2nd SIAM International Conference on Data Mining), Arlington, VA, USA, 2002, pp. 105-115.); Daylight Chemical Information Systems Inc., URLアドレス(http://www.daylight.com/);およびBarnard Chemical Information Ltd., URLアドレス(http://www.bci.gb.com/)を参照されたい。それにもかかわらず、すべての変形例は、依然として凝集アルゴリズムの主要な特徴を示し、随伴性の少ない要素の連結鎖を伴う大きな不適切な形のクラスターの生成を許してしまうものである。

【００１８】

H．次元削減方法の概説
潜在的意味インデキシング(LSI)は、ドキュメントのランク付け問題の次元を削減するためのベクトル空間に基づくアルゴリズムである。これについては、Deerwesterら(Scott Deerwester, Susan T. Dumais, George W. Furnas, Richard Harshman, Thomas K. Landauer, Karen E. Lochbaum, Lynn A. Streeter, Computer information retrieval using latent semantic analysis, U.S. Patent No. 4839853, filed Sept. 15, 1988, issued June 13, 1989; Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, Richard Harshman, Indexing by latent semantic analysis, Journal of the American Society for Information Science, 41, 6, 1990, pp. 391-407.)を参照されたい。LSIは、検索およびランク付けの問題を、著しく低次元の問題へと低下させることで、きわめて大規模なデータベースの検索をより効率的に実行させることができる。他の次元削減手法は、Kobayashiら (Mei Kobayashi, Loic Malassis, Hikaru Samukawa, Retrieval and ranking of documents from a database, IBM Japan, docket No. JP9-2000-0075, filed June 12, 2000; Loic Malassis, Mei Kobayashi, Statistical methods for search engines, IBM Tokyo Research Laboratory Research Report RT-413, 33 pages, May 2, 2001.)によりCOVとして参照される次元削減方法として提案されている。COVでは、ドキュメント・ベクトルの共分散行列を使用して、ドキュメント・ベクトルの射影を行う近似的な削減次元空間を決定している。LSIおよびCOVは、情報検索において互換的な方法であり、いくつかのデータベース、およびいくつかのクエリーについてLSIは、わずかに良好な結果をCOVに比較して与えるものの、それ以外についてCOVは、わずかながら良好な結果を与える。
【発明が解決しようとする課題】
さらに、従来の距離に基づくクラスター検出には、後述する他の不都合があることも知られている。
【００１９】
機械が学習するコンテキストの通常のクラスター化法は、データ・セットにおける主要なグループ化を見出すように設計されている。このためこの方法は、クラスターが高精度で未知のポイントを分類することができるのであれば良好なものと考えることができる。しかしながら、データ・マイニングの分野においては、データの主要クラスターは、多くの場合、ユーザには知られてはおらず、価値ある情報を抽出することができる可能性があるのは、より小さく、マイナーなクラスターである、といえる。分割および凝集に基づく既存のクラスター化技術は、多くの場合にバックグラウンドから小さなデータ・クラスターを分離するには効果的ではない。
【００２０】
さらに別の不都合は、大規模なテキスト・データベースは、典型的には情報検索操作を効率化するために小さな集団に分割されていることにある。この分布は、通常ではデータ・セットの最も大きなクラスターが単一のデータベース内に完全に残されるようにして実行される。しかしながら、主要クラスターに焦点を当てた分割方法は、価値の高いマイナー・クラスターを、いくつかのデータベースの間に分散させてしまうことになる。主要クラスターの他、マイナー・クラスターを同定することは、より効率的にマイナー・クラスターを保存した分割を可能とすることになる。
【００２１】
また、上述したように、クラスター化手段の幾分かのユーザは、しばしばその手段により生成されたクラスターの間の関係を知ることについて興味を有する。クラスターの重なり合いの蓄積を行ない、頂上にデータ・セット全部を含む単一のクラスターがあり、下に向かってより小さなクラスターを配置することにより、階層化クラスター化・アルゴリズムは、上述の要求を満たそうとしている。しかしながら、これらのクラスターのうちの多くは、階層的な組織化の複製物としてのみ存在し、そしてそれら自体では有益な解釈を有するものではない。ユーザは、データ・マイニング手段により報告された各クラスターに対して、ある種の独立した概念的解釈を与えることをまず希望することもある。一旦、意味

を有するクラスターのセットが同定されると、ユーザはクラスター間のいずれかの重なり合いや包含関係を知ることに興味を抱くものと考えられる。

【００２２】

加えて、多次元の設定を与えると、ユーザがクラスターについて容易に理解できるような仕方でデータのクラスター随伴性を表示したり、記述したりすることがきわめて困難である。クラスター・データを閲覧する場合に、ユーザは、クラスターの凝集性や突出性の度合いに対し、一目でアクセスすることができる必要性がある。

【００２３】

検索のハードウエア資源に関しても、クラスター化が概ね望ましいことではあるものの、データ・セットがきわめて大規模な場合には高品質のクラスターを得ることに伴う計算コストの故に、データ・マイニング用途のためには現実的ではないものと考えられてきた。通常のコンピュータにおいて有意義な時間内に大規模なデータ・セットの組織化へのいくらかの知見を提供することができる手段が強く高く必要とされてきた。

【００２４】

上述したように、多くの方法がこれまで提案されているにもかかわらず、高い効率、高い速度と共に高いスケーラビリティを有する情報検索のために好適な新規なデータ構造は、当技術分野で引き続き必要とされてきた。

【００２５】

【課題を解決するための手段】

本発明は、以下において、大規模なテキスト・データベースの情報検索及びデータ・マイニングのための、バックグラウンドに対して高い相互類似の度合いを示す要素（例えばドキュメント）のクラスターの同定に基づいたシステムおよび方法を提案するものである。

【００２６】

本発明においては、クラスターの分布構造は、グラフィカルに表示され、クラスターの質および顕著性として、視覚的に直ちにユーザに対してフィードバックを与えることができる。このシステムは、サイズおよび質といったクラスターの属性に対して自動的にアクセスを可能とする。また、このシステムはユーザに対し、大域的なクラスタリングのための予備的計算を必要とすることなく、クラスターのためのデータ・セットへのクエリーを可能とする。スケーラビリティは、次元削減技術手段、ランダム・サンプリング、および近似的な類似検索を支持するデータ構造の使用により達成される。

【００２７】

本発明は、マイナー・クラスターを保存させつつ、マイナー・クラスターの検出効率を改善することにより、上述した新規な情報検索特性を提供することを可能とする。本発明による新規な情報検索は、ユーザがクラスターを理解する手助けとなるグラフ構造としてクラスター間の相関関係を表現することを可能とする。本発明は、さらに、情報検索の計算の計算スケーラビリティを改善することを可能とするものである。

【００２８】

上述した側面は、クエリー・クラスターを決定するために近傍共有情報を使用する、テキスト・データベースの情報検索およびデータ・マイニングのためのシステムおよび方法により提供される。クラスター化方法は、クエリー要素（データ・セットの要素であっても要素でなくとも良い）と、データ・セットにおけるその近傍との間の相互随伴性のレベルとを評価する。２つの要素の間の随伴性は、これらの要素が共有する最近傍点の比が大きい場合に、強いものとして考察される。従来の共有近傍点の情報を使用する方法とは対照的に、提案する本発明の方法は、クラスター間の随伴性であるコンフィデンス（CONF）およびクラスター内での随伴性であるセルフ－コンフィデンス（SCONF）という新たな特有の概念に基づくものである。

【００２９】

本発明によれば、所定のキーワード・リストから生成され、かつドキュメントに対して与えられる階層構造のノードを構成するドキュメント－キーワード・ベクトルとしてデータベースに格納されたドキュメントの情報検索のためのデータ構造を生成するコンピュータ

・システムであって、前記コンピュータ・システムは、
前記ドキュメント－キーワード・ベクトルに対して階層構造を生成させる部分と前記ノード間にノード間の距離尺度に関連してパッチの関係を生成するパッチ規定部とを含んで構成され、階層構造を使用して決定される類似性を有するノード・グループを生成するための近傍パッチ生成部と、
前記パッチの間の類似性を使用して前記ドキュメント－キーワード・ベクトルのクラスター・データを生成するためのクラスター見積もり部とを含む
コンピュータ・システムが提供される。

【００３０】

本発明における前記コンピュータ・システムは、前記パッチ間のパッチ間コンフィデンス値およびパッチ内コンフィデンス値を決定するコンフィデンス決定部を含み、前記クラスター見積もり部は、前記パッチ間コンフィデンス値に基づいて前記パッチを選択して前記ドキュメント－キーワード・ベクトルのクラスターとすることができる。

【００３１】

本発明の前記クラスター見積もり部は、前記パッチ内コンフィデンス値に応じて前記クラスターのサイズを見積もることができる。

【００３２】

本発明によれば、所定のキーワード・リストから生成され、かつドキュメントに対して与えられる階層構造のノードを構成するドキュメント－キーワード・ベクトルとしてデータベースに格納されたドキュメントの情報検索のためのデータ構造を生成する方法であって、前記方法は、
前記ドキュメント－キーワード・ベクトルに対して階層構造を生成し、格納領域に階層データを格納するステップと、
前記階層構造を使用して決定された類似性を有するノードの近傍パッチを生成し、前記パッチを格納領域に格納するステップと、
前記階層データおよび前記パッチとを読み込んで、前記パッチ間のパッチ間コンフィデンス値およびパッチ内コンフィデンス値を算出し、前記パッチ間コンフィデンス値および前記パッチ内コンフィデンス値を格納領域に対応するリストとして格納するステップと、
前記パッチ間コンフィデンス値および前記パッチ内コンフィデンス値に応答して前記パッチを選択し、前記ドキュメント－キーワード・ベクトルのクラスターとするステップと
を含む方法が提供される。

【００３３】

本発明によれば、所定のキーワード・リストから生成され、かつドキュメントに対して与えられる階層構造のノードを構成するドキュメント－キーワード・ベクトルとしてデータベースに格納されたドキュメントの情報検索のためのデータ構造を生成するための方法をコンピュータ・システムに実行させるためのプログラムであって、前記プログラムは、前記コンピュータ・システムに対して
前記ドキュメント－キーワード・ベクトルに対して階層構造を生成し、格納領域に階層データを格納するステップと、
前記階層構造を使用して決定された類似性を有するノードの近傍パッチを生成し、前記パッチを格納領域に格納するステップと、
前記階層データおよび前記パッチとを読み込んで、前記パッチ間のパッチ間コンフィデンス値およびパッチ内コンフィデンス値を算出し、前記パッチ間コンフィデンス値および前記パッチ内コンフィデンス値を格納領域に対応するリストとして格納するステップと、
前記パッチ間コンフィデンス値および前記パッチ内コンフィデンス値に応答して前記パッチを選択し、前記ドキュメント－キーワード・ベクトルのクラスターとするステップと
を実行させるプログラムが提供される。

【００３４】

本発明によれば、所定のキーワード・リストから生成され、かつドキュメントに対して与えられる階層構造のノードを構成するドキュメント－キーワード・ベクトルとしてデータ

ベースに格納されたドキュメントの情報検索のためのデータ構造を生成するための方法をコンピュータ・システムに実行させるプログラムが記録されたコンピュータ可読な媒体であって、前記プログラムは、前記コンピュータ・システムに対して
前記ドキュメント－キーワード・ベクトルに対して階層構造を生成し、格納領域に階層データを格納するステップと、
前記階層構造を使用して決定された類似性を有するノードの近傍パッチを生成し、前記パッチを格納領域に格納するステップと、
前記階層データおよび前記パッチとを読み込んで、前記パッチ間のパッチ間コンフィデンス値およびパッチ内コンフィデンス値を算出し、前記パッチ間コンフィデンス値および前記パッチ内コンフィデンス値を格納領域に対応するリストとして格納するステップと、
前記パッチ間コンフィデンス値および前記パッチ内コンフィデンス値に応答して前記パッチを選択し、前記ドキュメント－キーワード・ベクトルのクラスターとするステップと
を実行させるコンピュータ可読な記憶媒体が提供される。

【００３５】

本発明によれば、所定のキーワード・リストから生成され、かつドキュメントに対して与えられる階層構造のノードを構成するドキュメント－キーワード・ベクトルとしてデータベースに格納されたドキュメントの情報検索のための情報検索システムであって、前記情報検索システムは、
前記ドキュメント－キーワード・ベクトルに対して階層構造を生成させる部分と前記ノード間にノード間の距離尺度に関連してパッチの関係を生成するパッチ規定部とを含んで構成され、階層構造を使用して決定された類似性を有するノード・グループを生成するための近傍パッチ生成部と、
前記パッチの間の類似性を使用して前記ドキュメント－キーワード・ベクトルのクラスター・データを生成するためのクラスター見積もり部と、
前記見積もられたクラスター・データをディスプレイ手段に表示させるグラフィカル・ユーザ・インタフェイスと
を含む情報検索システムが提供される。

【００３６】

本発明によれば、ユーザ入力クエリーに応答してディスプレイ・デバイス上に見積もられたクラスターをグラフィカルに表示するためのグラフィカル・ユーザ・インタフェイス・システムであって、前記グラフィカル・ユーザ・インタフェイス・システムは、
ドキュメントを格納するデータベースと、
前記データベースに格納された前記ドキュメントについてドキュメント－キーワード・ベクトルを生成すると共に、前記ユーザ入力クエリーに応答してドキュメントのクラスターを見積もるためのコンピュータと、
前記見積もられたクラスターと共に、前記クラスターの間のコンフィデンス関係とクラスター・サイズの階層情報とを画面上に表示するためのディスプレイと
を含むグラフィカル・ユーザ・インタフェイス・システムが提供される。

【００３７】

【発明の実施の形態】

パートＩ．本方法の本質的処理

以下に、本発明をドキュメントの情報検索のコンテキストを使用して説明するが、本発明はこれに制限されるものではなく、本発明のアルゴリズムは、対となった非類似尺度といった距離尺度（三角不等式といった可能な表現）の特性を満足し、各データ要素がキーワードや、アノテーション目的のための他の情報を含む、いかなる用途に対しても適用することができる。上述した用途の１つの例としては、上述した対となった非類似尺度が存在するマルチメディア・データベース（例えば、テキスト、オーディオ、静止画像、グラフィックス・イメージ、グラフィック・ビデオ、ｇｉｆアニメーションなどを含むデータベース）のためのデータ・マイニング・システムを挙げることができる。

【００３８】

本発明の方法の概略的なフローチャートを図１に示す。本発明は、主にテキストに対する適用を用いて説明するが、当業者によれば本発明の方法が、“より近い”（尺度に関連して）要素の対が、“より遠く離れた”要素の対よりもより類似するというどの２つの要素の間においても距離の算出が可能な明確に規定された尺度を含むようにモデル化されたコンテンツを含む、いかなるデータベースに対しても容易に適用することができることは理解されよう。

【００３９】

本発明の方法は、ステップＳ１０から開始し、データベース内のドキュメントがベクトル空間モデルを使用してベクトルへと変換される。次いで、方法は、ステップＳ１２において、データベースに格納されたデータのSASH類似性検索構造を生成する。次いで、ステップＳ１４においてデータベースのすべての要素に対してSASH構造を使用してデータベースの要素に対して、最も類似する要素のリストからなる近傍パッチを算出する。これらのパッチは、その後適切なメモリ領域に格納される。ステップＳ１６においては、セルフ－コンフィデンス値（以下SCONF値として参照する）のリストを、格納されたすべてのパッチについて算出する。これらのSCONF値は、相対セルフ－コンフィデンス値（以下、RSCONF値）を算出するために使用され、このRSCONF値は、各パッチ（それ自身もまたパッチである）の最良のサブセットのサイズを決定するために使用され、クラスター候補が生成される。次いで、本発明の方法は、ステップＳ１８へと進んで、コンフィデンス値（以下、CONF値として参照する。）を使用して冗長なクラスター候補を排除する。方法はその後ステップＳ２０へと進み、最終的なクラスターとされる所望する最小のRSCONF値を有するクラスター候補の選択をさらに行ない、選択されたクラスターを適切なメモリに格納する。方法はさらにステップＳ２２へと進み、ユーザに対してGUIインタフェイスによりコンピュータ・スクリーン上にクラスターの間の相互関係を示したグラフを表示させる。図１に示した方法は、さらに図１に示した各ステップを実行するための複数のサブステップを含んでおり、これらのサブステップについては、以下においてより詳細に説明する。

【００４０】

＜ドキュメント－キーワード・ベクトルの算出＞

ドキュメント－キーワード・ベクトルは、所与のキーワードおよびドキュメントとから、いくつか知られたうちのいかなる技術を使用しても算出することができる。本発明の特定の実施の形態においては、適切な重み付けを使用してドキュメントを数値化させることもでき、数値化の詳細については他の文献（例えば、Saltonら、前掲）に与えられているので、本発明においては説明を行わない。

【００４１】

＜SASHの構成と使用＞

図２は、空間近似サンプル階層、すなわちSASHとして知られているドキュメント－キーワード・ベクトルの階層構造の構築の手法を示した図である。処理は、図１のステップＳ１０の結果を受け取った後、ステップＳ２８から開始し、例えば周知の乱数発生プログラムを使用してSASHのノードとしてベクトルのランダムな割り当てを生成する。レベルには、0～ｈの数字が割り当てられており、各レベルは、その上のレベルの概ね２倍のベクトル・ノードを含む構成とされている。0の数字が付されたレベルは、データ・セットのベクトル・ノードの概ね半数を含んでおり、hの数字が付されたレベルは、トップ・ノードとして参照される単一のノードを含んでいる。SASH構造のトップ・ノードは、コンピュータ・システムのいずれかに含まれる、いかなる乱数発生手段を使用しても決定されている。次いで、ステップＳ３０においてLで参照される階層レベルを、hに初期化する。プロセスは、ステップＳ３２に進み、階層レベルLを1だけ減少させ、ステップＳ３４においてレベルLのノードを、レベルL+1のノードへと、ノード間の距離に依存して連結させる。上述した連結は、レベルL+1のノードが親ノードとなり、レベルLのノードが子ノードとなる。連結は、レベルLのノードからレベルL+1からの最も近い親ノードを選択することにより実行され、その後、これらの親－子ノード対を連結して、それぞれの親ノードが所定数の最も近接する子ノードに連結されるようにして行われる。連結をどのようにして行うかのさら

に詳細については、Houleらによる他の文献（前掲）を参照されたい。プロセスは、ステップＳ３６へと進み、階層レベルが最も低いレベル(0)に達したか否かを判断し、レベル0の場合(yes)、SASHの構築を完了させ、SASH構造を、メモリまたはハードディスクといった適切な記憶領域へと格納する。プロセスは、ステップＳ３８へと続いてノードのパッチを構築させる。レベル(0)ではない場合(no)には、プロセスは、ステップＳ３２へと戻って、ステップＳ３６において肯定的な結果が得られるまで繰り返される。

【００４２】

ステップＳ３８では、格納されたSASH構造が本発明にしたがって用いられ、データベースのすべての要素についてのパッチが生成される。データベースのサブセットRに関連して所与の要素qについて得られたパッチは、所定の基準である類似distに関連してRから抽出されたqの近傍要素のセットである。SASHを構築するための説明している実施の形態においては、データベース内の各ノードは、その階層レベルでラベル付けされている。また、各ノードについてのパッチは、所定の固定されたサイズとされ、かつ、同一のレベル以上のすべてのノードのセットに対して算出される。本発明においては、ノードごとのパッチを構築して格納するのに限定されるわけではなく、また別のノード・セットに関連する追加のパッチを構築して格納してゆくこともできる。

【００４３】

図３は、SASH構造の構成を本発明にしたがって生成されるパッチの構造と共に示した例示的な実施例である。図３に示されるように、パッチにより参照されるベクトル・ノードは、本質的には、基準のベクトル・ノードのレベル以上のSASH階層のいずれかに帰属される。加えて、これらの階層レベルにおけるノードの間から、パッチは、所定の“尺度距離”にしたがって、基準ノードに最も近い複数のノードを含んで構成されている。この基準ノードは、クラスターの大域的な構成を与えるために階層構造のいかなる、またすべてのノードから選択することができ、また本発明の別の実施の形態では、基準ノードは、クエリーに関する基準ノード、すなわち検索されたドキュメントに特に関連したクラスター情報を与えるように、ユーザ入力クエリーを使用して決定することができる。基準ノードは、図３において、☆により示されており、パッチ内のノードは、図３のようにユーザ・クエリーに関連して整列されている。このパッチ構造はまた、詳細には後述するシステム内の適切なメモリ領域に格納されている。本発明においては、これらのパッチはさらに、後述するようにコンフィデンスにより関連づけられる。

【００４４】

＜コンフィデンスの計算＞

本発明の方法では、情報検索と“パッチ・モデル”として本明細書おいて参照する随伴規則の発見との双方を利用したクラスター化の新たなモデルを使用する。パッチ・モデルは、ドメインへの（非）類似近似のある種の尺度にしたがって、データのクラスターがデータ・セットからの要素に基づく近傍クエリーの結果として表現することができるということを仮定する。より、定式化すれば、Sをあるドメインから抽出された要素のデータベースであるものとし、ここで、”dist”を、D上に定義される対の間の上述した定義により与えられる尺度を満足する距離関数であるものとする。ここでさらに、RをSのサブセットとする。所与のいかなるq∈Dのクエリー・パターンに対しても、distによるRから抽出されたqに対するk－最近傍点のセットをNN(R, q, k)とし、下記の条件で選択されたものとする：

もしも、q∈Rの場合には、NN(R, q, 1) = {q}とする。すなわち、もしqがデータ・セットのメンバーであれば、qをその最近傍点として考慮するものとする。

【００４５】

NN(R, q, k-1) ⊂ NN(R, q, k)（すべての1＜k≦|R|に対して）とする。すなわち、qのより狭い近傍は、より広い近傍に厳密に包含されるものとする。

【００４６】

これらの条件は、qがRにおける１つ以上のk－最近傍点セットを有する可能性を考慮するためである。固有的に決定されるセットNN(R, q, k)は、qのk番目のパッチとして参照さ

れる（Rについて）か、またはqのパッチのうちの１つとして参照される。
【００４７】
図４は、データベースのパッチ（７－パッチ、１２－パッチ、および１８－パッチ）の構成を示した図である。破線で示した円は、ドキュメント空間の全体を表す。
【００４８】
ここで、R内において可能性のある２つのクラスターが、２つのパッチ$C_i = NN(R, q_i, k_i)$ and $C_j = NN(R, q_j, k_j)$として表されるものとする。C_jとC_iとの間の関連性は、AgrawalおよびSrikantら（前掲）により提案された随伴ルール発見に類似するナチュラル・コンフィデンス尺度にしたがって評価することができる。
【００４９】
$CONF(C_i, C_j) = |C_i \cap C_j|/|C_i| = |NN(R, q_i, k_i) \cap NN(R, q_j, k_j)|/k_i$.
すなわち、コンフィデンスは、C_iを構成する要素であって、またC_jを構成する要素に比例するものとして表現することができる。コンフィデンス値が小さい場合には、候補C_jは、C_iに対してわずかな影響しか与えないか、またはまったく影響を与えない。一方で、その割合が大きいと、C_jは、C_iに強く関連し、これを包含する可能性もある。
【００５０】
図５は、それぞれ８ベクトルおよび１０ベクトルを有するクラスターAとBとに対する関数CONFの本質を示した図である。２つのベクトルが、共にAとBとの交わりに存在し、したがって関数CONFが、A,Bの順でパッチに対して適用される場合、すなわちCONF(A,B)は、0.25、すなわち25%を与える。関数が(B,A)の順で適用される場合、すなわちCONF(B,A)の場合には、結果は0.2すなわち20%として与えられる。関数CONFは、データベースに共に含まれるサンプルから抽出される２つのパッチに対して適用することができる。
【００５１】
コンフィデンスの尺度はまた、共有近傍点の距離尺度の例としても捉えることができる。しかしながら、共有近傍点の情報への使用方法は、凝集クラスター化方法の使用方法とは本発明においてはきわめて異なるものである。凝集法は、上述した尺度を２つのパッチを統合するべきかどうかを判断するために使用するが、本発明において提案される方法は、尺度を２つの判断するべきパッチの間の随伴性のレベルの質を判断するために使用する。
【００５２】
＜クラスター内随伴性の計算＞
パッチ内における随伴性の自然な評価はまた、コンフィデンスの考え方で可能である。ここで、$C_q = NN(R, q, k)$を、パッチのクラスター候補とする。ここで、パッチC_qを構成するパッチは、$v \in C_q$のすべての要素について$C_v = NN(R, v, k)$の形のパッチのセットであるものと定義する。C_qが高い内部随伴性を有している場合には、C_qと、その構成パッチとの間に強い関連性を充分に予測することができる。他方で、内部随伴性が低い場合には、C_qとその構成パッチとの間の関連性は弱いことが期待される。したがって、セルフ－コンフィデンスにおけるパッチのクラスター候補における内部随伴性が得られ、これを構成パッチに関連して候補パッチの平均コンフィデンスとして定義することができる。
【００５３】
$SCONF(C_q) = (1 / |C_q|) * \Sigma_{v \in Cq, |Cv|=|Cq|} CONF(C_q, C_v)$
$= (1 / k^2) * \Sigma_{v \in Cq} |NN(R, q, k) \cap NN(R, v, k)|$.
１のセルフ－コンフィデンス値は、クラスターのすべての要素にわたり完全な随伴性があることを示すが、これが０に近づくにつれて内部随伴性は小さくなるかまたはまったく随伴性のないことを示す。
【００５４】
＜クラスター間のコンフィデンスを使用したクラスター境界の決定＞
ここで、対象としているノードqが、我々が評価を希望するR内のあるクラスターに随伴するものと仮定する。セルフ－コンフィデンスの考え方を使用して、プロセスは、関心のある範囲$a \leqq k \leqq b$でこのクラスターを最も良く記述するqを元にしたk－パッチを決定する。
理想的なパッチは、クラスター要素を主要に含むものであり、相対的に高いセルフ－コン

フィデンスを有するものであるが、パッチがより大きくなると、クラスターの外部の多くの要素を含み、比較的低いセルフ－コンフィデンスを有することが予測される。２つのパッチに対する評価について焦点を当てる。サイズがkの内パッチ$C_{q,k}$ = NN(R, q, k)は、候補パッチのクラスターを示し、サイズがφ(k)＞kの外パッチ$C_{q,\phi(k)}$ = NN(R, q, φ(k))は、内パッチの判定に対する適切さに対する局所的なバックグラウンドを与えるものとする。

【００５５】

所定の選択したkに対して、外パッチのそれぞれの要素の近傍点セットを評価する。近傍点の対(v,w)を考えるものとし、vが外パッチに属し、wが外パッチNN(R, q, φ(k))の要素であるものとする。vはまた内パッチに属し、wが内パッチNN(R, v, k)の要素であるものとすると、ここで、(v,w)は、内側近傍点の対である。

【００５６】

ここで、wが、外パッチの要素であれば、対(v,w)は、外パッチのセルフ－コンフィデンスに寄与し、qを元にしたクラスターのデスクリプターとして内パッチの選択を妨げることになる。wがまた、内パッチの要素である場合には、(v,w)は内側対であり、したがってこの対は、内パッチのセルフ－コンフィデンスに寄与することにより、vとqとの間の随伴性を強化することになる。

【００５７】

本質的には、qを含むクラスターをもっとも良く記述するk－パッチは、下記のようにして得られる。

ｉ）内パッチのセルフ－コンフィデンスに寄与する内側ペアを高い割合で含むこと、および

ｉｉ）外パッチのセルフ－コンフィデンスに寄与しない近傍点の対（内側対である必要はない）を高い割合で含むこと、である。

【００５８】

ｉ）の種類の高い割合は、k－パッチ内での随伴性の高いレベルを示すが、ｉｉ）の種類の高い割合は、局所的なバックグラウンドとの間で高い差別性を有していることを示すものである。両方の考察は、等しく重要なので、これらの割合は、別々に考慮されるべきである。上述した考察は、a≦k≦bの範囲の選択されるすべてのｋにわたり、２つの割合SCONF($C_{q,k}$)および1 - SCONF($C_{q,\phi(k)}$)の和が最小となるようにして考慮される。

【００５９】

相対セルフ－コンフィデンス(RSCM)の問題は、後述するようにして定式化することができる。

【００６０】

$\max_{a \leq k \leq b}$ RSCONF($C_{q,k}$, φ),

上記式中、

【００６１】

RSCONF($C_{q,k}$, φ) = SCONF($C_{q,k}$) - SCONF($C_{q,\phi(k)}$)
= SCONF(NN(R, q, k)) - SCONF(NN(R, q, φ(k)))

であり、RSCONFは、k－パッチ$C_{q,k}$のRおよびφに関する相対セルフ－コンフィデンスとして参照される。最大を与えるk－パッチは、この領域にわたるqのクエリー・クラスターとして参照されるものとなる。RSCMは、最尤見積もり(maximum likelihood estimation)の形式として考えることができ、これは、近傍点の対がクエリー・クラスターとして内パッチを選択することを指示するか、または指示しないかを分類するものである。

【００６２】

図６は、クエリー要素のパッチ・プロファイルの一部として本発明の方法に含まれるSCONFの計算を実行するための擬似コードのサンプルを、近傍点リスト、NN(R, q, φ(b)) and NN(R, v, φ(b))がすでに、すべてのvについて利用可能であるものと仮定して示した図である。SCONF(NN(R, q, k))を直接的に算出するのではなく、SCONF(NN(R, q, k))は、１つのアイテムによりパッチを展開し、差分的に得るべく計算を行うことによって、SCONF(

NN(R, q, k-1))から得られる。

【００６３】

本発明においては、説明するRSCM方法は、外パッチのサイズがk（kは整数である。）の値に依存する仕方で多くの変形例を可能とする。φ(k)=2kとする簡単な選択が、内パッチに関連して外パッチの要素であるか、要素でないかということについて最良のバランスを与えるものの、φ(k)の選択にあたっては他の考察も影響を与える。例えば、境界を明確にするための値を算出するためのコストからは、最大パッチ・サイズとしてm＜2bを使用することが好ましい。この場合、外パッチ・サイズは、φ(k)＝min{2k,m}とし、外パッチ・サイズと内パッチ・サイズとの間のm/bの最小値が、依然として１よりも実質的に大きくなるように選択することができる。

【００６４】

本発明においては、RSCM法の設定は、内側のクラスターの随伴性が外側の差と同程度に重要であることを仮定する。しかしながら、本発明においては異なる重み付けを相対セルフ－コンフィデンスの値への内側及び外側の寄与に対して与えることもできる。すなわち、下記式、

【００６５】

$RSCONF'(C_{q,k},\phi) = w_1\ SCONF(C_{q,k}) - w_2\ SCONF(C_{q,\phi(k)})$,

の関数を最大化する代わりに、重み付け０＜w_1および０＜w_2の実数値を選択することもできる。

【００６６】

この段階で、格納されたそれぞれのパッチ$C_{q,m}$ = NN(R, q, m)は、1≦k≦mの範囲のkのすべての値に対して、$C_{q,m}$のそれぞれの副パッチ$C_{q,k}$=NN(R, q, k)それぞれに対する、セルフ－コンフィデンス値のリストを伴っている。SCONFとして以下に参照するデータ構造を図７に示す。このデータ構造は、ハードディスクまたはメモリといった適切な記憶手段内に格納され、本発明のクラスター選択機能により参照される。

【００６７】

本発明のさらなる別の変形例は、計算のコストを節約するものである。RSCONF値を計算するコストは、外パッチのサイズが増大するに連れて二次的に増加する。このコストは、データ・セット全体への直接的なRSCM法の実際的な適用において見出されるクラスターのサイズを制限してしまうことになる。しかしながら、これらの制限は、ランダム・サンプリング技術を使用して回避することができる。RSCM問題を解くための範囲をa≦k≦bに制限するように調整することで大きなクラスターに適応させる代わりに、サイズを変化させたデータ・サンプルを集めるのに対して、固定された範囲においてパッチ・サイズの検索することができる。

【００６８】

上述した変形例を理解するために、ある大きな値cについてR ⊆ Sの等しくランダムなサンプルと仮説的なクエリー・クラスターNN(S, q, c)との間の関係を考察する。NN(S, q, c)とRとの交わりは、パッチNN(R, q, k)を生成する。ここで、k = |NN(S, q, c) ∩ R|である。パッチNN(S, q, k)は、Rにおけるqのクエリー・クラスターとしてのサンプルR-NN(S, q, k)の選択値に関連して、NN(S, q, c)のプロキシを提供し、これは、全体のデータ・セットに関してqに対してのクエリー・クラスターとしてのNN(S, q, c)の適切さの指標とすることができる。

【００６９】

a≦k≦bの場合には、プロキシ・パッチをRSCM法により見積もることができる。それ以外には、kは、aとbとの間にはなく、パッチを見積もることができない。未知の“真の”クラスター・サイズcについては、評価されないプロキシ・パッチの可能性の限界は、MotwaniおよびRaghavan(R. Motwani and P. Raghavan, Randomized Algorithms, Cambridge University Press, New York, USA, 1995.)の標準的なChernoff境界技術を使用して得ることができる。

$E[k] = \mu = c\ |R| / |S|$

$Pr[k < a \mid c] \leqq e^{-\mu} [e\mu / (a-1)]^{a-1}$
$Pr[k > b \mid c] \leqq e^{-\mu} [e\mu / (b+1)]^{b+1}$.
これらの境界は、近似的なサイズの収集に加え、aおよびbの近似的な値を選択するためのガイドとして使用することができ、充分に大きなcについての所望する確率について、少なくとも一つのサンプルを、少なくとも１つのプロキシ・パッチがaとbとの間のサイズとなるようにすることができる。
【００７０】
例示的な実施例としてh、均等なランダム・サンプルである$|R_i| = |S| / 2^i$ for $i \geqq 0$である$\{R_0, R_1, R_2, ...\}$について考察する。ここで、$NN(R_i, q, k_i)$を、$NN(S, q, c)$のプロキシ・パッチとする。ここで、cは、少なくとも25以上であることが保証された未知の数である。a=25,b=120の制限を選択する場合には、少なくとも一つのサンプルR_iに対して、i番目のプロキシ・パッチの所望するサイズ$\mu_i = E[k_i]$は、$44 \leqq \mu_i \leqq 88$の範囲内に存在する。上述した境界を適用すると、μ_iがこの範囲に制限される場合には、$NN(R_i, q, k_i)$がRSCM法により評価できない確率は、低いと見積もられる（0.004285より小さい）。
【００７１】
言い換えれば、この範囲とサンプルとを選択すると、まったく評価されないプロキシ・パッチの確率は、1/233以下とすることができる。このエラー境界は、まったく従来と同様のものであり、失敗する確率は、実際的にはきわめて低いものである。
【００７２】
RSCM法がプロキシ・パッチ$NN(R_i, q, k_i)$を、クラスター評価として提供する場合であっても、Sにおけるクラスターに対応するサイズを推定するための正確な方法はない。しかしながら、後述する最大蓋然性評価の原理、$E[k] = k_i$での$c = E[k] |S| / |R_i|$の値は、真のクラスター・サイズの自然な見積もりを構成する。サンプルR_iについて見積もることができる最も小さなクラスター・サイズは、したがって、$(a |S|) / |R_i|$となる。
【００７３】
同一のクラスターがいくつかの異なるサンプルにわたって検出される場合には、真のクラスターのサイズの見積もりは、妥当でないことに留意されたい。にもかかわらず、実際には大きなRSCONF値は概ね、クラスターのサイズが正確に決定されない場合でさえも、クラスターの存在を示す信頼性のある指標となることが判明した。
【００７４】
＜要素の再分類＞
本発明においてはさらに、共通のクエリー要素へのメンバーの近接性の利点により、RSCM法で生成されたクラスターは、テキスト・マイニングにおいて望ましい特徴である凝集クラスター化法により生成されるクラスターよりも、より凝集性を有する傾向にあることが見出された。現実的には、クエリー・クラスターは、ペア間の距離尺度よりは、球形の形状に向かって行くことになる。
【００７５】
RSCM問題のためのクラスター・パッチの解は、全体として同様のクエリー要素に基づく他の方法に比較して高いレベルの相互随伴性を示すものの、上述したクラスターの要素は、クエリー要素それ自体に伴なわれても、伴われなくとも良い。むしろ、クエリー要素は単に、相互に良好な随伴性を有する近傍データを見出すためのスタート点を与えるにすぎない。クエリー要素がその随伴クラスターに比較してアウトライアーであるか、または見出されたクラスターの実質的な部分がアウトライアーを構成するように思われるような別の状況の場合には、第２のクラスター化基準にしたがって外パッチの要素を再評価することが効果的である。このような再評価は、より球形である凝集クラスターの発見を可能とする。
【００７６】
多くの方法によりクエリー・クラスターの近くで要素を再分類することが可能となる。図８には、このような変形例を説明した疑似コードの記述を示す。図８に示したプロセスは以下のようにして実行される。

ｉ）元のクエリー・クラスターを決定した内側のk－パッチを与え、すべての対応する外側パッチの要素をクエリー要素において共有されるk－近傍点の数にしたがって再評価を行う。具体的には、すべてのv ∈ NN(R, q, φ(k))を、コンフィデンス値CONF(C_q, C_v)にしたがって、最高から最低へとランク付けを行うqからの距離に応じて連結は破られることになる）。ここで、C_q = NN(R, q, k)であり、C_v = NN(R, v, k)である。
ｉｉ）最も高い方からk個の要素を、新たな調整されたクラスターとして報告する。これとは別に、外パッチの要素の全体のランキングを報告しておき、ユーザが最終的なクラスターの要素の判断を残しておくこともできる。この方法においては、元の内パッチの外側の要素であって依然として外パッチの内側にある要素が新たなクラスターに含まれるものとして選択され、近傍点間の元々のパッチの要素を多く有することが示される。

【００７７】
＜クラスターの選択＞
提案される全体的なクラスター化の方法であるPatchCluster機能は、均等なランダム・サンプル{R_0, R_1, R_2, ...}の収集から得られるクエリー・クラスター相互関係(query cluster relationship:QCR)のグラフを構築する。ここで、すべてのj＜iおよび|R_i| = ceil(|S|/2^i) for 0 ≦ i < $\log_2$ |S|について、R_i ⊂ R_jである。このグラフは、コンフィデンスに類似するいくつかのパラメータに依存し、これを支持するための随伴規則生成のしきい値は、下記の通りである。
ｉ）（クラスター品質）クラスターの相対セルフ－コンフィデンスについての最小しきい値α；
ｉｉ）（クラスター差）概ね同一のサイズの２つのクラスターの間のコンフィデンスの最大のしきい値β（共通のサンプルRiから得られる）；
ｉｉｉ）（随伴品質）随伴クラスターの間のコンフィデンスの最小のしきい値γ（共通のサンプルから得られるものである必要はない）；
ｉｖ）（随伴スケール）２つの随伴するクラスターの間のスケールの差の最大のしきい値δ（すなわち、差|i-j|であり、R_iとR_jとは、クラスターを導出するサンプルである。）
図９は、本発明に使用されるPatchCluster法を記述した擬似コードのサンプルである。基本的なQCRの構成手法は、下記のようにまとめることができる。

【００７８】
１．QCRノード・セット：
それぞれ0 ≦ t < $\log_2$ |S|について、サンプルR_tの要素から、R_tの異なるクエリー要素に基づくそれぞれC_i = NN(R_t, q_i, k_i)であって、a≦|C_i|≦bのクラスターからなるクエリー・クラスターQC_t = {C_1, C_2, ..., $C_{|Rt|}$}を収集する。RSCONFにしたがって使用できるクエリー・クラスターの中から、可能な限りQC_tの要素を選択する。この場合、i＜jであれば、RSCONF(C_i)≧RSCONF(C_j)とし、このための条件として、下記の２つの条件を適用する。
ｉ．（クラスター差）すべての1≦i＜j≦m_tについて、max {CONF(C_i,C_j),CONF(C_j,C_i)}＜βとし；
ｉｉ．（クラスター品質）すべての1≦i≦|R_t|について、RSCONF(C_i)≧αとする。
これらのクラスターは、レベルｔでのQCRグラフにおけるノードとなる。

【００７９】
２．QCRエッジ・セット：
QCiにおけるi≦j≦i＋δであるような異なるクエリー・クラスターC_i = NN(R_i, q_i, k_i)およびC_j = NN(R_i, q_i, k_i)のそれぞれのペアについて、C'$_j$ = NN(R_i,q_j,$2^{j-i}k_j$)として、max {CONF(C_i,C'$_j$),CONF(C'$_j$,C_i)}≧γの場合に、指示されたエッジ(C_i, C_j),および(C_j,C_i)を、QCRグラフに挿入する。CONF(C_i,C'$_j$)およびCONF(C'$_j$,C_i)の値をエッジ(Ci, C_j),および(C_j,C_i)の重み付けとして適用する。

【００８０】
グラフのそれぞれのレベルは、クラスターのセットのレベル、aおよびbとに依存する領域内にある見積もりサイズからなる、粗い断面として見ることができる。それぞれのスライ

スにおいて、候補は、それらのRSCONF値にしたがって余すことなく選択される。また、新たな候補は、それらが以前に許容された候補から充分に異なる場合にのみ許容される。

【００８１】

本発明においては、共通のレベルにおける重複したクラスターの生成は排除されるものの、重複したクラスターが、異なるレベルにおいて生成する場合には許容される。QCRグラフは、このためわずかな時間だけのみ所定のクラスターを含むことができる。いくつかの生成されたレベルにおける同一のクラスターの存在は、共通する概念を共有する２つのクエリー・クラスターが、重なり合っているものと判断されるので、実質的に構造の連結性を向上させることとなり、エッジにより連結されることになる。図９は、本発明において“PatchCluster法”として参照される、パッチを排除するための擬似コードのサンプルを示した図である。

【００８２】

αの値を低下または増加させることにより、ユーザは、グラフに出現するクラスター・ノードの数を増加または減少させることができる。βの値を増加させることはまた、クラスターの数を増加させることになるが、これは、所定のサンプルから１つ以上のクラスターにより個別的な概念が共有されてしまうというリスクをもたらす。クラスター化の規則を誘導する目的のため、高い連結性のあるグラフとすることが好ましい。QCRグラフの２つの随伴クラスターの間のスケールの差に対する最大しきい値δは、後述する理由から小さな固定値とするべきである。

【００８３】

PatchCluster法の別の変形例は、クラスターの数の制御を含むものである。上述したように、生成されるクラスターの数は、報告されたクエリー・クラスターの相対セルフ－コンフィデンスしきい値αを特定することによって制御される。その代わりに、ユーザは、それぞれのデータ・サンプルについて別々のクラスターの数を決定するオプションが与えられても良い。所定のレベルｔに対して、このことは下記のようにして実行される。

ｉ）レベルtから得られたクエリー・クラスターの相対セルフ－コンフィデンスについて最小のしきい値α_tを特定する。

ｉｉ）レベルtから得られるクエリー・クラスターの絶対数に付いての最大のしきい値を特定する。

【００８４】

クラスターの数についてのしきい値が与えられると、クラスターの選択は、所望するクラスターの数が得られた場合や、すべての候補を考慮した場合にどれが最初に発生しても停止される。

【００８５】

PatchCluster法では、PatchCluster/RSCMのパラメータは、上述した方法すなわちアルゴリズムが実装されるシステムに依存して決定することができる。決定されるパラメータは、下記のものを挙げることができる。

【００８６】

＜内パッチ・サイズの範囲＞

内パッチのサイズの範囲[a,b]は、この方法によって任意に大きなクラスターでも発見することができるように選択されるべきである。aとbとを、失敗の確率を分析してより精度良く選択することも可能であるが（上述したChernoff境界を使用する）、下記の汎用的な原理を適用することができる。パラメータは、サンプルのサイズが小さいことによる変動を克服するに充分なだけ大きくする必要がある。変数aは、20以上であることが要求される。パラメータbは、得られるレベルについてターゲットとするクラスター・サイズの範囲が実質的なオーバーラップを有しているように選択される必要がある。このことは、bをaの約3倍以上とすることにより達成されることが見出された。

【００８７】

＜最大パッチ・サイズ＞

また、最大パッチ・サイズは、効率の理由からできるだけ小さく選択することが必要であ

る。しかしながら、最大パッチ・サイズは、実質的にbよりも大きくなるように選択される必要があり、φ(b)=2bとして選択することが理想的である。しかしながら、φ(b)=1.25bを選択しても、また良好な結果を与えることができることが見出された。本発明の最良の実施の形態においては、a=25、b=120、φ(k)=min{2k,150}とすることで、多くのデータ・セットについて満足する結果が得られるので、好ましいことが見出された。

【００８８】

共通のサンプルからのいかなる２つのクラスターの間のコンフィデンスについての最大のしきい値βは、データ・セットにかかわらず概ね0.4に設定される必要がある。実験によれば、共通のサンプルから重なり合うクエリー・クラスターは、近似的にまたはわずかに重なり合ういずれかの傾向にあることが示された。PatchCluster法により与えられるクラスター化は、bの正確な選択には比較的感受性は高いものではない。

【００８９】

＜しきい値δ＞

QCRグラフの２つの随伴するクラスターのスケールにおける差の最大のしきい値δは、いくつかの理由から常に小さな固定値とされる必要がある。大きな値は、最大のクラスターが圧倒的な数のきわめて小さなクラスターへと連結されるグラフを与える。この結果、QCRグラフは、ユーザを誘導するのについて、きわめて困難性をもたらす。レベル0からのすべてのクエリー・クラスターについては、式$NN(R_i, q_j, 2^{\delta} k_j)$の近隣を算出する必要がある。スケーラビリティを保証するために、δは、小さな定数として選択されなければならない。この値を、実験においてはδ=4とすることで、概ね最大で2^4から2^5の程度の差のサイズのクラスター間で随伴エッジを生成することができた。δについて上述のように選択することがきわめて好ましい。

【００９０】

後述するパラメータは、ユーザの特定の要求にしたがって設定することができる。

（ａ）クラスターの相対セルフ－コンフィデンスに対する最小のしきい値α（または、これとは別に、それぞれのサンプル・レベルに対して最小のクラスターの相対セルフ－コンフィデンスおよび／または所望するクエリー・クラスターの最大数）。0.1≦α≦0.2の範囲の値が好ましい。値をより小さくすると、クラスターの数が大きくなる。

（ｂ）QCRグラフにおける随伴クラスターの間のコンフィデンスについての最小のしきい値γ（共通のサンプル・レベルから得られる必要はない）。0.15≦γ≦0.2の範囲の値が好ましい。小さな値は、グラフのエッジの数をより多くする。

（ｃ）それぞれのクエリー・クラスターに適用されるキーワード・ラベルの数。

【００９１】

PatchCluster法のさらなる変形例は、正確な近傍パッチを算出するのではなく、近似的な近傍点パッチを計算するものである。PatchCluster法により実行される近傍点計算は、データ要素の数が多く、正確な近傍の情報を探索する場合には、高コストなものとなる。この方法の効率を改善するためには、近似的な近隣情報で置き換えることができる。SASHといった類似検索構造は、高い精度でシーケンシャル検索よりも高速にこの情報を生成するために使用される。

【００９２】

さらなるPatchCluster法の変形例は、ドキュメント－キーワード・ベクトルおよびキーワード・ベクトルの次元削減を実行するものである。

【００９３】

基本的なPatchCluster法は、図９に示すようにテキスト・データに適用する場合、ドキュメントが適切な重み付けを使用してベクトルとしてモデル化されていることを仮定する。キーワード空間が大きくともドキュメントあたりの平均の数が小さい場合には、ベクトル間の距離計算は、ベクトルが本質的、すなわち、非ゼロのエントリだけで、しかもその位置が格納されているのであれば効率的に実行される。しかしながら、ドキュメントあたりのキーワードの平均の数が大きな場合は、距離比較のコストを制限するためにしばしば次元削減が実行される。ドキュメントあたりの元の平均のキーワード数にかかわらず、LSI

またはCOVといった次元削減技術で、所望される場合にはクラスター化の前に適用することができる。発明の実施の形態の欄において与えられる実験結果は、次元削減の使用および非使用について、それぞれの効果を示すものである。

【００９４】

PatchCluster法のさらに他の変形例は、QCRグラフの簡単化に含ませることができる。PatchCluster法により与えられるQCRグラフは、多数のクラスターの対についての随伴情報を含む。しかしながらこの情報は、時として簡略化することなしにユーザを容易に誘導するためには高い密度を与えることとなる。グラフを適切に簡略化することを可能にするいくつかの方法は、下記の通りである。

【００９５】

ｉ）（レベル間の過渡的なエッジを排除する）例えば、グラフが$u<v<w$であるクラスター・ノード$C_1 = NN(R_u, q_1, k_1)$、$C_2 = NN(R_v, q_2, k_2)$、$C_3 = NN(R_w, q_3, k_3)$および随伴エッジ(C_1,C_2)、(C_2,C_3)、(C_1,C_3)を含むものと仮定する。エッジ(C_1,C_3)は、ユーザが(C_1,C_2)および(C_2,C_3)を介してC_1からC_2へと依然として誘導されるのでユーザから隠すことができる。

ｉｉ）（類似クラスターの省略）２つのクラスター$C_1 = NN(R_u, q_1, k_1)$および$C_2 = NN(R_v, q_2, k_2)$が、$CONF(C_1,C_2)$および$CONF(C_2,C_1)$の両方が充分に高い値を有するためにきわめて類似するものと考えられるときには、それらのうちそれぞれのノードを省略することができる。２つのノードのうち、一方のノードを保持し、他のノードを削除する（保持されるクラスター・ノードは、より高いRSCONF値を有すること、またはサイズが大きいことといった種々の仕方で選択することができる。）。削除されたノードを含むいかなるエッジであってもその後、保持されたノードに帰属され、例えばC_1が保存され、C_2が削除された場合には、エッジ(C_2, C_3)が、(C_1,C_3)に変換される。その結果いかなる重複したエッジでも削除されることになる。当然のことながら、他の簡略化方法も本発明においては採用することができる。

【００９６】

＜グラフィカル・ユーザ・インタフェイス：クラスターのラベリング＞

検索されたクラスターを表示させるための有用なグラフィカル・ユーザ・インタフェイスを提供するために、クエリー・クラスター・ラベリングおよび同定の問題を、テキスト・データおよびベクトル空間モデルの文脈においてここでは検討する。クエリー・クラスターは、単一のクエリー要素の限定された近傍内に存在するので、表示ベースのクラスター化のために、クラスターのデスクリプターとしてクエリーを使用することを試みる。しかしながら、クエリー要素は、クラスターの最良の代表となる必然性はなく、実際には、全体を記述する適切なクラスターの要素がない可能性もある。

【００９７】

クラスターに対してラベルを帰属させる１つの通常の方法は、ドキュメント・ベクトル・モデルにおいて使用されるような用語重み付け法にしたがい、クラスターのドキュメント内においてもっとも頻出する用語のランク付けされたリストを使用することである。用語にはそれぞれクラスターのすべてのドキュメント・ベクトルの全部にわたる用語重み付けに対応する合計（または平均に等価）に等しくなるようなスコアが与えられる。もっとも高いスコアを与える用語の所定数をランク付けして、ラベルとしてユーザに提供する。

【００９８】

COVやLSIといった次元削減技術を使用する場合には、元の次元削減されないドキュメント・ベクトルは、もはや利用可能ではないか、または格納および検索するのにはコスト的に高くつく。それにもかかわらず、元のベクトルなしに意味のある用語のリストは依然として抽出することができる。まず、i番目の用語は、$z_{i,j} = 1, \text{ if } i = j, \text{ and } z_{i,j} = 0$などのように、元のドキュメント空間の単位ベクトル$z_i = (z_{i,1}, z_{i,2}, \ldots, z_{i,d})$とすることができる。ここで、クエリー・クラスター$NN(R, q, k)$に属するドキュメント・ベクトルの平均をμとする。この記述を使用して、ｉ番目の用語についてのスコアは、簡単に$z_i \cdot \mu$として記述することができる。しかしながら、$\| z_i \| = 1$であり、μは定数なの

で、これらのスコアにしたがう用語のランク付けは、下記の尺度にしたがうランク付けとなる。

【００９９】

$z_i \cdot \mu / \| \mu \| = \mathrm{cosangle}(z_i, \mu) = \cos \theta_i$,

上記式中、θ_iは、ベクトルz_iとμとの間の角度を表す。

【０１００】

次元削減と共に、元の空間のベクトルvとベクトルwとの間の対の距離cosangle(v, w)は、次元削減された空間のvおよびwにそれぞれ等価な、v'およびw'を使用してcosangle(v', w')として近似することができる。したがって、ベクトルz_i およびμの次元削減された相手方をそれぞれz'_iおよびμ'として、cosangle(z_i,μ)は、cosangle(z'_i, μ')で近似することができる。また、cosangle(z'_i, μ')は、クエリー・クラスターの次元削減されたベクトルの平均であるμ"を使用して、cosangle(z'_i, μ")で近似することができる。ベクトルz'_iは、すべての$1 \leqq i \leqq d$に対してあらかじめ計算することで、用語のランク付けされたセットが効果的に次元削減されたアトリビュート・ベクトルを集めることでμ"に基づいて最近傍検索の手段により、生成することができる。dは、典型的には相当に小さいので、このような検索のコストは、クラスターの生成のコストに比較にして無視できるものとなる。

【０１０１】

次元削減したクラスターのラベリング法は、下記の通りである。

ｉ）i番目のアトリビュートについて、すべての$1 \leqq i \leqq N$に対して、次元削減アトリビュート・ベクトル$z_i = (z_{i,1}, z_{i,2}, \ldots, z_{i,d})$を計算する。ここで、Wを次元削減アトリビュート・ベクトルとする。

ｉｉ）$\mu'' = S_{v \in NN(R, q, k)}$ vを計算する。ここで、vおよびqは、次元削減されたデータのベクトルである。

ｉｉｉ）λをクラスターの所望するラベルの数として、Wにおけるμ"のλ－最近点を、cosangle尺度の値を減少させながら計算する。

ｉｖ）クラスターのラベルとしてλ個の近傍のランク付けされたリストに対応するアトリビュートを取得する。これとは別に、cosangleそれ自体の値をユーザに対して表示させることもできる。また、さらに別に近似的な最近傍を、SASH、または別の類似検索方法を使用して生成されたように使用することもできる。

【０１０２】

パートＩＩ．情報検索のためのシステム

図１０は、本発明のアルゴリズムが実装されるシステムを示した図である。図１０に示されるように、システムは概ねコンピュータ１０と、ディスプレイ装置１２と、キーボード１４といった入力デバイスと、マウス１６といったポインタ・デバイスとを含んで構成されており、本発明にしたがい、ユーザが情報検索のためのクエリーを入力することができる構成とされている。コンピュータ１０は、また、検索されるドキュメントを格納するデータベース１８を管理していると共に、格納されたドキュメントをデータベースから検索する。コンピュータ１０は、LAN、またはWANまたはインターネットといった通信ライン２０へと、Ethernet（登録商標）、光通信、またはADSLにより、ハブまたはルータ２２を介して接続されている。

【０１０３】

通信ライン２０が、企業の複数のサイトを相互接続するLAN/WANおよび／またはインターネットであるものとすれば、コンピュータ１０は、クライアントおよび／またはユーザからの入力クエリーが伝送され、情報検索を実行するサーバとされても良い。サーバ・コンピュータ１０は、本発明のアルゴリズムによりドキュメントを受信したクエリーに関連して検索し、検索された結果を、クエリーを発行したクライアントへと返信する。当然ながら本発明は、上述した情報検索をインターネットを介した有料の情報サービスとして登録クライアントへと提供することもできる。これとは別に、コンピュータ１０は、特定の用途に対して好適となるように調整されたスタンドアローンのシステムとすることもできる

。

【０１０４】

図１１は、コンピュータ１０内に実装される機能ブロックの詳細図である。コンピュータ１０は、概ねベクトル生成部２４と、SASH生成部３６と、SCONFリストを生成するためのコンフィデンス決定部３８と、パッチ規定部２６とを含んで構成されている。ベクトル生成部２４は、キーワード・リストまたは所定の規則を使用して、データベース１８に格納されたドキュメントから、ベクトル生成を実行し、生成されたドキュメント－キーワード・ベクトルを、対応するドキュメントへの適切なリンクまたは参照を可能とするように、メモリ、またはデータベースといった適切な記憶領域へと格納する。SASH生成部３６およびパッチ規定部２６は、本発明における近傍パッチ生成部３４を構成する。

【０１０５】

SASH生成部３６は、図２において示したアルゴリズムを使用してSASH構造を構築し、生成したSASH構造をメモリ領域３０へと格納させて、詳細には後述する処理を可能としている。SASH生成部３６は、コンフィデンス決定部３８が、CONF、SCONF、RSCONFといったコンフィデンス値を算出することができるように利用できる構成とされており、上述したアルゴリズムにしたがってSCONFリストを生成している。生成されたパッチ・データおよびコンフィデンス値は、図１０に示されるように、ハードディスク３２へと格納される。

【０１０６】

クエリー・ベクトル生成部４６は、検索条件およびクエリー・キーワードを受け取り、対応するクエリー・ベクトルを生成し、生成されたクエリー・ベクトルを適切なメモリ領域へと格納させる。上述したクエリーとしては、２つのタイプのものがある。１つは、すでに計算されてデータベース３２に格納されたクラスター構造を抽出するためのものであり、他のものは、未だ計算されておらず、格納されていないクラスター構造を検索するためのものである。ユーザ入力クエリー・ベクトルは、まず、検索部４０へと伝送される。説明している実施の形態では、検索部４０は、クエリーのタイプを解析する。クエリーが検索部４０に対してすでに計算され、格納されたクラスター構造を検索するように指示する場合には、クエリーは、メモリ領域３０に格納されたSASH構造に対してクエリーを適用し、クエリーを受け取ったパッチ生成部４４がクラスター見積もり部２８へと検索されたデータを伝送する。クラスター見積もり部２８は、検索されたデータを受け取るとパッチ・データおよびそれに伴うSCONFリストを、ハードディスク３２から呼出し、クラスター内コンフィデンスSCONFおよびRSCONFと、クラスター間のコンフィデンスCONFとをそれぞれ使用してクラスターの見積もりを実行する。クエリーされたパッチ生成部４４において生成されたノードは、任意に選択されたノードであるか、またはユーザ入力クエリーにより検索されたノードである。

【０１０７】

得られたクラスター・データは、GUIデータ生成部４２へと伝送され、ディスプレイ部分（図示せず）のディスプレイ・スクリーン上にクラスター・グラフの構造をグラフィカルに表示させる。クラスター・グラフ構造の多くの表示の実施の形態が、本発明において可能である。代表的な１つの実施の形態では、クラスターを、クラスターの含む特徴的なキーワード（もっとも大きな数値）を水平方向に配置し、同時にクラスター・サイズの見積もりと共に垂直方向に配列するものである。このような表示がディスプレイ上に与えられる場合には、GUIデータ生成部４２は、パッチ格納部３２からのクラスター・データをソーティングし、ソートされたデータを適切なディスプレイ・バッファ（図示せず）といったメモリ領域またはコンピュータ１０の別の領域に格納する。

【０１０８】

本発明の特定の実施の形態では、検索部４０がクエリーがすでに検索されておらず、また格納もされていないクラスターの検索を指令していると判断した場合には、検索部４０は、SASHデータ３０を呼出し、ドキュメント－キーワード・ベクトルとクエリー・ベクトルとの間の類似を算出し、SASHの適切なノード・ベクトルを検索する。検索されたデータ・ベクトルは、その後、それら自信がSASHデータ３０内のクエリーのために使用されて、元

のクエリーにより検索されるすべてのベクトルに対して類似ノード・ベクトルのリストを与える。類似ノード・ベクトルの類似のそれぞれのリストは、パッチ規定部２６へと送られ、その後コンフィデンス決定部３８へと送られて、パッチが生成される。これらのパッチは、その後パッチ格納部３２へと追加される。検索されたパッチは、その後クラスター・見積もり部２８へとそれらの対応するSCONFリストと共に伝送されて、元のクエリーにおいて検索されたノードからなるクラスターが見積もられ、算出されたクラスター・データがGUIデータ生成部４２へと送られてクエリーの結果のグラフィカルな表示が行われる。

【０１０９】

GUIデータ生成部４２は、ソートされたクラスター・データをコンピュータ１０に直接接続されたディスプレイ装置（図示せず）へと送り、ディスプレイ・スクリーン上に検索されたクラスター・データの表示を行う。これとは別に、システムがブラウザ・ソフトウエアを使用してインターネットを介して検索された結果を提供する場合には、GUIデータ生成部４２は、クラスターの相関に関連するグラフィカル・データをブラウザ・ソフトウエアに適切な形式、例えばHTMLフォーマットで生成する。

【０１１０】

パートＩＩＩ．発明を実施するための実際のシナリオ

＜シナリオA－データベース内のノードの全体的なクラスター化＞

図１２は、データベース内に格納されたノードの全体的なクラスター化を実行する場合のシナリオのフローチャートを示した図である。シナリオＡのアルゴリズムは、まずドキュメントとキーワード・データとをステップＳ４０で読み込み、ステップＳ４２へと進んでドキュメント－キーワード・ベクトルとキーワード・リストとを生成する。アルゴリズムは、ステップＳ４４で上述したLSI法またはCOV法を使用して次元削減を実行する。その後シナリオAのプロセスは、ステップＳ４６で図２に示したプロセスにしたがって次元削減されたドキュメント－キーワード・ベクトルのSASHを構築する。図１２において示したアルゴリズムをステップ的に実行させるにしたがって生成されるデータ構造を、図１３に示す。

【０１１１】

一度SASH構造が構築されると、類似クエリーがその要素それぞれに対して実行されて、図１４（ａ）に示すように各ドキュメントの１つのパッチを生成する。シナリオＡのアルゴリズムは、その後ステップＳ４８で最適なパッチ・サイズおよびRSCONF値を計算して図１４（ｂ）を与え、その後パッチをそれらのRSCONF値と共に図１４（ｃ）に示す構造として格納する。

【０１１２】

再度図１２を参照して、シナリオAのアルゴリズムは、ステップＳ５０へと進み、各SASHレベルにおいてβ=0.4以下のレベルのコンフィデンスの内パッチ随伴性のすべてに対してパッチの集団を選択する。その後、RSCONF値がα=0.15以上のパッチをさらに選択して、ステップＳ５２でクラスターを決定する。ステップＳ４６～Ｓ５２に関連するデータ構造を、図１５に示す。

【０１１３】

次いで、シナリオAのアルゴリズムは、ステップＳ５４へと進み、所定のしきい値γ以上の随伴コンフィデンス値を有するクラスター間の連結を行う。このデータ構造を、図１６に示す。これらの連結の結果は、クラスター・ラベルおよび対応するキーワードと共にステップＳ５６において、図１７に示されるようなグラフ表示として、グラフ的に与えられる。図１７ではシナリオAにしたがって生成されたクラスターのグラフの部分（γ=0.2）が示されており、その際には、COV次元削減法を使用した。図においてクラスター・ノード（長円で示した。）は、数x/yの対として示されており、xは、クラスターの見積もりサイズであり、yは、それに伴われる要素のパッチ・サイズである。キーワード・ラベルは、各クラスターに対して示されており、ボックスは、同一のラベルのセットを共有するクラスターのサブセットを連結している（ラベルの順番には、わずかな違いが可能である）

。図１７では、106/53で示されるノードに対応するクラスターが示されている。このクラスターは、２つの大きなクラスターのセットの交差のニュース記事を含んでいるので特に興味が持たれるものであり、谷とそれらの開発および保存の記事の他、ゴミ用ダンプ車および他の埋め立ての記事を含んでいる。

【０１１４】

図１２に示したプロセスに含まれる詳細な手順は、下記の通りである。

ｉ）M個のドキュメントを、例えば、バイナリモデルまたはTF-IDF重み付けを使用してベクトルへとモデル化する。これらのベクトルの次元はNであり、これがデータ・セットのアトリビュートの数である。

ｉｉ）さらなる実施例としては、ベクトルのセットをNよりもきわめて小さな数（典型的には200～300）のベクトル・セットとなるように、COVまたはLSI次元削減法を使用して次元削減を適用する。次元削減を採用する場合には、また次元削減したアトリビュート・ベクトルを生成する。

ｉｉｉ）ｋ－最近傍点クエリーを適用するSASHを構築する。Stを、0≦t≦hのｔ番目のSASHレベルとし、ランダムなサンプル$R_t = S_t \cup S_{t+1} \cup \ldots \cup S_h$とする（ここで、$S_0$は、SASHレベルの最低部とする。）

ｉｖ）0≦t≦hのすべてについてv ∈ S_tである各要素について、m =φ(b)の要素について近似的なｍ－最近傍点リスト（ｍ－パッチ）NN(R_t, v, m)を算出し、格納する。

ｖ）図１６に概略的に示したクエリー・クラスターのセットとクラスター構造とのグラフを算出する。

ｖｉ）次元削減が行われた場合には、セットのそれぞれのクエリー・クラスターについてクラスターを構成する次元削減されたドキュメント・ベクトルからアトリビュートであるキーワードのセットを生成する。

ｖｉｉ）好適なユーザ・インタフェイスを使用してユーザがブラウジングすることができるクラスターの得られたセット、それらのサイズ、ラベルおよびクラスター構造のグラフを作成する。

【０１１５】

＜シナリオB－個別的なクラスター：クエリー検索＞

シナリオBでは、シナリオAと同一の処理を使用してSASHを生成する。この後の本質的なステップを図１８に示す。また、シナリオBの処理において生成されるデータ構造を、図１９および図２０に示す。図１８に示されるように、シナリオBのプロセスは、ステップＳ６０でSASH構造を生成し、ステップＳ６２へと進んでユーザ入力クエリーqと、共にターゲットとするクラスター・サイズkとを受け取り、これらを適切なメモリ空間へと格納する。その後SASH内におけるノードを、ステップＳ６４においてSASH構造を使用してクエリーに関連して検索する。ステップＳ６４においては、SASHは、それぞれのランダム・サンプルR_tに関連して0≦t≦hのすべてに対して１つの近隣パッチをクエリー要素qに関連して生成する。その後、プロセスはステップＳ６６へと進んで、RSCONFを計算し、すべてのランダム・サンプルに対してRSCM問題をユーザ入力クエリーについて解く。書くサンプルに対して、クラスターは、これにより生成される。シナリオBのプロセスは、その後これらのクラスターを代表するラベルとキーワードとを、ステップＳ６８において与える。ステップＳ６４～ステップＳ６８において得られたデータ構造が、図２０に示されている。

【０１１６】

シナリオBの詳細な処理を下記に示す。

２－ｉ）シナリオAの手順をｉ～ｉｉｉまで繰り返す。

２－ｉｉ）ユーザに対してクエリー要素qおよびターゲット・クラスターのサイズkを要求する（データ要素に対しては必要ではない）。

２－ｉｉｉ）$t_a = \max\{t \mid k / 2^t \geqq a\}$および$t_b = \min\{t \mid k / 2^t \leqq b\}$を計算する。すべての$t_b \leqq t \leqq t_a$に対して、NN($R_t$, q, m)を計算する。ここで、m =φ(b)である。v ∈ NN(R_t, q, m)のすべてに対して、NN(R_t, v, m)を計算する。

２－ｉｖ）すべての$t_b \leqq t \leqq t_a$に対して、R_tについてのqに対するRSCM問題の解であるk(q,

t)を見出す。
２－ｖ）すべての$t_b \leqq t \leqq t_a$に対して、クエリー・クラスターNN(R_t, q, k(q,t))を構成する次元削減されたドキュメント・ベクトルからアトリビュートであるキーワードのセットを生成する。この手順については図１４において説明した通りである。
２－ｖｉ）得られたクラスターのセット、それらのサイズ、それらの対応するｍ－パッチのSCONFの値、およびそれらのクラスターのラベルを、ユーザに対して提供する。
【０１１７】
【実施例】
本発明を検討するために、本発明の方法を上述した２つのシナリオとして実装した。両方のシナリオについて、TREC-9テキスト検索実験の一部として利用可能なL.A.Timesニュース・ドキュメントのデータベースに適用することにより検討した。データベースは、M=127,742ドキュメントからなり、これらからN=6590キーワード（アトリビュート）をアトリビュート・セットとして抽出した。有効性および汎用性を検討するために、データベースに対して次元削減を行わない手法と、次元削減(COV)を行ない手法の２つの手法を適用した。実装条件は、下記の通りであった。
（ａ）TD-IDF用語重み付けを6590個のアトリビュートに対して行った。
（ｂ）１つの実験セットにおいて、COV次元削減（6590から200へ）を行ない、他については次元削減を行わない。
（ｃ）ドキュメントの最近傍点の検索について、デフォルト・セッティングのSASH（親ノードのキャパシティp=4、子ノードのキャパシティc=16）を使用した。
（ｄ）アトリビュート・ベクトルの最近傍点の検索についても、デフォルト・セッティングのSASH（親ノードのキャパシティp=4、子ノードのキャパシティc=16）を使用した。
【０１１８】
それぞれのシナリオについて、パラメータφ、a、b、β、およびδを、次元削減に伴ういかなるパラメータ（削減次元であるdなど）と、または近似的な類似検索と同様にシステム管理者により設定できるものとした。
【０１１９】
実験条件は、以下の通りである。
（ａ）パッチ範囲のデリミター：a=25, b=120,φ(k)=min{2k, 150}
（ｂ）ドキュメントの最近傍点検索については、近似の精度に影響を与えることになるものの、時間スケーリングする因子として、
【０１２０】
$\mu' = 1.25$
$\mu = 1.25\ \phi(b)$
を用いた。すべての検索について、μ’個の近傍点を生成し、最も近い方からｍ個を使用した（μ’のより大きな値は、より高精度の結果を与えるものの、より検索時間を必要とする。）。
（ｃ）クラスターの相対セルフ－コンフィデンスに対する最大しきい値を、α=0.15とした。
（ｄ）共通のサンプルからのいかなる２つのクラスターの間のコンフィデンスの最大のしきい値をβ=0.4とした。
（ｅ）QCRグラフにおける随伴クラスターの間のコンフィデンスの最小しきい値を、γ=0.15とした（共通のサンプル・レベルから得られる必要はない）。
（ｆ）QCRグラフの２つの随伴クラスターの間のスケールにおける差の最大のしきい値を、δ=4とした。
【０１２１】
計算アルゴリズムを、Java（登録商標）(JDK1.3)を使用して記述し、計算ハードウエアは、Windows（登録商標）2000を走らせた1GHzのプロセッサ速度および512Mbのメインメモリを搭載したIBM社製のIntelliStation（商標）E Proとした。
【０１２２】

２－１．実行時間および記憶コスト

RSCONF値は、一見すると計算するコストが高いように思われるが、注意深く実装することにより、コストを充分低くすることができる。これは、1～φ(b)の範囲のkについて、SCONF(NN(R, q, k))の値のプロファイルを効率的に計算することにより達成される。パッチ・プロファイルのプロットはまた、クエリー要素の近傍において随伴性の変動の効果的な表示を与える。

【０１２３】

後述する表には、シナリオAに伴う時間と、記憶空間のコストとをリストした。時間は、メインメモリに次元削減ドキュメント・ベクトルとアトリビュート・ベクトルとを読み込んだ時点から、クラスターの完全なセットおよびそれらのクラスター・グラフを算出するまでの計算に要した実時間の尺度である。クラスター化およびグラフ構築の時間的なコストは、すべての最近傍パッチがすでに計算されたものと仮定している。

【０１２４】

【表１】

Table 1

記憶のためのコスト（Mb）－次元削減の場合	
ドキュメントのSASH格納	30.1
キーワードのSASH格納	1.6
NNパッチの格納	161.6
次元削減ドキュメントの格納	204.4
次元削減キーワードの格納	5.3
格納全体	403

【０１２５】

【表２】

Table 2

時間的なコスト	次元削減なし	COV次元削減
ドキュメントのSASH構築時間（s）	460.7	898.8
キーワードのSASH構築時間（s）	--	26.6
全NN事前計算時間（s）	7,742.9	13,854.6
クラスター化とグラフ構築の時間（s）	126.2	81.8
全時間（s）	8,329.8	14,861.8
全時間（hr）	2.3	4.1

【０１２６】

２－２．近似的な最近傍点計算

後述する表は、ランダムに選択したSASHに対するサイズをm’として指定したクエリーについて、M個のドキュメントの完全なセットから近似的なm-最近傍点リストを見出すためのコストの平均を示す。比較のため、シーケンシャル検索を使用して正確なクエリーを実行させ、SASHの平均的な精度を算出した（取得されたリストにおける真の最近傍点の割合の尺度となる）。これらの値を使用して、シナリオBについての単一クエリー・クラスターの直接的な生成のコストを決定することができる。これら後者の見積もりにおいては、あらかじめ算出された最近傍点の情報がないドキュメントについてのSASHを使用するものと仮定した。

【０１２７】

【表３】

Table 3

SASH性能	次元削減なし	COV次元削減
平均SASHクエリー距離計算(s)	3,039.03	2,714.57
平均SASHクエリー時間(ms)	38.85	70.74
平均SASHクエリー精度(%)	62.93	94.27
正確なNNクエリー距離計算(s)	127,742	
正確なNNクエリー時間(ms)	1,139.19	2,732.5
単一クエリー・クラスターの距離計算($\times 10^5$)	4.59	4.07
単一クエリー・クラスター時間(s)	5.87	10.68

【０１２８】
２－３．クエリーによる全体のクラスター化
図２１には、COV次元削減を使用した例におけるパッチ・プロファイルの実施の形態を示す。このプロファイルは、シナリオAの方法により生成されたクラスターについてのものである。
【０１２９】
シナリオAにおいて次元削減の有無により生成されたクラスターの数を下記表に示す。
【０１３０】
【表４】

Table 4

見積りクラスター・サイズ（低-高）	次元削減なし	COV次元削減
6400 - 30720	1	1
3200 - 15360	1	2
1600 - 7680	8	8
800 - 3840	15	25
400 - 1920	32	50
200 - 960	70	84
100 - 480	206	135
50 - 240	405	216
25 - 120	760	356

【０１３１】
次元削減した実施例では、基本的な実施例に比較してより少数のマイナー・クラスターが見出されているが、より大きなクラスターも見出している。実験によれば、また次元削減を行う実施例では、より多くの相互連結を与えるクエリー・クラスターのグラフを与え、キーワードの多様性をより解析することができた。
【０１３２】
本発明の方法は、コンピュータ実行可能なプログラムとして実装することができ、本発明のコンピュータ・プログラムは、C言語、C++言語、Java（登録商標）、またはいかなる他のオブジェクト指向言語においても記述することができる。本発明のプログラムは、コンピュータによりデータを書き込み読み込みが可能なフロッピー（登録商標）・ディスク、磁気テープ、ハードディスク、CD-ROM、DVD、光磁気ディスクなどに格納することができる。
【０１３３】
【発明の効果】
改善１：マイナー・クラスター
本発明のクラスター化方法は、通常のコンピュータにおいても良好な随伴性および良好に区別させつつ、データベース内の0.05%程度の小さなサイズのクラスターでも検出することを可能とする。この方法は、セット内におけるクラスターの数に関連して先験的な仮定を必要としない。また、この方法は、生成されるクラスターの局所的な影響のみしか考慮

せずに済ませることを可能とする。重なり合うクラスターは、また許容される。これらの特徴は、従来の方法においてへ現実的でないものであるか、または不可能でさえあったマイナー・クラスターの発見を可能とする。

【０１３４】

改善２：クエリーに基づくクラスター化

本発明のクラスター化方法は、セットの完全なクラスター化の計算という過剰なコストなく、クエリーに近接する有意義な主要クラスターおよびマイナー・クラスターを効率的に生成することができる。本発明者が知る限りにおいて、大規模なテキスト・データに対して上述したことを可能とする方法は新規なものである。

【０１３５】

改善３：クラスター相関関係の自動的な決定

極めて僅かなクラスター化方法のみがクラスター間の重なり合いを可能とするにすぎない。本発明の方法は、クラスターの重なりを使用して、クラスター間の対応を確立するので、ユーザを誘導する“クラスター・マップ”すなわち、関連した概念のグラフを生成することができる。関連性は、データのグループの間において階層の概念のようにではなく、むしろアトリビュート空間内での分類のようにして確立される。データ要素の重なり合いにしたがう組織化は、表現される概念をより柔軟なものとし、本発明の方法は、特に２つ以上の主要クラスターの交差部におけるマイナー・クラスターの発見を可能とする。

【０１３６】

改善４：クラスター品質の自動的な評価

RSCONF値およびパッチ・プロファイルは、クラスターを同定し、比較するばかりではなく、ユーザがクラスター内の随伴性のレベルおよび近接する要素との間の差異を評価するための手段となる。パッチの構造分布は、効果的により高次のテキスト・クラスターの可視化のための既存の空間表現方法を補うことができる。

【０１３７】

改善５：データ分布に対する知見の依存

ほとんどの分割に基づくアルゴリズムのようにではなく、本発明のクエリーに基づくクラスター化は、データの分布に対する知識や仮定を必要とせず、データが均一に分布しているかまたは分布に大きな変動があるかは問題にしない。これは、SASHが上述した特性を有しているように、最近傍リストの生成についても適用することができる。

【０１３８】

改善６：スケーラビリティ

SASH構造が近似的類似クエリーに使用される場合には、データ・セットSの全体のクラスター化のためにPatchCluster法に必要とされる漸近的な時間は、$O(|S| \log_2 |S| + c^2)$となる（Oは、定数である。）。ここで、cは、生成されるクラスターの数である（典型的には、|S|よりもかなり小さい）。先に出現するタームは、プロファイルを生成し、RSCONF値にしたがってクエリー・クラスターの候補をランク付けする際のコストである。重複したクラスターの排除およびグラフのエッジの生成は、合計で、$O(|S| + c \log_2 |S| + c^2)$の時間で実行することができる。

【０１３９】

クエリー・クラスター構築のボトルネックは、最近傍点パッチの事前の計算にある。しかしながら、クラスター化は、近似的なクラスター境界および重なり合いを検出するためには完全に正確な最近傍点リストを必要とはしない。実際には近似的に正確な最近傍点リストを迅速に生成するためにSASHといった回避技術の一つを使用することで、さらにコスト効果が生じる。COV次元削減を使用したL.A.Timesニュース記事のデータ・セットについては、SASHは、シーケンシャル検索のほぼ95%の精度の精度で、粗々４０倍のスピードアップを達成する。パッチの事前計算のための漸近的な負荷は、$O(|S| \log_2 |S|)$で与えられるSASH処理の全コストにより支配される。

【０１４０】

これまで、本発明を図面に示した特定の実施の形態を使用して説明してきた。当然のこと

ながら当業者によれば、多くの別の実施例、変更例、および／または付加例が開示された実施の形態に対して可能であることが理解され、したがって、本発明の真の範囲は、特許請求の範囲にしたがって決定されるものである。

【図面の簡単な説明】

【図１】　図１は、本発明のデータ構造を構築するためのフローチャートを示した図。

【図２】　SASH構造を構築するための処理の概略的なフローチャート。

【図３】　パッチ構造を含むSASH構造の概略図。

【図４】　本発明によるパッチ構造の例示的な概略図。

【図５】　コンフィデンス関数CONFの計算の代表例を示した図。

【図６】　SCONFの計算のための例示的な擬似コードを示した図。

【図７】　パッチおよびセルフ－コンフィデンスの格納構造を示した図。

【図８】　パッチ・プロファイルの更新を行うための例示的な擬似コードを示した図。

【図９】　PatchCluster法（パッチランク付けおよび選択を含む）の例示的な擬似コードを示した図。

【図１０】　本発明において典型的に使用されるコンピュータ・システムのブロック図。

【図１１】　本発明のコンピュータ・システムの機能ブロック図。

【図１２】　クラスターとそれらの相関関係のグラフを生成するための処理のフローチャート（シナリオA）。

【図１３】　図１２に示したシナリオAの処理に関連したデータ構造を図示した図。

【図１４】　図１２に示したシナリオAの処理に関連したデータ構造を図示した図。

【図１５】　図１２に示したシナリオAの処理に関連したデータ構造を図示した図。

【図１６】　クラスターの相関関係のグラフを示した図。

【図１７】　クラスターの相関関係の例示的なグラフ表示を示した図

【図１８】　クラスターとそれらの相関関係のグラフを生成するための処理のフローチャート（シナリオB）。

【図１９】　図１８に示したシナリオBの処理に関連したデータ構造を図示した図。

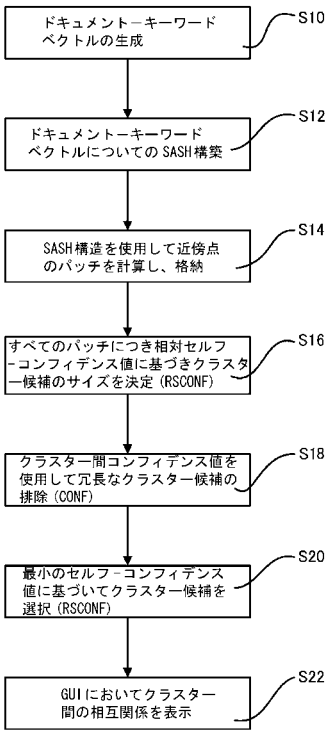
【図２０】　図１８に示したシナリオBの処理に関連したデータ構造を図示した図。

【図２１】　SCONF値と見積もられたクラスター・サイズとによりプロットされたプロファイルを示した図。

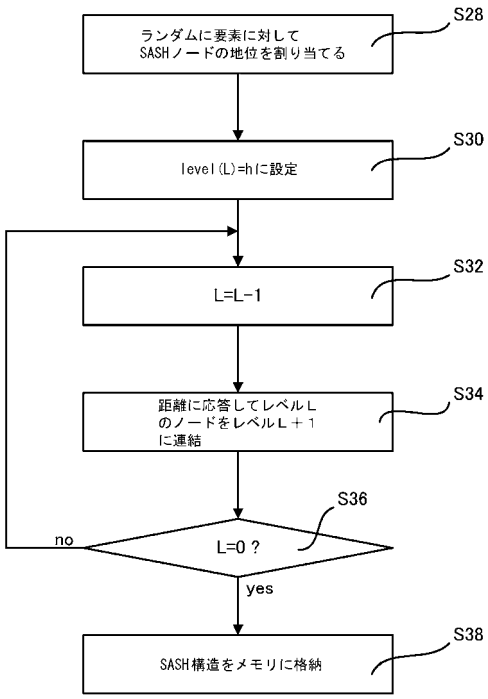
【符号の説明】

１０…コンピュータ
１２…ディスプレイ装置
１４…キーボード
１６…マウス
１８…データベース
２０…通信ライン
２２…ハブ／ルータ
２４…ドキュメント・ベクトル生成部
２６…パッチ規定部
２８…クラスター見積もり部
３０…メモリまたはデータベース（SASH記憶部）
３２…ハードディスク
３４…近傍パッチ生成部
３６…SASH生成部
３８…コンフィデンス決定部
４０…検索部
４２…GUIデータ生成部
４４…クエリー・パッチ生成部
４６…クエリー・ベクトル生成部

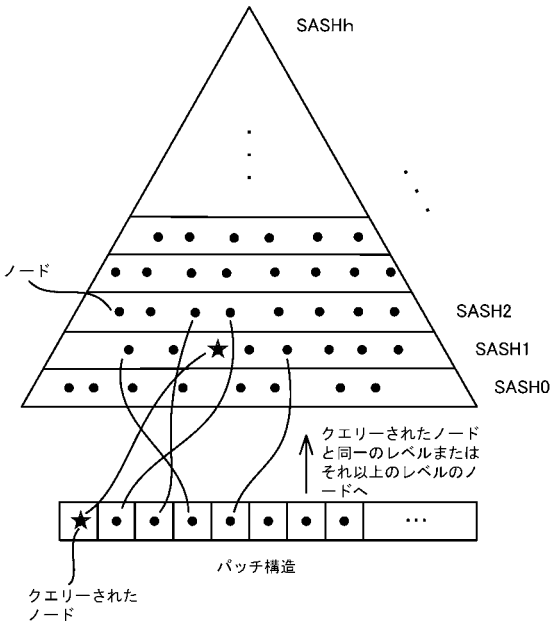
【図１】


ドキュメント－キーワード
ベクトルの生成
S10
ドキュメントーキーワード
ベクトルについての SASH構築
S12
SASH構造を使用して近傍点
のパッチを計算し、格納
S14
すべてのパッチにつき相対セルフ
-コンフィデンス値に基づきクラスタ
ー候補のサイズを決定（RSCONF）
S16
クラスター間コンフィデンス値を
使用して冗長なクラスター候補の
排除（CONF）
S18
最小のセルフ-コンフィデンス
値に基づいてクラスター候補を
選択（RSCONF）
S20
GUIにおいてクラスター
間の相互関係を表示
S22

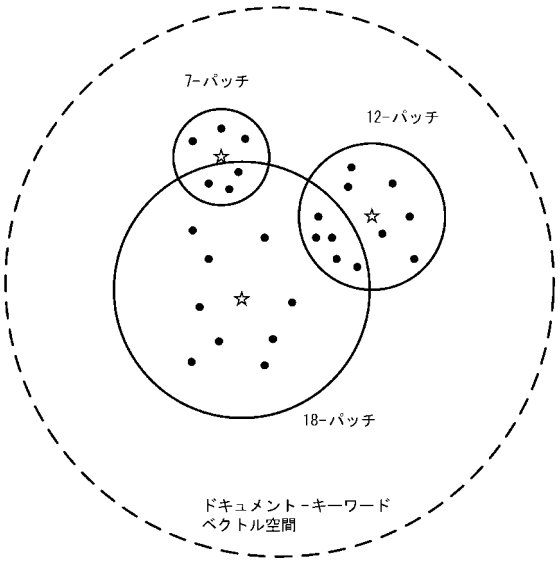
【図２】


ランダムに要素に対して
SASHノードの地位を割り当てる
S28
level(L)=hに設定
S30
L=L-1
S32
距離に応答してレベルL
のノードをレベルL＋1
に連結
S34
L=0？
S36
no
yes
SASH構造をメモリに格納
S38

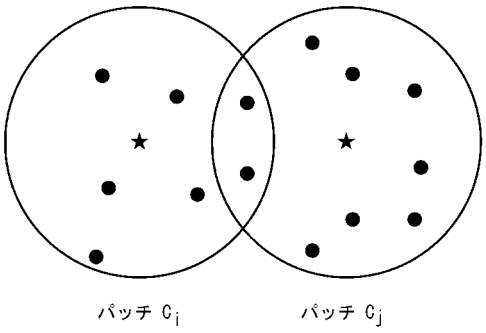
【図３】


SASHh
ノード
SASH2
SASH1
SASH0
クエリーされたノード
と同一のレベルまたは
それ以上のレベルのノ
ードへ
パッチ構造
クエリーされた
ノード

【図４】


7-パッチ
12-パッチ
18-パッチ
ドキュメント-キーワード
ベクトル空間

【図５】



$CONF(C_i, C_j) = 2/8 = 25\%$

$CONF(C_j, C_i) = 2/10 = 20\%$

【図６】

```
Profile (query q; maximum patch size m): SCONF list SCONFL
    {Let QNL be the m-patch precomputed for query q.}
    {Let NNL be a list of the m-patches precomputed for every element of QNL.}
    {Initially, w.count = ∅ is assumed for every element v in the data set.}
1.  score ← 0;
    {Initially, no query neighbors are in the current patch.}
    for i = 1 to m do
2.      QNL[i].count ← 0;
    end for
    for i = 1 to m do
        {Retrieve the number of times QNL[i] has been encountered as an external neighbor so far.}
3.      score ← score + QNL[i].count;
        {Indicate that henceforth QNL[i] is in the current i-patch.}
4.      QNL[i].count ← present;
        for j = 1 to i − 1 do
5.          w ← NNL[j, i];
            if w.count = present then
6.              score ← score + 1;
            else if w.count ≥ 0 then
7.              w.count ← w.count + 1;
            end if
8.          w ← NNL[i, j];
            if w.count = present then
9.              score ← score + 1;
            else if w.count ≥ 0 then
10.             w.count ← w.count + 1;
            end if
        end for
11.     w ← NNL[i, i];
        if w.count = present then
12.         score ← score + 1;
        else if w.count ≥ 0 then
13.         w.count ← w.count + 1;
        end if
14.     SCONFL[i] = score/i^2;
    end for
    {Reset the counts to their default value.}
    for i = 1 to m do
15.     QNL[i].count ← ∅;
    end for
```

【図７】

アイテムq_i	SASH レベル	パッチ
i=n−1	0	$NN(R_{0,q(n-1)},m)$
i=n−2	0	$NN(R_{0,q(n-2)},m)$
i=n−3	0	$NN(R_{0,q(n-3)},m)$
・ ・ ・	・ ・ ・	・ ・ ・
i=n/2−1	1	$NN(R_{1,q(n/2-1)},m)$
・ ・ ・	・ ・ ・	・ ・ ・
i=n/4−1	2	$NN(R_{2,q(n/4-1)},m)$
・ ・ ・	・ ・ ・	・ ・ ・
i=0	h	$NN(R_{h,q(0)},m)$

【図８】

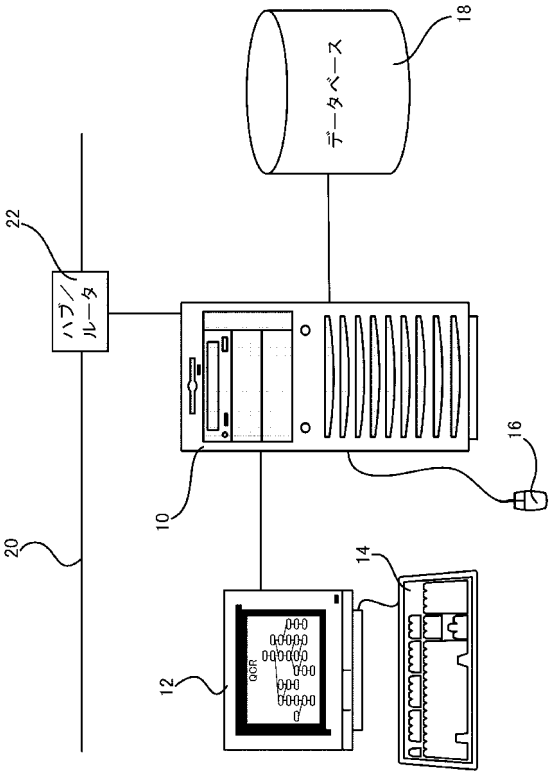
```
RefineProfile (query q;
               inner patch size k_I;
               outer patch size k_O): reordered query k_I-patch RQNL
    {Let QNL be the k_O-patch precomputed for query q.}
    {Let NNL be a list of the k_O-patches precomputed for every element of QNL.}
    {Initially, v.inpatch = false is assumed for every element v in the data set.}
    {Identify the inner patch members.}
    for i = 1 to k_I do
1.      QNL[i].inpatch ← true;
    end for
    {Initialize the confidence value CONFc of every patch element to zero.}
    for i = 1 to k_O do
2       CONFc[i] ← 0;
    end for
    {For each element of the outer patch, count the number of elements
       of their k-nearest-neighbor sets shared with that of q.}
    for i = 1 to k_O do
        for j = 1 to k_I do
3.          w ← NNL[i, j];
            if w.inpatch = true then
4.              CONFc[i] ← CONFc[i] + 1;
            end if
        end for
5.      CONFc[i] ← CONFc[i]/k_O;
    end for
    {Reorder the outer patch elements according to their confidence values, from highest to lowest.}
6.  RQNL ← sort(QNL, CONFc, k_O);
    {Reset the patch membership indicators to their default values.}
    for i = 1 to k_I do
7.      QNL[i].inpatch ← false;
    end for
```

【図９】

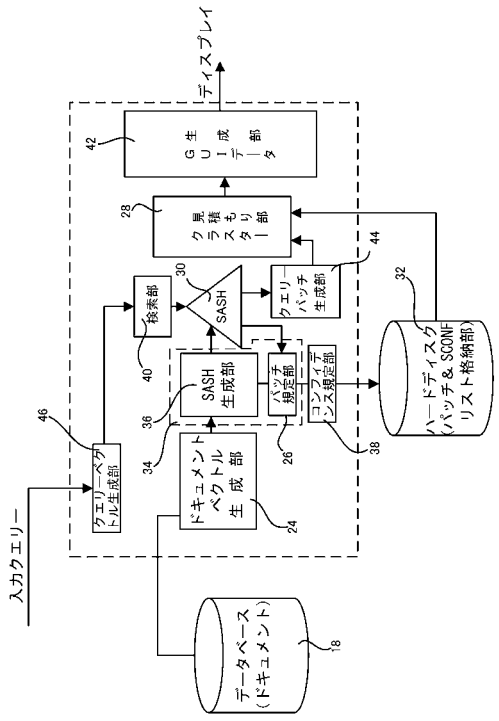
PatchCluster (data set S;
RSCM parameters a, b, $m = \varphi(b)$;
Thresholds α, β, γ, δ): query cluster graph G

1. Randomly partition the set S into subsets S_t of approximate size $\frac{|S|}{2^t}$, for $0 \le t \le h = \lfloor \log_2 |S| \rfloor$.
2. For all $0 \le t \le h$ do:
   (a) For every element $v \in S_t$, compute nearest-neighbor patches $\mathbf{NN}(R_t, v, m)$, where $R_t = \bigcup_{i \ge t} S_i$.
   (b) For each element $v_{t,i} \in S_t$, compute the optimal query cluster size $k(v_{t,i})$ maximizing $\mathbf{RSCONF}(\mathbf{NN}(R_t, v_{t,i}, k), \varphi)$, for values of k between a and b
   The ranked collection of patches
   $$\mathcal{C}_t = \langle C_{t,i} | i < j \Rightarrow \mathbf{RSCONF}(C_{t,i}, \varphi) \ge \mathbf{RSCONF}(C_{t,j}, \varphi) \rangle$$
   form the candidates for the query clusters associated with sample $R_t \subseteq S$, where $C_{t,i} = \mathbf{NN}(R_t, v_{t,i}, k(v_{t,i}))$ and $C_{t,j} = \mathbf{NN}(R_t, v_{t,j}, k(v_{t,j}))$.
   (c) Let Q_t be a list of patches of $\mathcal{C}_t$ that have been confirmed as query clusters of R_t. Initially, Q_t is empty.
   (d) For all $1 \le i \le |\mathcal{C}_t|$ do:
      i. If $\mathbf{RSCONF}(C_{t,i}, \varphi) < \alpha$, then break from the loop.
      ii. For all $w \in C_{t,i}$ do: if $\mathbf{NN}(R_t, w, k) \notin |\mathcal{C}_t|$ for any value of k, or failing that, if $\max\{\mathbf{CONF}(\mathbf{NN}(R_t, w, k), C_{t,i}), \mathbf{CONF}(C_{t,i}, \mathbf{NN}(R_t, w, k))\} < \beta$, then add $C_{t,i}$ to the list Q_t.
3. Let h' be the largest index for which $|Q_{h'}| > 0$. Let $\{C_{t,j}\}$ be the set of patches comprising Q_t, where $C_{t,j} = \mathbf{NN}(R_t, q_{t,j}, k(q_{t,j}))\}$, for all $0 \le t \le h'$. Initialize the node set of the query cluster graph G to contain these patches, one patch per node.
4. For all $\delta \le t \le h'$, all $1 \le j \le |Q_t|$, and all $\max\{0, t - \delta\} \le s \le t$, do:
   (a) Compute $C'_{t,j} = \mathbf{NN}(R_s, q_{t,j}, 2^{t-s}k(q_{t,j}))\}$.
   (b) For all $1 \le i \le |Q_s|$, if $C_{s,i} \ne C'_{t,j}$ and $\max\{\mathbf{CONF}(C_{s,i}, C'_{t,j}), \mathbf{CONF}(C'_{t,j}, C_{s,i})\} \ge \gamma$, then introduce the edges $(C_{s,i}, C_{t,j})$ and $(C_{t,j}, C_{s,i})$ into G, with weights $\mathbf{CONF}(C_{s,i}, C_{t,j})$ and $\mathbf{CONF}(C_{t,j}, C_{s,i})$, respectively.

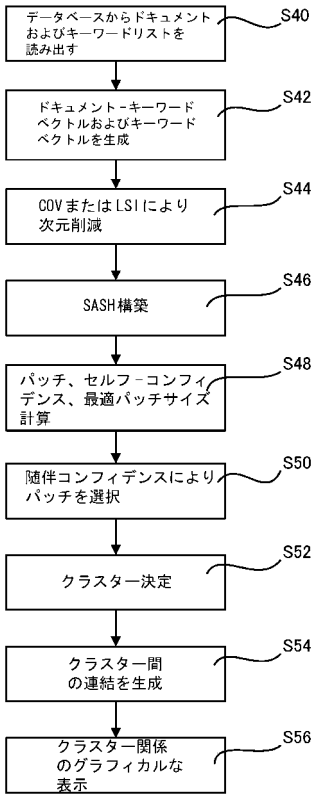
【図１０】



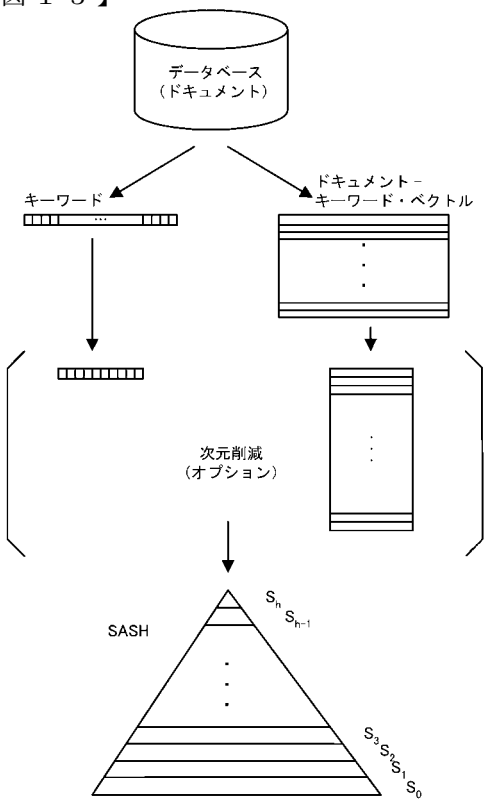
【図１１】



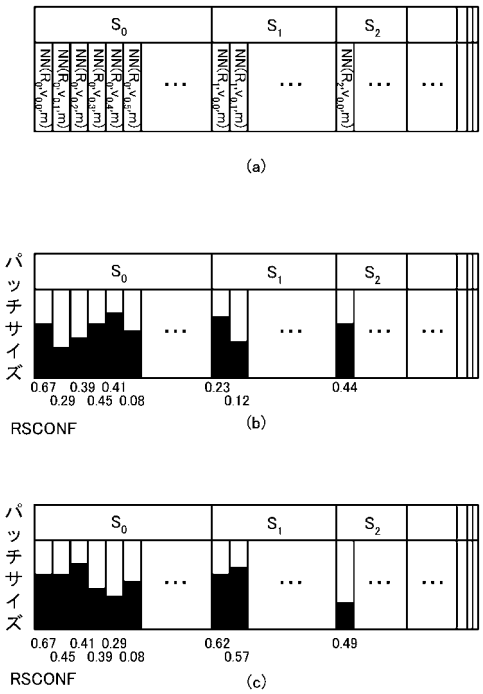
【図１２】



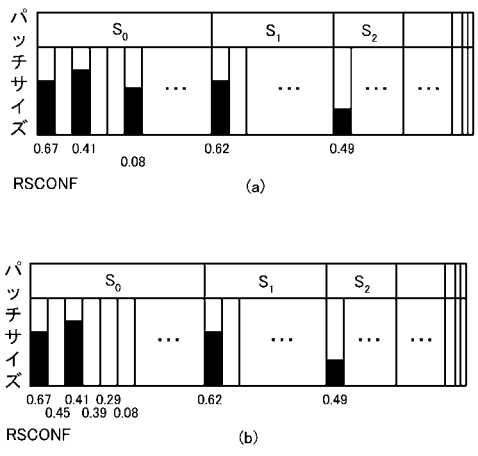
【図１３】



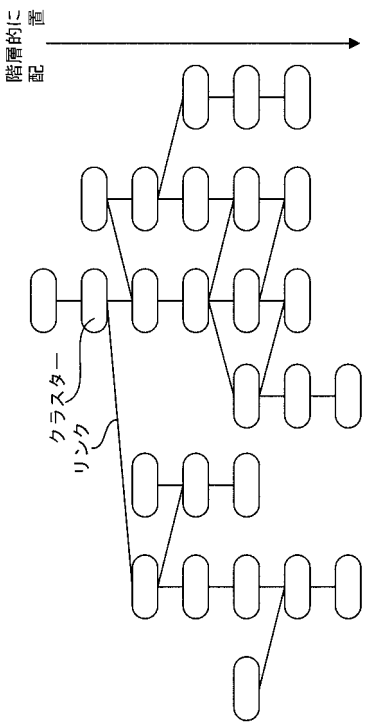
【図１４】



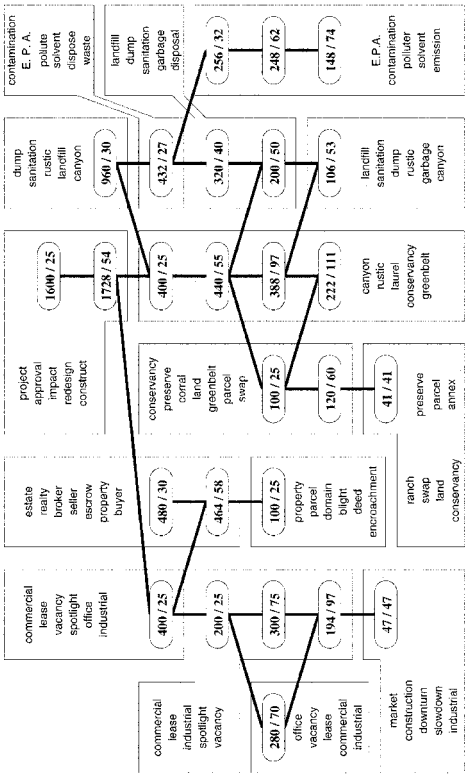
【図１５】



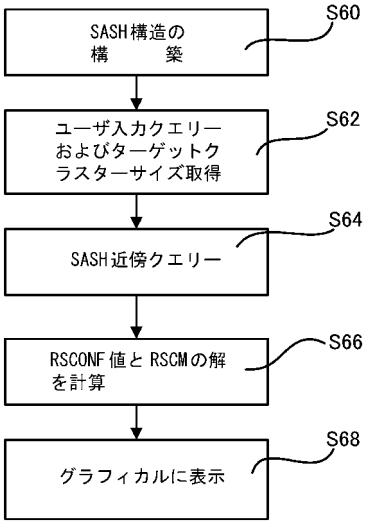
【図１６】



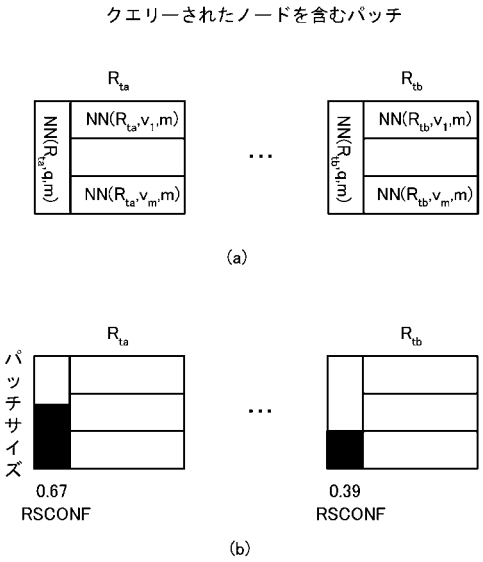
【図１７】



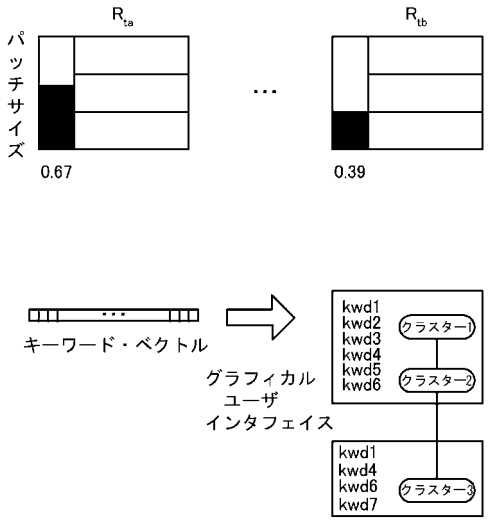
【図１８】



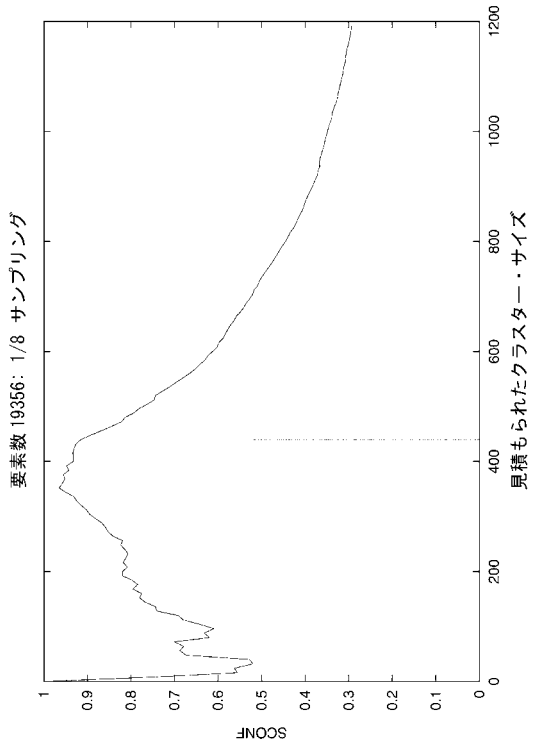
【図１９】



【図２０】



【図２１】


要素数 19356: 1/8 サンプリング
SCONF
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
200
400
600
800
1000
1200
見積もられたクラスター・サイズ

フロントページの続き

(72)発明者　マイケル・エドワード・フール
神奈川県大和市下鶴間１６２３番地１４　日本アイ・ビー・エム株式会社　東京基礎研究所内

審査官　辻本　泰隆

(56)参考文献　特開２００２－１８３１７１（ＪＰ，Ａ）
特開２００１－２５６２４４（ＪＰ，Ａ）
徳永 健伸，言語と計算５　情報検索と言語処理，日本，財団法人東京大学出版会，１９９９年１１月２５日，初版，６０－６５頁

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F　17/30

(54)【発明の名称】情報検索のためのデータ構造を生成するコンピュータ・システム、そのための方法、情報検索のためのデータ構造を生成するコンピュータ実行可能なプログラム、情報検索のためのデータ構造を生成するコンピュータ実行可能なプログラムを記憶したコンピュータ可読な記憶媒体、情報検索システム、およびグラフィカル・ユーザ・インタフェイス・システム