(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3953295号
(P3953295)
(45)発行日　平成19年8月8日(2007.8.8)　(24)登録日　平成19年5月11日(2007.5.11)

(51)Int.Cl.　F I
G06F 17/30 (2006.01)　G06F 17/30 350C
G06F 17/30 330A
G06F 17/30 340Z
G06F 17/30 412

請求項の数 12　(全 44 頁)

(21)出願番号　特願2001-324437(P2001-324437)
(22)出願日　平成13年10月23日(2001.10.23)
(65)公開番号　特開2003-141160(P2003-141160A)
(43)公開日　平成15年5月16日(2003.5.16)
審査請求日　平成14年10月23日(2002.10.23)

(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

最終頁に続く

(54)【発明の名称】情報検索システム、情報検索方法、情報検索を実行させるためのプログラムおよび情報検索を実行させるためのプログラムが記録された記録媒体

(57)【特許請求の範囲】
【請求項１】
データベースに時間の経過と共に追加されるドキュメントを検索するための情報検索システムであって、前記ドキュメントは、ドキュメント－アトリビュート行列へと変換されて前記情報検索システムに保持され、かつ前記ドキュメント－アトリビュート行列は、逐次的に追加されるドキュメント－アトリビュート副行列から構成され、
前記ドキュメント－アトリビュート行列から共分散行列を生成し、ドキュメントベクトルの積和行列（ＳＵＭ（Ｍ）$_1$）と、ドキュメント－ベクトルの平均（ＭＥＡＮ（Ｍ）$_1$）と、ドキュメント－ベクトルの平均の積行列（ＳＵＭ（Ｍ）$_2$）と、ドキュメントの全数（Ｍ）とを保持させるための手段と、
前記データベースに所定の期間の間に追加されたドキュメントからドキュメント－アトリビュート副行列を生成するための手段と、
生成された前記共分散行列と、前記ＳＵＭ（Ｍ）$_1$と、前記ＭＥＡＮ（Ｍ）$_1$と、前記ＳＵＭ（Ｍ）$_2$と、前記Ｍと、追加された前記副行列のドキュメント数（Ｈ）とからなる前記ドキュメント－アトリビュート副行列に関連する情報を使用して前記共分散行列を更新し、更新された前記共分散行列を特異値分解して、データベースに保持されたすべてのドキュメント－アトリビュート行列の次元削減を実行するための手段と、
前記次元削減されたドキュメント－アトリビュート行列を使用してユーザが入力したクエリーによる情報検索を行うための手段と
を含む、情報検索システム。

【請求項２】
前記すべてのドキュメント－アトリビュート行列の次元削減を実行するための手段は、追加された前記副行列を含むすべてのドキュメント－アトリビュート行列の共分散行列Ｃ’を、下記式

$$C' = \frac{1}{(M+H)}\ \mathrm{SUM}(M+H)_1 - \mathrm{SUM}(M+H)_2$$

により生成する手段を含む
請求項１に記載の情報検索システム。
【請求項３】
さらに、ドキュメント－ベクトルに含まれるアトリビュートを自動的に検索し、アトリビュート・ハッシュ・テーブルを生成して前記アトリビュートを追加または削除するための手段を含む
請求項１または２に記載の情報検索システム。
【請求項４】
データベースに時間の経過と共に追加されるドキュメントを検索するための情報検索方法であって、前記方法は情報検索システムによって実行され、前記ドキュメントは、ドキュメント－アトリビュート行列へと変換されて保持され、かつ前記ドキュメント－アトリビュート行列は、逐次的に追加されるドキュメント－アトリビュート副行列から構成され、
前記情報検索システムが備える保持させるための手段が、前記ドキュメント－アトリビュート副行列から共分散行列を生成し、ドキュメントベクトルの積和行列（ＳＵＭ（Ｍ）$_1$）と、ドキュメント－ベクトルの平均（ＭＥＡＮ（Ｍ）$_1$）と、ドキュメント－ベクトルの平均の積行列（ＳＵＭ（Ｍ）$_2$）と、ドキュメントの全数（Ｍ）とを保持させるステップと、
前記情報検索システムが備えるドキュメント－アトリビュート副行列を生成するための手段が、前記データベースに所定の期間の間に追加されたドキュメントからドキュメント－アトリビュート副行列を生成するステップと、
前記情報検索システムが備える次元削減を実行するための手段が、生成された前記共分散行列と、前記ＳＵＭ（Ｍ）$_1$と、前記ＭＥＡＮ（Ｍ）$_1$と、前記ＳＵＭ（Ｍ）$_2$と、前記Ｍと、追加された前記副行列のドキュメント数（Ｈ）とからなる前記ドキュメント－アトリビュート副行列に関連する情報を使用して前記共分散行列を更新し、更新された前記共分散行列を特異値分解して、データベースに保持されたすべてのドキュメント－アトリビュート行列の次元削減を実行するステップと、
前記情報検索システムが備える情報検索を行うための手段が、前記次元削減されたドキュメント－アトリビュート行列を使用してユーザが入力したクエリーによる情報検索を行うステップと
を含む、情報検索方法。
【請求項５】
前記すべてのドキュメント－アトリビュート行列の次元削減を実行するステップは、前記情報検索システムが備える共分散行列Ｃ’を生成する手段が、追加された前記副行列を含むすべてのドキュメント－アトリビュート行列の共分散行列Ｃ’を、下記式

$$C' = \frac{1}{(M+H)}\ \mathrm{SUM}(M+H)_1 - \mathrm{SUM}(M+H)_2$$

により生成するステップを含む
請求項４に記載の情報検索方法。
【請求項６】
前記情報検索システムが備えるアトリビュートを追加または削除するための手段が、さらに、ドキュメント－ベクトルに含まれるアトリビュートを自動的に検索し、アトリビュ

ート・ハッシュ・テーブルを生成して前記アトリビュートを追加または削除するステップを含む

請求項４または５に記載の情報検索方法。

【請求項７】

データベースに時間の経過と共に追加されるドキュメントを検索するための情報検索方法をコンピュータに実行させるためのプログラムであって、前記ドキュメントは、ドキュメント－アトリビュート行列へと変換されて保持され、かつ前記ドキュメント－アトリビュート行列は、逐次的に追加されるドキュメント－アトリビュート副行列から構成され、

前記ドキュメント－アトリビュート副行列から共分散行列を生成し、ドキュメントベクトルの積和行列（ＳＵＭ（Ｍ）$_1$）と、ドキュメント－ベクトルの平均（ＭＥＡＮ（Ｍ）$_1$）と、ドキュメント－ベクトルの平均の積行列（ＳＵＭ（Ｍ）$_2$）と、ドキュメントの全数（Ｍ）とを保持させるステップと、

前記データベースに所定の期間の間に追加されたドキュメントからドキュメント－アトリビュート副行列を生成するステップと、

生成された前記共分散行列と、前記ＳＵＭ（Ｍ）$_1$と、前記ＭＥＡＮ（Ｍ）$_1$と、前記ＳＵＭ（Ｍ）$_2$と、前記Ｍと、追加された前記副行列のドキュメント数（Ｈ）とからなる前記ドキュメント－アトリビュート副行列に関連する情報を使用して前記共分散行列を更新し、更新された前記共分散行列を特異値分解して、データベースに保持されたすべてのドキュメント－アトリビュート行列の次元削減を実行するステップと、

前記次元削減されたドキュメント－アトリビュート行列を使用してユーザが入力したクエリーによる情報検索を行うステップと

を含む、情報検索方法を前記コンピュータに実行させるためのプログラム。

【請求項８】

前記すべてのドキュメント－アトリビュート行列の次元削減を実行するステップにおいて、追加された前記副行列を含むすべてのドキュメント－アトリビュート行列の共分散行列Ｃ’を、下記式

$$\mathbf{C}' = \frac{1}{(M+H)}\ \mathrm{SUM}(M+H)_1 - \mathrm{SUM}(M+H)_2$$

により生成するステップを実行させる

請求項７に記載のプログラム。

【請求項９】

さらに、ドキュメント－ベクトルに含まれるアトリビュートを自動的に検索し、アトリビュート・ハッシュ・テーブルを生成して前記アトリビュートを追加または削除するステップを実行させる

請求項７または８に記載のプログラム。

【請求項１０】

データベースに時間の経過と共に追加されるドキュメントを検索するための情報検索方法をコンピュータに実行させるためのプログラムが記録されたコンピュータ可読な記録媒体であって、前記ドキュメントは、ドキュメント－アトリビュート行列へと変換されて保持され、かつ前記ドキュメント－アトリビュート行列は、逐次的に追加されるドキュメント－アトリビュート副行列から構成され、

前記ドキュメント－アトリビュート行列から共分散行列を生成し、ドキュメントベクトルの積和行列（ＳＵＭ（Ｍ）$_1$）と、ドキュメント－ベクトルの平均（ＭＥＡＮ（Ｍ）$_1$）と、ドキュメント－ベクトルの平均の積行列（ＳＵＭ（Ｍ）$_2$）と、ドキュメントの全数（Ｍ）とを保持させるステップと、

前記データベースに所定の期間の間に追加されたドキュメントからドキュメント－アトリビュート副行列を生成するステップと、

生成された前記共分散行列と、前記ＳＵＭ（Ｍ）$_1$と、前記ＭＥＡＮ（Ｍ）$_1$と、前記ＳＵＭ（Ｍ）$_2$と、前記Ｍと、追加された前記副行列のドキュメント数（Ｈ）とからなる

前記ドキュメント－アトリビュート副行列に関連する情報を使用して前記共分散行列を更新し、更新された前記共分散行列を特異値分解して、データベースに保持されたすべてのドキュメント－アトリビュート行列の次元削減を実行するステップと、

前記次元削減されたドキュメント－アトリビュート行列を使用してユーザが入力したクエリーによる情報検索を行うステップと

を含む、情報検索方法を前記コンピュータに実行させるためのプログラムが記録された記録媒体。

【請求項１１】

前記すべてのドキュメント－アトリビュート行列の次元削減を実行するステップにおいて、追加された前記副行列を含むすべてのドキュメント－アトリビュート行列の共分散行列Ｃ’を、下記式

$$\mathbf{C}' = \frac{1}{(M+H)}\ \mathrm{SUM}(M+H)_1 - \mathrm{SUM}(M+H)_2$$

により生成するステップを実行させる

請求項１０に記載の記録媒体。

【請求項１２】

さらに、ドキュメント－ベクトルに含まれるアトリビュートを自動的に検索し、アトリビュート・ハッシュ・テーブルを生成して前記アトリビュートを追加または削除するステップを実行させる

請求項１０または１１に記載の記録媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、データベースに保持されたデータの情報検索に関し、より詳細にはデータベースに対して情報が逐次的に追加され、情報が更新されるきわめて大きなデータベースのための情報検索システム、情報検索方法、情報検索を行うためのプログラムおよび該プログラムが記録されたコンピュータ可読な記録媒体に関する。

【０００２】

【従来の技術】

ドキュメントのベクトル空間モデルを使用する情報検索システムは、きわめて大きな、静的なデータベースに対して比較的成功したものといえる。Deerwesterらは、テキストに基づいたドキュメントの情報検索に関連する問題をより低次元のサブスペースへとマッピングするアルゴリズムである、潜在的意味解析法(Latent Semantic Indexing)を開発し、実時間における検索を可能とさせている。Deerwesterらのアルゴリズムは、ドキュメント－アトリビュート行列の特異値分解(Singular value decomposition: SVD)の効率的で、数値的に正確な計算に基づくものである。上述したDeerwesterらの手法は、精度が高く充分なものではあるが、そのために使用するサブスペースの基底ベクトルの決定は通常、逐次的なドキュメントデータの追加および削除の要求がある場合でもドキュメントデータに対する特異値分解を含む計算により実行されるのでコストが高く、例えば一晩を要するという不都合がある。

【０００３】

情報検索における上述したＳＶＤに基づく次元削減のためのアルゴリズムは、例えば、ニュース・データベース、医療データベース、消費者プロファイル・データベースといったきわめて迅速に情報が追加される、きわめて大きなデータベースから情報を検索するのに適切ではない。この理由は、主として（１）データベースは、新たに追加される情報を考慮すると共に、日付が古くなったり不要になった情報を削除するため、本明細書においてアップデートまたはダウンデートとして参照するプロセスを通じて頻繁に更新されなければならないこと、（２）データベース内のコンテンツの変更に対応するためのデータベースの更新（すなわちアップデートまたはダウンデート）のたびに、ドキュメント－アトリ

ビュート行列に対して再度基底ベクトルを算出する必要が生じ、このため、計算時間、労力、ハードウエア資源などに関連してコストが極めて高くなることを挙げることができる。

【０００４】

＜先行技術の開示＞

これまで上述した問題点を解決するべく、いくつかの方法が提案されている。Berry, DumaisおよびO’Brienは、“Using linear algebra for intelligent information retrieval”、 1995、 pp.573-595、において、およびZhaおよびSiomonは、“On updating problems in latent semantic indexing”、 SIAM Journal of Scientific Computation、 Vol. 21、 No. 2、 pp. 782-791、 March 2000において、ＳＶＤに基づく次元削減につき、ベクトル空間モデルの下でドキュメントをアップデート、すなわちドキュメントを付け加える方法を提案している。しかしながら、Berryらの方法は、ベクトル空間モデルに対するアップデートの正確な計算ではなく単に近似でしかないために、信頼性において充分ではないという不都合があった。

【０００５】

Berryら（１９９５）およびZhaら（２０００）によるDeerwesterらのアルゴリズムに対するアップデート方法も特異値分解に対する非線形な近似解であり、信頼性に欠けるという不都合がある。

【０００６】

一方、Witterは、“Downdating the latent semantic model for information retrieval”、 M.A. Thesis、 Univ. of Tennessee、 Knoxville、 Dec. 1997、において、Deerwesterらのアルゴリズムのためのダウンデート方法を開示している。しかしながら、Witterの方法を急速に変化するデータベースのダウンデートに対して適用すると、（１）ドキュメントが一度に削除されるので、以後に連続するダウンデートのための浮動小数点計算における誤差が蓄積し、これが重大な影響を与えること、（２）ダウンデートは、次元削減されたドキュメント－アトリビュート行列に対してのみ実行されるので、次元削減された行列は、その次元を失い元々の行列にある少量ではあるが意味的に重要ないわゆるアウトライア・ドキュメントを検出できずに、ドキュメントが削除されるにつれて主要トピックスのみになってしまう可能性があることである。これについては、本願出願人らによる特許出願、特願２００１－２０５１８３号にも詳細に記載されている。

【０００７】

上述した理由から、Witterのアルゴリズムによるダウンデートされたドキュメント－アトリビュート行列の出力は不正確であり、情報検索を含んだ多くの用途に対して適切ではない。なお、浮動小数点計算における誤差に関する検討は、最も標準的な数値解析のテキストに記載されており、このようなテキストとしては、GolubおよびVan Loan、”Matrix Computations”、第２版、John Hopkins Univ. Press、 Baltimore、 MD、 1989およびGoldbergによる総説論文”What every computer scientist should know about floating-point arithmetic”、 ACM Computing Surveys、 Vol. 23、 No. 1、１９９１年３月を挙げることができる。

【０００８】

またこれまで、急速に変化するデータベースを含み、ＳＶＤに基づいた情報検索用途に使用することができるに足る、充分な精度および効率で、次元削減されたドキュメント－アトリビュート行列のアップデートを実行するための方法は知られていない。精度および効率を必要とするアップデート用途の重要なクラスの例としては、本願出願人らによるデータベースにおける新たなトピックス／事象の検出およびその累積の追跡を挙げることができる。

【０００９】

一方で、ＳＶＤのプロセスについて考察すれば、一般的なドキュメント－アトリビュート行列ＡのＳＶＤは、下記式により与えられる。

【００１０】

【数１】

$$A = U \Sigma V^{T}$$

（上記式中、Ｕ、Ｖは、直交行列を示し、Σは、対角行列を示し、Ｔは、行列の転置を示す。以下、本発明において同様である。）

【００１１】

この場合、上述した行列ＡのＳＶＤを、$A^{T}A$やAA^{T}の固有値問題として実行する方法もあり得る。しかしながら、$A^{T}A$やAA^{T}生成させる場合には、小さな特異値に対して激しい精度低下を生じさせるという問題が生じ、反復してドキュメント－アトリビュート行列のアップデートに対応すると、出力精度が著しく低下してしまうという問題が生じることにもなる。

【００１２】

【発明が解決しようとする課題】

すなわち本発明は、ＳＶＤを使用してきわめて大きく、かつ急速に変化するデータベースから実時間での情報検索を行うための情報検索システム、情報検索方法、情報検索を実行させるためのプログラムおよび情報検索を実行させるためのプログラムが記録された記録媒体を提供することを目的とする。

【００１３】

【課題を解決するための手段】

本発明は、上述した課題を解決するに際して、ドキュメント－アトリビュート行列の以前に算出した結果を一部再利用すれば、計算時間の短縮、精度向上、および効率化を達成することができる、という新奇な着想に基づいてなされたものである。

【００１４】

本発明は、上述した以前の計算結果を使用するプロセスとして、ドキュメント－アトリビュート行列をＱＲ分解し、新たに付け加えられたドキュメントを含むドキュメント－アトリビュート行列を、すでに算出されたＱＲ分解の結果のうちの直前に生成された行列Ｒを使用する。具体的には行列Ｒと付け加えられたドキュメント－アトリビュート副行列を含む行列をハウスホルダー変換し、付け加えられたドキュメント－アトリビュート副行列の結果を反映させて新たな行列Ｒとして更新する。

【００１５】

本発明においては、上述のようにして更新された上三角行列ＲをＳＶＤに用い、得られた特異ベクトルを使用してドキュメント－アトリビュート行列の次元削減を実行することにより、急速にデータが追加されるデータベースにおける次元削減プロセスを高精度、かつ効率的に実行させ、最新のデータベースでの情報検索を可能にするものである。

【００１６】

また、本発明の別の実施の形態においては、ドキュメント－アトリビュート行列について共分散行列を生成し、追加されたドキュメントからなるドキュメント－アトリビュート行列について以前に算出された共分散行列に関連する所定の行列を使用して共分散行列を更新する。本発明においては、共分散行列の更新の際にアトリビュートを追加・削除することもできる。上述のようにして得られた共分散行列に対しＳＶＤを実行して特異ベクトルを生成し、ドキュメント－アトリビュート行列の次元削減に使用することにより、次元削減プロセスの高精度化および効率化を達成するものである。

【００１７】

すなわち、動的に変化するドキュメント－アトリビュート行列の本発明による特異値トリプレット（特異値と、それに対応する左および右特異ベクトル）のアップデートは、上述した先行技術において提案された結果よりも、より数値的に正確な結果を与えることができる。本発明は、特にアップデートされた行列のＳＶＤを迅速に計算すると共に、特異値トリプレットの近似を与えるものではなく、数値的に正確な計算を実行させるものである。

【００１８】
さらに、本発明は、多数回のアップデートが行われることに対応しているので、浮動小数点誤差の蓄積を最小化させるものである。情報検索システムによる次元削減の問題に対してＳＶＤを使用する結果の出力の品質は、一般には、算出された特異値トリプレットの精度に依存するので、本発明は、従来の技術においてこれまで開示されている方法よりもより良好な結果を与えることが可能となる。
【００１９】
さらに、本発明は、ドキュメント－アトリビュート行列のＳＶＤに基づいたアップデートの正確な計算を可能とするので、本発明は、例えば新たなトピックス／事象の検出といった、内容の著しく変化するデータベースにおけるドキュメントといったデータの検出に使用することができる。また、本発明の方法は、例えば特願２０００－１７５８４８号、特願２００１－１５７６１４号にも記載されているトピックス／事象の追跡にも適用することができる。
【００２０】
すなわち、本発明によれば、データベースに時間の経過と共に追加されるドキュメントを検索するための情報検索システムであって、前記ドキュメントは、ドキュメント－アトリビュート行列へと変換されて前記情報検索システムに保持され、かつ前記ドキュメント－アトリビュート行列は、逐次的に追加されるドキュメント－アトリビュート副行列から構成され、
前記ドキュメント－アトリビュート行列から所定の行列を生成して保持させるための手段と、
前記データベースに所定の期間の間に追加されたドキュメントからドキュメント－アトリビュート副行列を生成するための手段と、
前記ドキュメント－アトリビュート副行列に関連する情報を使用して前記所定の行列を更新し、更新された前記所定の行列を特異値分解して、データベースに保持されたすべてのドキュメント－アトリビュート行列の次元削減を実行するための手段と、
前記次元削減されたドキュメント－アトリビュート行列を使用してユーザが入力したクエリーによる情報検索を行うための手段と
を含む、情報検索システムが提供される。
【００２１】
本発明においては、前記ドキュメント－アトリビュート副行列をＱＲ分解し、前記所定の行列として行列Ｒのみを使用する手段を含むことができる。本発明においては、前記保存された行列Ｒの更新を、前記ドキュメント－アトリビュート副行列に関連したハウスホルダー変換により実行させるための手段を含むことができる。
【００２２】
本発明においては、前記ドキュメント－アトリビュート副行列から前記所定の行列として共分散行列を生成し、ドキュメント・ベクトルの積和行列（ＳＵＭ（Ｍ）$_1$）と、ドキュメント・ベクトルの平均（ＭＥＡＮ（Ｍ）$_1$）と、ドキュメント・ベクトルの平均の積行列（ＳＵＭ（Ｍ）$_2$）と、ドキュメントの全数（Ｍ）とを保持するための手段とを含むことができる。また、本発明においては、追加された前記副行列を含むすべてのドキュメント－アトリビュート行列の共分散行列Ｃ’を、下記式

$$\mathbf{C}' = \frac{1}{(M+H)}\ \mathrm{SUM}(M+H)_1 - \mathrm{SUM}(M+H)_2$$

（上記式中、Ｈは、追加された副行列のドキュメント数を示す。）
により生成する手段を含むことができる。本発明においては、さらに、ドキュメント・ベクトルに含まれるアトリビュートを自動的に検索し、アトリビュート・ハッシュ・テーブルを生成して前記アトリビュートを追加または削除するための手段を含むことができる。
【００２３】
本発明によれば、データベースに時間の経過と共に追加されるドキュメントを検索するた

めの情報検索方法であって、前記ドキュメントは、ドキュメント－アトリビュート行列へと変換されて保持され、かつ前記ドキュメント－アトリビュート行列は、逐次的に追加されるドキュメント－アトリビュート副行列から構成され、
前記ドキュメント－アトリビュート副行列から所定の行列を生成して保持させるステップと、
前記データベースに所定の期間の間に追加されたドキュメントからドキュメント－アトリビュート副行列を生成するステップと、
前記ドキュメント－アトリビュート副行列に関連する情報を使用して前記所定の行列を更新し、更新された前記所定の行列を特異値分解して、データベースに保持されたすべてのドキュメント－アトリビュート行列の次元削減を実行するステップと、
前記次元削減されたドキュメント－アトリビュート行列を使用してユーザが入力したクエリーによる情報検索を行うステップと
を含む、情報検索方法が提供される。

【００２４】

さらに本発明によれば、データベースに時間の経過と共に追加されるドキュメントを検索するための情報検索方法を実行させるためのプログラムであって、前記ドキュメントは、ドキュメント－アトリビュート行列へと変換されて保持され、かつ前記ドキュメント－アトリビュート行列は、逐次的に追加されるドキュメント－アトリビュート副行列から構成され、
前記ドキュメント－アトリビュート副行列から所定の行列を生成して保持させるステップと、
前記データベースに所定の期間の間に追加されたドキュメントからドキュメント－アトリビュート副行列を生成するステップと、
前記ドキュメント－アトリビュート副行列に関連する情報を使用して前記所定の行列を更新し、更新された前記所定の行列を特異値分解して、データベースに保持されたすべてのドキュメント－アトリビュート行列の次元削減を実行するステップと、
前記次元削減されたドキュメント－アトリビュート行列を使用してユーザが入力したクエリーによる情報検索を行うステップと
を含む、情報検索方法を実行させるためのプログラムが提供できる。

【００２５】

また、本発明によれば、データベースに時間の経過と共に追加されるドキュメントを検索するための情報検索方法を実行させるためのプログラムが記録されたコンピュータ可読な記録媒体であって、前記ドキュメントは、ドキュメント－アトリビュート行列へと変換されて保持され、かつ前記ドキュメント－アトリビュート行列は、逐次的に追加されるドキュメント－アトリビュート副行列から構成され、
前記ドキュメント－アトリビュート副行列から所定の行列を生成して保持させるステップと、
前記データベースに所定の期間の間に追加されたドキュメントからドキュメント－アトリビュート副行列を生成するステップと、
前記ドキュメント－アトリビュート副行列に関連する情報を使用して前記所定の行列を更新し、更新された前記所定の行列を特異値分解して、データベースに保持されたすべてのドキュメント－アトリビュート行列の次元削減を実行するステップと、
前記次元削減されたドキュメント－アトリビュート行列を使用してユーザが入力したクエリーによる情報検索を行うステップと
を含む、情報検索方法を実行させるためのプログラムが記録された記録媒体が提供される。

【００２６】

【発明の実施の形態】

以下、本発明につき図面に示した実施の形態に基づいて詳細に説明するが、本発明は後述する特定の実施の形態に限定されるものではない。

【００２７】
図１は、本発明において、情報検索を実行するためのデータベースの概略構成を示した図である。図１に示されたデータベースにおいては、ドキュメントは、バイナリ・モデルやアトリビュート頻度・モデルといった適切な方法を使用して、ドキュメント・ベクトルへと変換されている。図１には、上述したドキュメントにより生成されたドキュメント－アトリビュート行列としてデータベースのデータ構成を示している。図１においては、行方向にドキュメント・ベクトルが並べられており、列方向には、所定のアトリビュートがドキュメントに含まれている場合には、適切な重み、所定のアトリビュートが含まれていない場合には、０とする方法により、数値要素が並べられている。
【００２８】
なお、本発明においては、図１で示したバイナリ・モデルの他に、ユーザが指定する重み付け因子（ウエイト・ファクタ）を適用した、アトリビュート頻度モデル与えることもできる。以下、本発明においては、上述して得られた行列をドキュメント－アトリビュート行列Ａとして参照する。アトリビュートとしては、テキスト・ドキュメント・データに用いられるキーワード・アトリビュートの他にも、タイム・スタンプ、画像、オーディオ・データなど、いかなるアトリビュートでも本発明においては使用することができる
【００２９】
図１に示されるように、本発明においては、データベースには、ドキュメント／データが頻繁に追加されており、これらのドキュメントは、それぞれが含むアトリビュートに基づいて、ドキュメント・ベクトルへと変換される。これらのドキュメント・ベクトルは、例えば、日単位、週単位、月単位、或いは、データベースの管理者が設定した単位に区切られて、それぞれの副行列Ａ１，．．．，Ａｎとして構成されている。
【００３０】
例えば、最初にデータベースに対して蓄積されていたドキュメント・ベクトルの集合は、行列Ａとして区切られており、Ａについて処理を実行する。その後追加されたドキュメント・ベクトルの集合を、例えば１週間といった所定の期間まとめてドキュメント－アトリビュート行列として構成したものが、副行列Ａ１として示されている。同様にして、順次追加されたドキュメント・ベクトルを所定の期間ごとにまとめたもので、最も新しい副行列が、図１においては副行列Ａｎとして示されている。
【００３１】
上述した所定の期間としては、上述したように日単位、週単位、月単位としてまとめることができるが、特に常に一定の期間ではなく、必要に応じてその時点までに蓄積されたデータをまとめて副行列Ａｉとすることができる。本発明は、上述した副行列を使用して、ＱＲ分解で得られる上三角行列Ｒまたは共分散行列を算出することで、データが蓄積されていくドキュメント－アトリビュート行列すべてを一割してＳＶＤを実行させる労力を削減することにより、次元削減の計算時間を低減して効率化を達成する。また、本発明においては、重要ではあるがデータ数として数％程度でデータベースに含まれるいわゆるアウトライア・ドキュメントを次元削減プロセスにおいて無視してしまう可能性を可能な限り低減させ、検索精度を向上させることを可能とする。以下、本発明の実施の形態について詳細に説明する。
【００３２】
＜第１の実施の形態＞
本発明の第１の実施の形態においては、上述した副行列と前回までに得られた行列Ｒからなる行列に対してＱＲ分解を適用して、新たに得られる上三角行列Ｒに、ＳＶＤを適用して特異ベクトルを得、得られた大きなものからｋ番目までの特異ベクトルを含んで構成された特異行列を使用して、データベースに含まれるドキュメント－アトリビュート行列をｋ次元へと次元削減させるものである。
【００３３】
また、副行列Ａ１は、図２で示されるように、もとのドキュメント－アトリビュート行列ＡをＱＲ分解した時の行列Ｒの底に加えた形式で保持する。

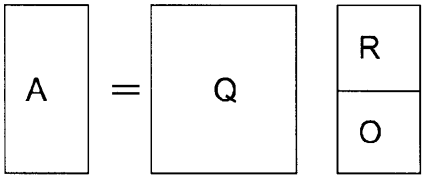
【００３４】
図２の左側の行列Ｒに副行列Ａｉが追加された部分は適当な直交置換行列Ｐにより行列Ｒの直下に移動させる。
【００３５】
図３は、本発明の第１の実施の形態としてＱＲ分解法を使用してドキュメント－アトリビュート行列Ａの次元削減を実行するプロセスを示したフローチャートである。図３に示されるように、プロセスは、ステップＳ１から開始し、ステップＳ２において、ドキュメント－アトリビュート行列ＡのＱＲ分解を実行する。ステップＳ３においては、ＱＲ分解して得られた行列Ｑと、行列Ｒのうち、行列Ｒのみを使用する。本発明において行列Ｒのみを使用する理由は、（ａ）Ｒ行列が上三角行列であり、ＳＶＤを行うためにきわめて迅速に計算を実行することができること、（ｂ）下記式に示されるように行列Ｒを使用してもドキュメント－アトリビュート行列における特異値または固有値は保存されていること、の理由に基づくものである。
【００３６】
【数２】

$$A^{T}A=(QR)^{T}QR=R^{T}Q^{T}QR=R^{T}R$$

【００３７】
次いで、本発明の第１の実施の形態におけるプロセスにおいては、ステップＳ４において追加された副行列Ａｉを使用して行列Ｒを更新する。この際、更新された行列Ｒの算出は、本発明の好適な実施の形態においては、この更新された行列Ｒの算出については、より詳細に後述する。ステップＳ５では、上述したようにして得られたＲ行列を使用してＳＶＤを実行する。
【００３８】
【数３】

$$R=U\Sigma V^{T}$$

（上記式中、Ｕ、Ｖは、Ｎ×Ｎ正規直交行列であり、Σは、Ｎ×Ｎ対角行列である。）
上述のようにして得られた行列Ｒの特異値または固有値は、上述したようにドキュメント－アトリビュート行列Ａの特異値と同じ特異値を保持している。また、Ｒは、上三角行列であるため、ＳＶＤをきわめて容易に行うことができ、従来のプロセスにおいてきわめて計算時間を要したＳＶＤに割り当てられる計算時間を著しく低減することが可能となる。
【００３９】
さらに本発明のプロセスにおいては、ステップＳ６で、得られた特異値から特異ベクトルまたは固有ベクトルを得、得られた特異ベクトルまたは固有ベクトルを、特異値または固有値の大きな方から特異ベクトルまたは固有ベクトルのｋ番目までを使用して次元削減したｋ次元の特異行列を生成してドキュメント－アトリビュート行列の次元を削減させる。
【００４０】
ステップＳ７において次元の減少した行列を使用して情報検索を実行し、ステップＳ８において本発明の第１の実施例の情報検索方法を終了する。
【００４１】
以下各ステップにおける処理を詳細に説明する。まず、ステップＳ２で、ドキュメント－アトリビュート行列ＡについてＱＲ分解を実行する。ＡについてのＱＲ分解を下記式に示す。
【００４２】
【数４】



【００４３】
上記式中、Ｏで示された行列は、要素がすべて０の行列を意味する。図４には、一般的な行列Ｄに対する上述したＱＲ分解を実行させるための擬似コードを示す。
本発明においては、ステップＳ３において、行列Ａについて上述のＱＲ分解により得られた行列Ｒのみを使用する。図５には、行列ＡのＱＲ分解により得られる行列の構成要素を概略的に示す。図５において、Ｍは、ドキュメントの数であり、Ｎは、アトリビュートの数である。また、図５においては、行列Ｑの列ベクトルをq_i（ｉ＝１、．．．、Ｍ）により示し、行列Ｒの要素がゼロの部分を０で示している。また、図５に示した実施の形態においては、得られる行列Ｒは、Ｎ×Ｎの上三角行列として得られている。
【００４４】
本発明においてはさらに、ステップＳ４において、副行列Ａｉとしてデータが加えられたドキュメント－アトリビュート行列につき、ハウスホルダー変換を実行する。
【００４５】
本発明においては、ステップＳ５において行列Ｒに対して直接ＳＶＤを適用して、Ｒ行列の特異値または固有値を求める。この際、ＳＶＤに使用することができる方法としては、これまで知られた種々の方法を使用することができ、例えばハウスホルダー変換を使用する方法、またはランチョス法を使用することができる。
【００４６】
さらに本発明においては、ステップＳ６においてＳＶＤ計算により得られた特異値または固有値から特異ベクトルまたは固有ベクトルを生成し、特異値または固有値の大きな方から所定の数の特異ベクトルまたは固有ベクトルを降順に配置して特異行列を形成させ、これをドキュメント－アトリビュート行列Ａに乗じて、下記式（５）にしたがってドキュメント－アトリビュート行列Ａの次元を削減させる。
【００４７】
【数５】

$$Ak = Uk \quad \Sigma k \quad Vk^{T}$$

図６に示すように上記式中、Ａｋは、ｋ番目までの特異値を使用して得られた次元削減されたドキュメント－アトリビュート行列であり、Σ_kはｋ個の特異値から成るｋ×ｋ次元の対角行列であり、U_kはｋ個の特異値に対する左特異ベクトルから成る行列であり、Ｖｋはｋ個の特異値に対する右特異ベクトルから成る行列である。
【００４８】
ステップＳ７においては、上述したようにして次元削減されたドキュメント－アトリビュート行列を使用して、クエリー・ベクトルとの乗算を実行し、ユーザが所望するクエリー・ベクトルに基づいた情報検索を実行させることになる。
【００４９】
本発明の第１の実施の形態におけるドキュメント－アトリビュート行列Ａの次元削減は、上述したようにＱＲ分解により得られた行列Ｒを更新しつつ、直接ＳＶＤに提供し、精度良くＳＶＤを実行することができると共に、次元削減に対する計算時間を著しく低減させることができ、この結果、メモリ資源を節約しつつ、高精度、高効率の情報検索を実行することが可能となる。
【００５０】
＜第２の実施の形態＞

本発明の第２の実施の形態においては、共分散行列を使用した次元削減プロセスにおいて、すでに計算された以前のドキュメント－アトリビュート行列Ａに対する共分散行列を使用して新たに加えられた副行列Ａ１を含む新たな共分散行列を生成し、得られた副行列Ａ１を反映した新たな共分散行列をＳＶＤプロセスに提供するものである。本発明の第２の実施の形態は、ドキュメントの追加に対応できるばかりではなく、容易にドキュメントの削除についても適用することができる。

【００５１】

以下、本発明の第２の実施の形態について詳細に説明する。本発明の第２の実施の形態を詳細に説明する前に、共分散行列を用いた情報検索について概略的な説明を行う。Ｍ×Ｎの要素からなるドキュメント－アトリビュート行列Ａに対して、その共分散行列Ｃは、下記式により与えられる。

【００５２】

【数６】

$$\mathbf{C} = \frac{1}{M}\sum_{i=1}^{M}\mathbf{d}_i\mathbf{d}_i^{\mathrm{T}} - {}^{\mathrm{bar}}\mathbf{d}(M)\,{}^{\mathrm{bar}}\mathbf{d}(M)^{\mathrm{T}} = \frac{1}{M}\,\mathrm{SUM}(M)_1 - \mathrm{SUM}(M)_2$$

上式中、ＳＵＭ（Ｍ）$_1$は、ドキュメント・ベクトルの積和行列であり、ＳＵＭ（Ｍ）$_2$は、ドキュメント・ベクトルの平均の積行列である。ｄ$_i$、^{bar}ｄ（Ｍ）、^{bar}ｄ（Ｍ）$_i$は、それぞれドキュメント－アトリビュート行列Ａの要素ｉ、ｊを使用して下記式で定義される。このうち、ＳＵＭ（Ｍ）$_1$およびＳＵＭ（Ｍ）$_2$は、共にＮ×Ｎの対称な正方行列である。

【００５３】

【数７】

$$\mathbf{d}_i = [d(i,1),\ldots,d(i,N)]^T$$

$$^{\mathrm{bar}}\mathbf{d}(M) = [\,^{\mathrm{bar}}d(M)_1,\ldots,\,^{\mathrm{bar}}d(M)_N]^T$$

$$^{\mathrm{bar}}d(M)_i = \frac{1}{M}\sum_{j=1}^{M}d(j,i)$$

【００５４】

上述したように定義された共分散行列は、Ｎ×Ｎの正方行列として得られる。また、共分散行列の特異値または固有値は、例えば特願２０００－１７５８４８号に記載されているように、ドキュメント－アトリビュート行列の特異値または固有値を保存し、かつ正方行列であるのでドキュメントが著しく多い場合であっても特異値または固有値、ひいては特異ベクトルまたは固有ベクトルを迅速に計算でき、高い効率の情報検索の実行を可能とする。

【００５５】

図７は、本発明の第２の実施の形態の情報検索方法のフローチャートを示した図である。なお、本発明の第２の実施の形態においてもドキュメント－アトリビュート行列構成は、図１に示したようにＡおよび逐次的に蓄積されて行く副行列Ａ１、Ａ２，．．．，Ａｎから構成されているものとして説明する。以下に説明する実施の形態においては、まず最初に行列Ａから共分散行列を生成して行くものとして説明する。

【００５６】

図７に示されるように、本発明の情報検索方法の第２の実施の形態は、ステップＳ１０から開始し、ステップＳ１１において行列Ａから上記式にしたがって共分散行列を生成する。

【００５７】

ステップＳ１２においては、Ｈ個のドキュメント・ベクトルを含む副行列Ａ１が、行列Ａ

に追加され、ドキュメント－アトリビュート行列が形成される。ステップＳ１３においては、副行列Ａ１のドキュメント－アトリビュート行列から、下記式の計算を実行させる。

【００５８】

【数８】

$$\mathrm{SUM}(H)_1 = \sum_{i=1}^{H} \mathbf{d}_{\mathbf{M+i}} \mathbf{d}_{\mathbf{M+i}}^{\mathbf{T}}$$

さらにステップＳ１４において、下記式の計算を実行させる。

【００５９】

【数９】

$$\text{SUM-MEAN}(H)_1 = \sum_{j=1}^{H} d(M+j,i) = H\,\mathrm{MEAN}(H)_1$$

次いで、ステップＳ１５において、下記式の計算を実行させる。

【００６０】

【数１０】

$$\mathrm{SUM}(M+H)_1 = \mathrm{SUM}(M)_1 + \mathrm{SUM}(H)_1$$

【００６１】

ここで、下記式の関係が各構成要素について成り立つ。

【００６２】

【数１１】

$$\overset{\text{bar}}{d}(M+H)_i = \frac{1}{M+H}\sum_{j=1}^{M+H} d(j,i) = \frac{1}{M+H}\left(M\overset{\text{bar}}{d}(M)_i + H\overset{\text{bar}}{d}(H)_i\right)$$

【００６３】

次いで、下記式を使用して、ＳＵＭ－ＭＥＡＮ（Ｍ＋Ｈ）$_1$を更新する。この際の計算は、単なるＮ回の加算により実行できるので計算時間を短縮することができる。

【００６４】

【数１２】

$$\text{SUM-MEAN}(M+H)_1 = \text{SUM-MEAN}(M)_1 + \text{SUM-MEAN}(H)_1$$

【００６５】

次いで、ステップＳ１６においてＳＵＭ（Ｍ＋Ｈ）$_2$を、下記式にしたがってＳＵＭ－ＭＥＡＮ（Ｍ＋Ｈ）$_1$を使用して更新する。

【００６６】

【数１３】

$$(M+H)^2\mathrm{SUM}(M+H)_2 = \text{SUM-MEAN}(M+H)_1\text{SUM-MEAN}(M+H)_1^T$$

この更新の後、アップデートされた共分散行列Ｃ’を、下記式にしたがって得ることができる。

【００６７】

【数１４】

$$\mathbf{C}' = \frac{1}{(M+H)}\ \mathrm{SUM}(M+H)_1 - \mathrm{SUM}(M+H)_2$$

この後、ステップＳ１７において更新された共分散行列Ｃ’を、ＳＶＤに提供して特異値または固有値を得、大きな方からｋ個の特異値または固有値に対応する特異ベクトルまたは固有ベクトルを選択して次元の減少された特異行列または固有行列を生成する。ステッ

プＳ１８において、次元の減少されたｋ本の特異行列または固有ベクトルまたは特異ベクトルを使用してドキュメント－アトリビュート行列Ａの次元削減を行い、ステップ１９で情報検索を実行させ、ステップＳ２０で、本発明の第２の実施の形態のプロセスを終了する。図８には、上述した本発明のプロセスのうち、ステップＳ１１において説明した共分散行列の生成のための擬似コードを示す。

【００６８】

また、本発明においては、上述したプロセスにおいてＳＵＭ（Ｍ－Ｈ）$_1$を生成させて同様に計算を繰り返すことにより、ドキュメント・ベクトルが何らかの理由により削減された場合にも容易に対応することが可能となる。

【００６９】

さらに、本発明の第２の実施の形態においては、アトリビュート自体のアップデートおよびダウンデートを行うことも可能となる。アトリビュートのアップデートは、新たなアトリビュートが加えられた場合に実行され、アトリビュートの削減は、例えばアトリビュートが検索において非現実的なものとなったとき、または検索する必要が無くなったときに実行される。アトリビュートの追加・削除は、アトリビュート・ハッシュ・テーブルを使用して実行される。

【００７０】

図９には、本発明の第２の実施の形態に適用される、ドキュメントの追加または削除におけるアトリビュート・ハッシュ・テーブルを変更するプロセスを示した図である。図９（ａ）に示すように、ユーザからのドキュメント（ｉ）の追加／削除の要求があると、本発明においては、ドキュメント（ｉ）を追加する場合には、まずバイナリ・モデルといった適切な方法を使用してドキュメント・ベクトルを形成する。また、ユーザがドキュメント（ｉ）を削除しようとする場合には、削除しようとするドキュメント（ｉ）を特定する。

【００７１】

次いで、図９（ｂ）に示すようにドキュメント（ｉ）に含まれる非ゼロのアトリビュートＡＴを特定する。図９（ｂ）においては、ドキュメント（ｉ）が含む非ゼロのアトリビュートａｔｔ３，ａｔｔｎ－１が、数値１に対応することが示されているが、本発明において重み付けをそれぞれのアトリビュートＡＴに対して適用する場合には、非ゼロの要素は、ウエイト・ファクタに対応した１以外の値とされていても良い。

【００７２】

上述したようにドキュメント（ｉ）における非ゼロのアトリビュートが特定されると、本発明においては、図１０に示されたアトリビュート・ハッシュ・テーブルを参照する。

【００７３】

アトリビュート・ハッシュ・テーブルには、それぞれのアトリビュートと、当該アトリビュートを含むドキュメント数とが対応して記憶されており、アトリビュートからドキュメント数を参照することが可能とされている。図１０を使用してドキュメントの追加／削除の実施の形態について説明すると、例えば、アトリビュート３とアトリビュートｎ－１とを含むドキュメント（ｉ）が追加／削除される場合には、もともとアトリビュート３を含むドキュメント数６が、追加の場合にはドキュメント数７とされ、削除の場合にはドキュメント数５と変更される。

【００７４】

これに対応して、図１０に示される実施の形態の場合には、アトリビュートｎ－１も非ゼロの要素となっているので、もともとアトリビュートｎ－１に関連するドキュメントのドキュメント数３３が、追加／削除に対応して３４または３２へと変更される。上述したアトリビュート・ハッシュ・テーブルは、さらに別の識別子を使用することにより所定のアトリビュートを有する個々のドキュメントの参照を可能とするようにされていてもよい。

【００７５】

図１１は、上述したアトリビュートの削除を実行させる場合の概念図を示す。図１１（ａ）が、アトリビュート削除前のドキュメント・ベクトルの構成を示したものであり、図１１（ｂ）がアトリビュート削除後のドキュメント・ベクトルの構成を示した実施の形態を

示した図である。図１１に示した実施の形態においては、アトリビュート４が削除される実施の形態を示しているが、本発明において削除されるアトリビュートの示された位置、または一度に削除されるアトリビュートの数は、図１１に示される以外にもいかなるものでも用いることができる。

【００７６】

図１２は、図１１に示されたアトリビュート４を削除する場合のドキュメント・ベクトルの積和行列の変更を例示した図である。図１２（ａ）は、アトリビュート４の削除前のドキュメント・ベクトルの積和により得られる積和行列を示し、図１２（ｂ）は、アトリビュート４の削除後のドキュメント・ベクトルから得られる積和行列を示す。上述したように、本発明においては、共分散行列を生成する際に積和行列を用いるので、アトリビュートの追加、削除も容易に含めることができる。

【００７７】

図１３は、アトリビュートを加える際の処理を説明した概略図である。図１３に示した実施の形態においては、図１３（ａ）にアトリビュート追加前のドキュメント・ベクトルが示されており、このドキュメント・ベクトルに対して、図１３（ｂ）に示されるように、アトリビュートｎ＋１がドキュメント・ベクトルに対して加えられる。図１４には、上述したように加えられたアトリビュートｎ＋１を含んだドキュメント・ベクトルから積和により形成された（Ｎ＋１）×（Ｎ＋１）行列を示す。

【００７８】

上述した従来の共分散行列を作る場合のドキュメント・データは、ドキュメント数とアトリビュート数とがあらかじめわかっていることが前提とされていた。これに対して本発明は、文書数の総数もアトリビュート数も、最初はわかってない状態から計算を逐次実行させることを可能とする。

【００７９】

図１５には本発明の情報検索方法を実施するためのコンピュータ・システムの概略図を示す。図１５に示した本発明のコンピュータ・システムは、コンピュータ１０と、このコンピュータ１０とデータの伝送を行うことが可能であるように接続されたデータベース１２とを含んで構成されている。本発明において使用することができるコンピュータ１０としては、本発明の方法を実行することができる中央処理装置（ＣＰＵ）、ＲＡＭなどのメモリなどのハードウエア資源を含んで構成されている限り、パーソナル・コンピュータ、ワークステーションといったいかなるものでも用いることができる。また、本発明において使用することができるデータベース１２としては、データが追加して書込みができるものであれば、従来知られているいかなるデータベースでも使用することができる。

【００８０】

また、図１５に示したコンピュータ・システムは、例えばインターネット、ローカル・エリア・ネットワーク（ＬＡＮ）、ワイド・エリア・ネットワーク（ＷＡＮ）といった、これまで知られたいかなるＴＣＰ／ＩＰといったプロトコルを使用して遠隔的に配置されネットワーク１４を介して接続されたコンピュータ１６と通信を行うことができるように構成することができる。図１５に示した本発明の実施の形態においては、データベース１２に接続されたコンピュータ１０をサーバとして使用し、このサーバに対して遠隔的に接続されたコンピュータ１６をクライアント・コンピュータとして使用する、いわゆるクライアント－サーバ・システムとして構成することができる。クライアント・コンピュータのユーザは、所望する情報を検索するべく、コンピュータ１６へと例えばキーワードといったアトリビュートの入力を行う。

【００８１】

入力されたキーワードは、ネットワーク１４を介してコンピュータ１０へと伝送され、コンピュータ１０において情報検索を行うために使用される。データベース１２には、オリジナルのドキュメントＤが保持されている。また、データベース１２には、このドキュメントＤから所定のアトリビュートを抽出し、例えばバイナリ・モデルを使用して予め数値化されたドキュメント－アトリビュート行列も保持されている。本発明においては、上述

したドキュメント－アトリビュート行列は、一度すでに本発明によるＱＲ分解法または共分散行列法などの方法を使用してすでに次元削減が行われて、実際の情報検索に提供されている。

【００８２】

図１５に示されたデータベース１２には、随時にドキュメントＤＮが追加されていて、副行列として蓄積されて行く。このドキュメントＤＮは、本発明においては、例えば日単位、週単位、月単位、あるいはサーバ管理者の規定する間隔で、前回ＳＶＤを実行した時点からその時点までに蓄積された分ごとに上述したＱＲ分解法または共分散行列法により、以前に得られているＲまたはＳＵＭ（Ｍ）$_1$，ＳＵＭ（Ｍ）$_2$を使用することで、効率よく以前の結果を含めて次元削減が実行される。本発明においては、上述したようにしてアップデートまたはダウンデートされたドキュメントに対しての情報検索を行うために使用されるドキュメント－アトリビュート行列の効率的な次元削減を可能とし、高精度、かつ高効率の情報検索システムが提供されている。

【００８３】

また図１５においては、情報検索システムを特にクライアント－サーバ・システムとして説明したが、本発明においては、特にネットワークを介して接続されたクライアント－サーバ・システムばかりではなく、スタンド・アローンのパーソナル・コンピュータまたはワークステーションを使用した情報検索システムとすることができる。

【００８４】

以下、本発明をより具体的に実施例をもって説明するが、本発明は後述する実施例によって限定されるものではない。

【００８５】

【実施例】

（実施例１）

本発明のＱＲ分解法を用いた情報検索について、表１に一部を示したサンプル・データベースを使用して、本発明の効果について検討を加えた。サンプル・データベースは、ドキュメント数３００、キーワード数５０の３００×５０のドキュメント－キーワード行列として構成した。

【００８６】

表１に示したサンプル・データベースは、現実に使用されるデータベースよりも遙かに小さいので、計算速度においては、従来例で知られた方法に比較して大きな違いは見られなかった。このため、本発明の方法により得られた結果と、後述する比較例により得られた結果との計算精度の比較を行った。

【００８７】

【表１】

300 50

1 1.65244e-05 7 0.19628 8 3.61624e-08 9 0.00018 10 0.01373 12 2.23141e-05
14 4.64130e-05 15 1.78852e-05 19 2.19083e-10 20 0.00028 23 1.21350e-07 24
9.98935e-10 25 7.85754e-07 28 1.00379e-05 29 2.20989e-05 39 8.26993e-08 40
3.46313e-09 42 3.42076e-10 43 1.40159e-05 45 1.12157e-06 50 5.29161e-07
5 1.95247e-05 6 9.65067e-09 11 4.55447e-07 12 0.00840 14 4.10627e-09 16
2.13934e-10 17 0.00022 22 0.04135 27 3.80890e-09 31 4.73880e-10 32
9.24676e-05 34 6.28455e-06 40 2.17201e-05 41 2.23796e-05 42 1.28397e-06 46
1.50510e-08 47 1.45181e-06 50 9.61504e-09
2 2.20380e-05 3 9.09460e-06 7 0.01776 8 0.02260 10 4.63745e-06 12 0.00147
13 0.00493 14 1.09471e-06 15 0.00366 17 1.82302e-05 18 2.04803e-05 20
2.29817e-05 22 3.91119e-09 25 3.34495e-06 26 0.02291 34 0.00014 37
7.98834e-09 39 1.33136e-07 40 5.08289e-08 41 9.11475e-09 42 3.64340e-05 43
5.91690e-10 44 8.69801e-09 45 0.01932 46 1.31295e-06 47 4.29275e-08 49
1.30851e-10
4 1.62408e-07 7 4.75727e-06 8 2.24351e-05 9 4.66984e-06 10 3.25484e-06 12
2.15089e-05 14 5.29682e-06 17 1.63133e-10 26 0.00766 28 7.18088e-09 31
1.53839e-09 34 8.06182e-10 39 6.13628e-08 41 4.69203e-06 45 2.34841e-10 46
8.81336e-07 47 1.23390e-05 48 3.38329e-09 50 3.24139e-08
2 4.46380e-10 5 5.26576e-09 7 2.09751e-07 12 7.67514e-06 13 2.75356e-05 15
2.15944e-09 18 3.78191e-07 19 8.21337e-09 21 0.18159 27 1.10242e-08 28
2.09905e-05 32 1.38427e-07 34 1.04496e-06 37 2.14666e-08 38 0.00268 40
0.00482 45 6.75967e-09 46 9.43715e-07 47 1.75043e-08

・
・
・

【００８８】
得られた計算精度を実施例１、比較例１、比較例２の計算結果と共に以下に示す。なお、特異値計算の精度を比較する簡単な方法として、最大の特異値σ_1と最小の特異値σ_nの比（以下、この値をコンディション・ナンバーとして参照しｃとで示す）を採用した。なお、比較例１は、ドキュメント・アトリビュート行列Aに更新された副行列Ａｉを加えたものを新たにＡとして、Ａ全体に対してＳＶＤ法を適用して計算を行うもの（以降、ナイーブなＳＶＤ法と呼ぶ）であり、比較例２は、$A^T A$の逐次アップデート法によりＳＶＤを更新する方法である。また、データベースの更新をシミュレートするため、３００のドキュメント－キーワード行列を５０×５０チャンクに区切り、チャンク毎に所定の方法を用いてＳＶＤを実行させた。
【００８９】
実施例および比較例とも、Ｐｅｎｔｉｕｍ（登録商標）　ＩＩＩ、クロック周波数７３３ＭＨｚ（Ｉｎｔｅｌ社、登録商標）を使用したＷｉｎｄｏｗｓ（登録商標）　２０００（

Ｍｉｃｒｏｓｏｆｔ社、登録商標）をＯＳとして使用するパーソナル・コンピュータを使用した。また、データは、６４ビット浮動小数点精度を使用して計算を実行した。

【００９０】

上述した条件の下で得られた結果を、コンディション・ナンバーの比として示す。

【００９１】

c_1＝１０７３２．７１４５７０２２３１８３（実施例１）

c_2＝１０７３２．７１４５７０２２３２１７（比較例１）

c_3＝１０７３２．７１４５７０４１１２３４（比較例２）

という結果が得られた。

【００９２】

したがって、本発明によるＱＲ分解法を使用してＳＶＤを実行させる場合には、ナイーブなＳＶＤ法を使用して得られる結果c_2と本発明による方法c_1との比が、c_2／c_1＝１．０００００００００００００００３となり、小数点以下９桁まで一致していることが示された。一方で、c_2／c_3の比は、０．９９９９９９９９９９８２４８１９となって、精度が本発明の方法よりも劣ることが示された。最小の特異値σ_nについてみれば、実施例１の方法と、比較例１の方法とは、小数点以下１５桁まで一致していることが見出された。実施例１の結果から、本発明の方法は、ほぼマシンエプシロンまで精度的に問題ないことが示された。

【００９３】

（実施例２）

実施例２においては、本発明のＱＲ分解法を使用する実施の形態において算出するデータのスケーラビリティをあげて、ＣＰＵの全占有時間、反復計算時間からみた計算速度およびメモリ資源の必要量の点について検討を加えた。使用したデータは、

(a) 100x100、

(b) 1000x100、

(c) 10000x100、

(d) 100000x100、

(e) 1000000x100

の５つの行列を使用した。行列の要素は、ランダムに生成した密行列として作成した。計算にあたって使用したチャンクのサイズは、いずれも１００×１００とした。したがって、たとえば(e)で示される場合には、１万回反復してアップデートを行いながら計算を実行させることになる。また、本発明の方法を実行するプログラムは、言語Ｊａｖａ（登録商標）で実装した。下記表に示すメモリ使用量のうち、概ね４ＭＢがＪａｖａ（登録商標）（Ｓｕｎ　Ｍｉｃｒｏｓｙｓｔｅｍｓ社の登録商標）のＶｉｒｔｕａｌ　Ｍａｃｈｉｎｅが占める。したがって、データが使用する実質のメモリサイズは、下記表４におけるメモリ使用量から４ＭＢを引いた値と見積もることができる。

【００９４】

【表２】

CPU total	(a)	(b)	(c)	(d)	(e)
比較例1	454 ミリ秒	3 秒 922 ミリ	92 秒 843 ミリ	1040 秒 391 ミリ	N/A
比較例2	453 ミリ秒	1 秒 203 ミリ	8 秒 781 ミリ	106 秒 969 ミリ	764 秒 203 ミリ
実施例2	578 ミリ秒	1 秒 922 ミリ	12 秒 297 ミリ	126 秒 983 ミリ	1291 秒 532 ミリ

【００９５】

【表３】

CPU／反復	(a)	(b)	(c)	(d)	(e)
比較例1	454 ミリ秒	3 秒 922 ミリ	92 秒 843 ミリ	1040 秒 391 ミリ	N/A
比較例2	453 ミリ秒	120.3 ミリ	87、8 ミリ	106.9 ミリ	76.4 ミリ
実施例2	578 ミリ秒	192.2 ミリ	122.9 ミリ	126.9 ミリ	129.1 ミリ

【００９６】
【表４】

メモリ使用量	(a)	(b)	(c)	(d)	(e)
比較例1	5.9MB	8.4MB	31.0MB	251.6MB	割り当て不能
比較例2	6.5MB	6.5MB	6.5MB	6.5MB	6.5MB
実施例2	6.5MB	6.5MB	6.5MB	6.5MB	6.5MB

【００９７】
計算時間の点でいえば、比較例２で示されるＡ^{T} Ａを反復して特異値を求めるのが最速の結果を与えた。このことは理論的にも明らかで、比較例２ではＡ^{T} Ａの計算における行列－行列積の計算と行列－行列和の計算だけで済むためである。
【００９８】
一方、各反復計算の時間についてみれば、実施例２で示される本発明の方法は、ＱＲ分解の時間が必要とされる分だけ、反復時間を要することになることが示された。より正確にいえば、実施例２で使用した密行列に対してチャンク・サイズをＨとしたとき、比較例２では、乗算に$O(HN^2)$の計算量が必要となる。一方、実施例２で用いるＱＲ分解法は、一回につき、$O(2N^2((H+N)-N/3)$の計算量を必要とする(Golub & Van Loan)。
【００９９】
ただし、実施例１においても説明したように、比較例２のＡ^{T} Ａの反復繰り返し計算は、速度を優先し、誤差を犠牲にするものである。なお、データが１００×１００の場合（上述した（ａ）の行列の場合）は、反復がないので計算手法そのものに要する時間の差を示しているといえる。一回の反復におけるＣＰＵ時間は、比較例２で概ね平均１００ミリ秒程度、実施例２では、１３０ミリ秒程度となる。しかしながら、比較例１のナイーブなＳＶＤと本発明の方法を比較すると、遙かに計算時間が早くなっていることが示される。
【０１００】
表４に示すメモリ使用量に関してみれば、比較例２も実施例２もほぼ同程度にナイーブなＳＶＤにより得られた比較例１において使用されるメモリよりも著しく小さなメモリですむことが示されている。したがって、本発明の方法は、メモリ資源の消費といった点でも従来の方法と劣るものではないことが示される。
【０１０１】
（実施例３）
＜アップデート例＞
実施例３として、本発明において共分散行列を使用してドキュメントのアップデートを行う場合について、検討を加えた。実施例３においては、図１６に示すような７個のデータ・ファイル（その内容は、キーワードとその出現頻度）が、時間順に与えられたものとした。図１６に示されるように、データ・ファイルは、タイムス・タンプとデータ・ファイル名とを記したデータ例を示す。この実施例では、“ｄａｔａｓｅｔ”という名前のファイルとして参照する。例えば、図１６中、２００１０７０１は、２００１年７月１日を意味するタイム・スタンプである。
【０１０２】
図１６に現れる個々のファイルの内容を、図１７から図２３に示す。個々のファイルの内容は、アトリビュートとして使用されるキーワードと、その重み付け（重み付けは、正の実数値で、大きいほどそのキーワードがその文書に含まれる寄与率が大きい）とからなるペアとして構成している。また、図１７から図２３までの文書データ・ファイルは、説明する実施例では２０の文書数として構成した。
【０１０３】
以上のデータが与えられた場合、まず２００１年７月１日のタイム・スタンプを含むｄａｔａ１から順に読み込んでいき、図２３のｄａｔａ７まで本発明の共分散行列法を順次適用する。最後に、ｄａｔａ７まで処理したところで共分散行列を使用してＳＶＤを実行して、特異値および特異ベクトルを得た。比較のため、はじめから表５～表１０に示す１４

０文書のデータ・ファイルを与え、これから共分散行列を生成してＳＶＤを実行して得られた場合と特異値および特異ベクトルの結果を比較した（ベクトルの場合、順序を除いた。）。

【０１０４】

なお、表５の第１行目の１４０　４０は、文書数のトータルが１４０個で、４０個のキーワードからなるものであることを示している。図２４には、本発明の実施例３において使用したキーワードを示す。

【０１０５】

【表５】

1 行目 :140 40

2 行目 :1 1.0 2 1.0 4 0.4

3 行目 :1 1.0 2 1.0 5 0.3

4 行目 :1 1.0 2 1.0 6 0.1

5 行目 :1 1.0 2 1.0 4 0.3

6 行目 :1 1.0 2 1.0 6 0.2

7 行目 :1 1.0 2 1.0 4 0.2

8 行目 :1 1.0 2 1.0 6 0.3

9 行目 :1 1.0 2 1.0 4 0.1

10 行目 :1 1.0 2 1.0 6 0.4

11 行目 :1 1.0 2 1.0

12 行目 :1 1.0 2 1.0 3 0.5 5 0.3

13 行目 :1 1.0 2 1.0 3 0.5 4 0.4

14 行目 :1 1.0 2 1.0 3 0.5 6 0.1

15 行目 :1 1.0 2 1.0 3 0.5

16 行目 :1 1.0 2 1.0 3 0.5 5 0.3

17 行目 :1 1.0 2 0.5 3 1.0 4 0.1

18 行目 :1 1.0 2 0.5 3 1.0 4 0.4

19 行目 :1 1.0 2 0.5 3 1.0 6 0.1

20 行目 :1 1.0 2 0.5 3 1.0

21 行目 :1 1.0 2 0.5 3 1.0 5 0.3 6 0.2

22 行目 :1 1.0 3 1.0 6 0.1

23 行目 :1 1.0 3 1.0 4 0.2

24 行目 :1 1.0 3 1.0

25 行目 :1 1.0 3 1.0 5 0.3

26 行目 :1 1.0 3 1.0 4 0.4

27 行目 :7 1.0 9 1.0 10 0.1 11 0.4

【０１０６】

【表６】
（続き）

28 行目 : 7 1.0 9 1.0 11 0.3

29 行目 : 7 1.0 9 1.0 10 0.2

30 行目 : 7 1.0 9 1.0

31 行目 : 7 1.0 9 1.0 10 0.1

32 行目 : 7 1.0 9 1.0 11 0.2

33 行目 : 7 1.0 9 1.0 10 0.3

34 行目 : 7 1.0 9 1.0 11 0.4

35 行目 : 7 1.0 9 1.0 11 0.1

36 行目 : 7 1.0 9 1.0 10 0.1

37 行目 : 7 1.0 8 0.5 9 1.0 11 0.2

38 行目 : 7 1.0 8 0.5 9 1.0 10 0.3

39 行目 : 7 1.0 8 0.5 9 1.0 10 0.2

40 行目 : 7 1.0 8 0.5 9 1.0 11 0.1

41 行目 : 7 1.0 8 0.5 9 1.0

42 行目 : 7 1.0 8 1.0 9 0.5 10 0.1

43 行目 : 7 1.0 8 1.0 9 0.5 11 0.3

44 行目 : 7 1.0 8 1.0 9 0.5 11 0.2

45 行目 : 7 1.0 8 1.0 9 0.5 11 0.1

46 行目 : 7 1.0 8 1.0 9 0.5

47 行目 : 7 1.0 8 1.0 10 0.1

48 行目 : 7 1.0 8 1.0

49 行目 : 7 1.0 8 1.0 11 0.3

50 行目 : 7 1.0 8 1.0 10 0.2

51 行目 : 7 1.0 8 1.0 11 0.1

52 行目 : 12 1.0 18 0.1

53 行目 : 12 1.0 21 0.1

【０１０７】
【表７】

（続き）

54 行目 : 12 1.0 24 0.1

55 行目 : 12 1.0 35 0.1

56 行目 : 12 1.0

57 行目 : 13 1.0 19 0.1

58 行目 : 13 1.0 22 0.1

59 行目 : 13 1.0 28 0.1

60 行目 : 13 1.0 33 0.1

61 行目 : 13 1.0

62 行目 : 14 1.0 16 0.1

63 行目 : 14 1.0 23 0.1

64 行目 : 14 1.0 29 0.1

65 行目 : 14 1.0 37 0.1

66 行目 : 14 1.0

67 行目 : 15 1.0 17 0.1

68 行目 : 15 1.0 25 0.1

69 行目 : 15 1.0 30 0.1

70 行目 : 15 1.0 38 0.1

71 行目 : 15 1.0

72 行目 : 21 0.3 29 0.4 33 0.2 37 1.0

73 行目 : 19 0.2 28 0.3 29 0.2 33 0.3 36 0.2 38 1.0

74 行目 : 20 0.2 32 0.2 35 0.2 39 1.0

75 行目 : 16 0.3 22 0.4 23 0.2 26 0.1 28 0.3 40 1.0

76 行目 : 16 1.0 18 0.2 40 0.3

77 行目 : 17 1.0 26 0.2 27 0.4 40 0.1

78 行目 : 16 0.2 18 1.0 22 0.3 23 0.4 24 0.2 27 0.2 29 0.4 34 0.3 36 0.2

79 行目 : 19 1.0 22 0.1 25 0.3 29 0.4 33 0.3 34 0.4 35 0.4

【０１０８】

【表８】

（続き）

80 行目 : 20 1.0 22 0.1 24 0.3 25 0.3

81 行目 : 21 1.0 29 0.3 33 0.2 35 0.2 36 0.2

82 行目 : 22 1.0 26 0.4 34 0.4 35 0.2

83 行目 : 23 1.0 30 0.3 37 0.3

84 行目 : 20 0.3 24 1.0

85 行目 : 22 0.3 25 1.0 30 0.2 38 0.4

86 行目 : 22 0.1 26 1.0 29 0.2 31 0.2 34 0.3 36 0.3

87 行目 : 18 0.2 25 0.3 26 0.1 27 1.0 34 0.3 35 0.3

88 行目 : 27 0.2 28 1.0

89 行目 : 17 0.3 23 0.4 29 1.0 37 0.4 40 0.4

90 行目 : 18 0.4 24 0.3 27 0.3 29 0.3 30 1.0 37 0.2

91 行目 : 16 0.3 17 0.2 25 0.3 27 0.3 29 0.2 31 1.0 33 0.3 35 0.3

92 行目 : 20 0.2 25 0.1 30 0.3 32 1.0 36 0.4

93 行目 : 19 0.2 20 0.3 26 0.4 28 0.2 33 1.0 35 0.4

94 行目 : 17 0.3 34 1.0

95 行目 : 21 0.4 30 0.3 34 0.3 35 1.0 38 0.4

96 行目 : 17 0.4 20 0.4 21 0.3 23 0.4 27 0.4 36 1.0

97 行目 : 18 0.3 22 0.2 29 0.2 37 1.0 39 0.2

98 行目 : 18 0.2 23 0.3 24 0.3 29 0.2 32 0.3 36 0.2 38 1.0

99 行目 : 18 0.2 28 0.3 39 1.0

100 行目 : 40 1.0

101 行目 : 16 1.0 18 0.3 20 0.2 26 0.3 30 0.3 37 0.3 39 0.3

102 行目 : 17 1.0 20 0.4 24 0.2 26 0.2 27 0.2 30 0.4 36 0.4 38 0.4

103 行目 : 18 1.0 19 0.3 30 0.2

104 行目 : 19 1.0 25 0.2 31 0.3 33 0.3 36 0.4

105 行目 : 18 0.2 20 1.0 22 0.4 23 0.3 33 0.3 37 0.4 38 0.4 40 0.2

【０１０９】

【表９】

（続き）

106 行目 : 19 0.2 21 1.0 22 0.2 25 0.1 37 0.2

107 行目 : 22 1.0 28 0.2 32 0.4 37 0.4

108 行目 : 21 0.3 23 1.0 34 0.3 40 0.2

109 行目 : 24 1.0

110 行目 : 19 0.3 24 0.1 25 1.0 27 0.4 30 0.3

111 行目 : 20 0.4 26 1.0 28 0.4 30 0.4 35 0.2

112 行目 : 24 0.3 25 0.1 27 1.0 29 0.3

113 行目 : 26 0.3 28 1.0 30 0.3 35 0.4 40 0.4

114 行目 : 16 0.4 20 0.4 24 0.3 29 1.0 32 0.2 33 0.1 35 0.4

115 行目 : 23 0.4 30 1.0 35 0.3

116 行目 : 31 1.0 34 0.4

117 行目 : 26 0.3 32 1.0 39 0.4

118 行目 : 33 1.0 35 0.2 36 0.2 37 0.2 38 0.2

119 行目 : 17 0.2 22 0.2 27 0.2 31 0.2 34 1.0

120 行目 : 18 0.2 35 1.0

121 行目 : 28 0.1 33 0.3 35 0.2 36 1.0 38 0.3

122 行目 : 37 1.0

123 行目 : 18 0.4 19 0.2 20 0.3 22 0.3 23 0.3 38 1.0 39 0.3

124 行目 : 20 0.3 29 0.2 34 0.2 39 1.0

125 行目 : 16 0.3 17 0.3 23 0.3 25 0.4 33 0.4 34 0.3 36 0.2 37 0.4 39 0.2 40 1.0

126 行目 : 16 1.0 17 0.4 19 0.2 24 0.3 27 0.2 31 0.1 32 0.1 36 0.3 39 0.1

127 行目 : 17 1.0 23 0.2 26 0.4 28 0.2 39 0.3

128 行目 : 18 1.0 27 0.2 33 0.3 35 0.2

129 行目 : 19 1.0 24 0.2 26 0.2 30 0.1 33 0.2

130 行目 : 20 1.0 21 0.2 31 0.2 33 0.2

131 行目 : 21 1.0 36 0.2 37 0.3

【０１１０】

【表１０】

（続き）

132 行目:16 0.4 22 1.0 30 0.3 32 0.3 35 0.2 37 0.4

133 行目:23 1.0 28 0.3 30 0.3 33 0.3 40 0.2

134 行目:24 1.0 33 0.4 35 0.2 38 0.4

135 行目:23 0.3 25 1.0 35 0.2 38 0.4 39 0.2 40 0.4

136 行目:17 0.2 18 0.3 22 0.3 24 0.2 26 1.0 38 0.4

137 行目:16 0.3 18 0.1 21 0.2 25 0.3 27 1.0 36 0.2

138 行目:23 0.2 26 0.2 28 1.0

139 行目:29 1.0 31 0.4 38 0.3 40 0.4

140 行目:18 0.1 26 0.2 28 0.3 29 0.3 30 1.0 33 0.2 35 0.1

141 行目:24 0.1 26 0.2 31 1.0

【０１１１】

実施例３では、まず図１７に示したｄａｔａ１が入力されるものとした。図２５には、ｄａｔａ１だけを処理した時点でのＳＵＭ－ＭＥＡＮ（Ｍ）$_1$と、ＳＵＭ（Ｍ）$_1$とを示す。なお、最初の６行は、新しいキーワードが見つかった文書番号とそのキーワード名とを示す。図２５に示されるように、ｄａｔａ１を処理した時点では、キーワード総数は６で、したがってＳＵＭ－ＭＥＡＮ（Ｍ）$_1$は、６次元ベクトルであり、ＳＵＭ（Ｍ）$_1$は、６×６次元の対称行列となる。

【０１１２】

図２５においては、対称性から要素の半分だけを書き出して示している。また、キーワードは見つかるたびに、キーワードを管理するキーワード・ハッシュ・テーブルに追加される。キーワード・ハッシュ・テーブルにはこの他に、何個の文書がそのキーワードを含んでいるかのカウント数も保持されている。実施例３においては、キーワード・ハッシュ・テーブル以外に、総文書数Ｍ＝２０と、総キーワード数Ｎ＝６とを保持させた。その後、ｄａｔａ２をアップデートすることにより、ドキュメントの追加を実行した。その結果を、図２６に示す。なお、図２６においてもデータは、図２５と同様の順および構成として示されている。ｄａｔａ３からｄａｔａ６までを同様に処理した。最後にｄａｔａ７をアップデート処理した。この処理後のＳＵＭ－ＭＥＡＮ（Ｍ）$_1$を図２７に示す。また、ＳＵＭ（Ｍ）１の内容を表１１から表１２に示す。なお、ｄａｔａ３からｄａｔａ６までのアップデート処理で、キーワードが合計４０個出揃っているため、ｄａｔａ７の処理においては、新たなキーワードの追加はされなかった。なお、表１１および表１２中「*」で示された要素は、０．０を意味する。また、コロンの左側の数字は、行列の行番号を示す。

【０１１３】

【表１１】

1: 25 17.5 2.5 1.5 1.5 12.5 *

2: 16.25 1.65 1.05 1.25 5.0 *

3: 0.83 * * 1.30 *

4: 0.45 0.06 0.90 *

5: 0.37 0.45 *

6: 11.25 *

7: 25.00 17.50 1.70 2.70 12.50 *

8: 16.25 1.35 2.00 5.00 *

9: 0.35 0.04 0.65 *

10: 0.75 1.15 *

11: 11.25 *

12: 5.00 0.10 0.10 0.10 0.10 *

13: 3.86 0.02 0.44 0.47 * 0.38 0.65 0.09 0.38 * 0.73 0.64 0.65 0.55 * 0.06 0.09 0.79 0.80 0.26 0.38 0.06 0.47 0.43 0.10 0.82 0.36 *

14: 3.52 * 0.60 * 0.20 0.20 * 0.30 * 0.06 0.42 0.42 0.80 * 0.12 0.16 0.12 0.16 0.74 0.32 * * * 0.06 0.32 0.21 0.04

15: 3.73 0.32 * 0.29 0.15 * 0.47 * 0.46 0.17 0.62 0.06 * 0.36 0.22 0.43 0.86 0.27 0.80 0.18 0.03 0.30 * 0.57 0.06 0.13

16: 3.33 * 0.48 0.44 0.61 1.11 * 0.33 0.18 0.71 0.12 * 0.06 0.50 0.96 0.66 0.28 0.40 0.18 0.24 0.61 0.24 0.43 0.63 0.30

17: 5.00 0.10 0.10 0.10 0.10 * * * * * * * * * * * * * * * * * * *

18: 3.39 0.2 0.10 1.06 * 0.2 0.06 0.44 0.04 * 0.08 0.82 0.25 0.4 0.5 0.12 0.02 0.08 0.28 * 0.16 0.4 0.32

19: 3.84 0.32 0.15 * 0.58 0.41 0.22 1.2 * 0.1 0.38 0.36 0.7 0.09 0.59 0.7 0.13 0.84 0.48 0.1 0.76 0.06

20: 3.75 0.47 * 0.09 0.60 0.15 0.08 * 0.20 * 0.85 0.33 0.16 0.22 0.08 0.36 1.15 0.76 0.20 * *

21: 3.26 * 0.25 0.51 0.54 0.68 * 0.18 0.40 0.31 0.87 0.80 0.84 0.02 0.08 0.48 0.52 0.15 0.24

【０１１４】
【表１２】

0.43

22: 5.00 0.10 0.10 0.10 0.10 * * * * * * * * * * * * * *

23: 3.73 0.23 0.54 0.58 * 0.55 0.30 0.42 * 0.46 0.36 0.30 0.46 0.33 0.90 0.63 0.15 0.40

24: 4.22 0.62 0.70 * 0.57 0.42 1.00 0.84 0.60 0.55 0.09 0.27 0.14 1.24 0.24 0.51 *

25: 4.09 1.06 * 0.34 0.21 0.60 0.70 0.28 0.46 0.26 0.24 0.26 0.80 0.53 0.38 0.64

26: 4.20 * 0.24 0.18 0.50 0.20 0.18 0.46 0.28 0.37 0.09 0.64 0.06 0.12 *

27: 5.00 0.10 0.10 0.10 0.10 * * * * * * * * *

28: 3.72 0.18 0.40 0.48 0.98 0.56 0.04 0.40 1.00 0.52 0.94 0.59 0.28

29: 3.69 0.53 0.80 0.26 0.32 0.10 0.28 0.03 0.80 1.19 0.33 0.36

30: 4.14 0.36 0.28 0.44 0.39 0.09 0.88 0.18 0.50 0.09 *

31: 4.35 0.90 0.86 0.30 0.38 0.48 0.36 0.08 0.12 0.12

32: 2.98 0.64 0.49 0.07 0.38 0.20 0.78 0.21 0.21

33: 4.12 0.32 0.65 0.66 0.20 0.24 0.06 0.20

34: 2.43 0.61 0.30 * 0.02 * 0.01

35: 3.56 0.33 0.28 0.02 0.26 0.01

36: 4.01 0.24 0.22 0.49 0.40

37: 3.86 0.04 0.36 0.16

38: 3.90 0.56 0.36

39: 3.06 0.66

40: 3.38

【０１１５】

本発明においては、上述したように共分散行列を使用してｄａｔａ１からｄａｔａ７までのデータのアップデートを終了する。また、本発明において、この時点で共分散行列を作って欲しいとのユーザ・リクエストがある場合には、この時点でＳＵＭ（Ｍ）$_2$を下記式で計算する。これは４０×４０の対称行列となる。

【０１１６】

【数１５】

$$\mathrm{SUM}(M)_2 = {}^{\mathrm{bar}}\mathbf{d}(M)\ {}^{\mathrm{bar}}\mathbf{d}(M)^{\mathbf{T}}$$

その後、ＳＵＭ（Ｍ）$_1$と、ＳＵＭ（Ｍ）$_2$とから共分散行列を計算させる。得られた共分散行列の結果を表１３から表１９に示す。得られる共分散行列は対称行列なので、表１３～表１９では、要素の半分だけを示している。

【０１１７】

【表１３】

共分散行列　C

〔1行目〕

1.5e-01 1.0e-01 1.5e-02 8.8e-03 8.8e-03 7.3e-02 -3.2e-02 -2.2e-02 -2.2e-03 -3.4e-03 -1.6e-02 -6.4e-03 -7.9e-03 -6.1e-03 -7.5e-03 -8.5e-03 -6.4e-03 -6.0e-03 -7.7e-03 -7.3e-03 -8.2e-03 -6.4e-03 -6.8e-03 -8.7e-03 -8.5e-03 -8.4e-03 -6.4e-03 -6.9e-03 -7.0e-03 -8.7e-03 -8.5e-03 -6.9e-03 -8.2e-03 -4.5e-03 -6.4e-03 -8.3e-03 -7.1e-03 -7.7e-03 -6.6e-03 -5.6e-03

〔2行目〕

1.0e-01 9.6e-03 6.2e-03 7.6e-03 2.5e-02 -2.2e-02 -1.6e-02 -1.5e-03 -2.4e-03 -1.1e-02 -4.5e-03 -5.5e-03 -4.3e-03 -5.3e-03 -6.0e-03 -4.5e-03 -4.2e-03 -5.4e-03 -5.1e-03 -5.7e-03 -4.5e-03 -4.7e-03 -6.1e-03 -6.0e-03 -5.9e-03 -4.5e-03 -4.8e-03 -4.9e-03 -6.1e-03 -6.0e-03 -4.8e-03 -5.7e-03 -3.1e-03 -4.5e-03 -5.8e-03 -5.0e-03 -5.4e-03 -4.6e-03 -3.9e-03

〔3行目〕

5.6e-03 -1.9e-04 -1.9e-04 7.7e-03 -3.2e-03 -2.2e-03 -2.2e-04 -3.4e-04 -1.6e-03 -6.4e-04 -7.9e-04 -6.1e-04 -7.5e-04 -8.5e-04 -6.4e-04 -6.0e-04 -7.7e-04 -7.3e-04 -8.2e-04 -6.4e-04 -6.8e-04 -8.7e-04 -8.5e-04 -8.4e-04 -6.4e-04 -6.9e-04 -7.0e-04 -8.7e-04 -8.5e-04 -6.9e-04 -8.2e-04 -4.5e-04 -6.4e-04 -8.3e-04 -7.1e-04 -7.7e-04 -6.6e-04 -5.6e-04

〔4行目〕

3.1e-03 3.1e-04 5.5e-03 -1.9e-03 -1.3e-03 -1.3e-04 -2.1e-04 -9.6e-04 -3.8e-04 -4.7e-04 -3.7e-04 -4.5e-04 -5.1e-04 -3.8e-04 -3.6e-04 -4.6e-04 -4.4e-04 -4.9e-04 -3.8e-04 -4.1e-04 -5.2e-04 -5.1e-04 -5.1e-04 -3.8e-04 -4.1e-04 -4.2e-04 -5.2e-04 -5.1e-04 -4.1e-04 -4.9e-04 -2.7e-04 -3.8e-04 -5.0e-04 -4.3e-04 -4.6e-04 -4.0e-04 -3.4e-04

【０１１８】

【表１４】

共分散行列　C（続き）

〔5行目〕

2.5e-03 2.3e-03 -1.9e-03 -1.3e-03 -1.3e-04 -2.1e-04 -9.6e-04 -3.8e-04 -4.7e-04
-3.7e-04 -4.5e-04 -5.1e-04 -3.8e-04 -3.6e-04 -4.6e-04 -4.4e-04 -4.9e-04 -3.8e-04
-4.1e-04 -5.2e-04 -5.1e-04 -5.1e-04 -3.8e-04 -4.1e-04 -4.2e-04 -5.2e-04 -5.1e-04
-4.1e-04 -4.9e-04 -2.7e-04 -3.8e-04 -5.0e-04 -4.3e-04 -4.6e-04 -4.0e-04 -3.4e-04

〔6行目〕

7.2e-02 -1.6e-02 -1.1e-02 -1.1e-03 -1.7e-03 -8.0e-03 -3.2e-03 -4.0e-03 -3.1e-03
-3.8e-03 -4.3e-03 -3.2e-03 -3.0e-03 -3.8e-03 -3.6e-03 -4.1e-03 -3.2e-03 -3.4e-03
-4.3e-03 -4.3e-03 -4.2e-03 -3.2e-03 -3.4e-03 -3.5e-03 -4.3e-03 -4.3e-03 -3.4e-03
-4.1e-03 -2.2e-03 -3.2e-03 -4.1e-03 -3.6e-03 -3.8e-03 -3.3e-03 -2.8e-03

〔7行目〕

1.5e-01 1.0e-01 1. e-02 1.6e-02 7.3e-02 -6.4e-03 -7.9e-03 -6.1e-03 -7.5e-03 -
8.5e-03 -6.4e-03 -6.0e-03 -7.7e-03 -7.3e-03 -8.2e-03 -6.4e-03 -6.8e-03 -8.7e-03
-8.5e-03 -8.4e-03 -6.4e-03 -6.9e-03 -7.0e-03 -8.7e-03 -8.5e-03 -6.9e-03 -8.2e-03
-4.5e-03 -6.4e-03 -8.3e-03 -7.1e-03 -7.7e-03 -6.6e-03 -5.6e-03

〔8行目〕

1.0e-01 8.1e-03 1.2e-02 2.5e-02 -4.5e-03 -5.5e-03 -4.3e-03 -5.3e-03 -6.0e-03 -
4.5e-03 -4.2e-03 -5.4e-03 -5.1e-03 -5.7e-03 -4.5e-03 -4.7e-03 -6.1e-03 -6.0e-03
-5.9e-03 -4.5e-03 -4.8e-03 -4.9e-03 -6.1e-03 -6.0e-03 -4.8e-03 -5.7e-03 -3.1e-03
-4.5e-03 -5.8e-03 -5.0e-03 -5.4e-03 -4.6e-03 -3.9e-03

〔9行目〕

2.4e-03 5.2e-05 3.6e-03 -4.3e-04 -5.4e-04 -4.2e-04 -5.1e-04 -5.8e-04 -4.3e-04
-4.1e-04 -5.2e-04 -4.9e-04 -5.6e-04 -4.3e-04 -4.6e-04 -5.9e-04 -5.8e-04 -5.7e-04
-4.3e-04 -4.7e-04 -4.8e-04 -5.9e-04 -5.8e-04 -4.7e-04 -5.6e-04 -3.0e-04 -4.3e-04
-5.6e-04 -4.9e-04 -5.2e-04 -4.5e-04 -3.8e-04

【０１１９】
【表１５】

共分散行列　C（続き）

〔10行目〕

5.0e-03 6.5e-03 -6.9e-04 -8.5e-04 -6.6e-04 -8.1e-04 -9.2e-04 -6.9e-04 -6.5e-04

-8.3e-04 -7.9e-04 -8.8e-04 -6.9e-04 -7.3e-04 -9.4e-04 -9.2e-04 -9.1e-04 -6.9e-04

-7.4e-04 -7.6e-04 -9.4e-04 -9.2e-04 -7.4e-04 -8.8e-04 -4.8e-04 -6.9e-04 -9.0e-04

-7.7e-04 -8.3e-04 -7.2e-04 -6.1e-04

〔11行目〕

7.2e-02 -3.2e-03 -4.0e-03 -3.1e-03 -3.8e-03 -4.3e-03 -3.2e-03 -3.0e-03 -3.8e-03

-3.6e-03 -4.1e-03 -3.2e-03 -3.4e-03 -4.3e-03 -4.3e-03 -4.2e-03 -3.2e-03 -3.4e-03

-3.5e-03 -4.3e-03 -4.3e-03 -3.4e-03 -4.1e-03 -2.2e-03 -3.2e-03 -4.1e-03 -3.6e-03

-3.8e-03 -3.3e-03 -2.8e-03

〔12行目〕

3.4e-02 -8.7e-04 -5.1e-04 -7.9e-04 -9.9e-04 -1.3e-03 -1.2e-03 -1.5e-03 -1.5e-03

-1.6e-03 -1.3e-03 -1.4e-03 -1.7e-03 -1.7e-03 -1.7e-03 -1.3e-03 -1.4e-03 -1.4e-03

-1.7e-03 -1.7e-03 -1.4e-03 -1.6e-03 -8.9e-04 -1.3e-03 -1.7e-03 -1.4e-03 -1.5e-03

-1.3e-03 -1.1e-03

〔13行目〕

2.6e-02 -1.4e-03 1.3e-03 1.2e-03 -1.6e-03 1.2e-03 2.7e-03 -1.2e-03 6.9e-04 -

1.6e-03 3.5e-03 2.4e-03 2.5e-03 1.8e-03 -1.6e-03 -1.3e-03 -1.1e-03 3.5e-03

3.6e-03 1.5e-04 6.9e-04 -6.8e-04 1.8e-03 1.0e-03 -1.1e-03 4.0e-03 9.3e-04 -

1.4e-03

〔14行目〕

2.4e-02 -1.4e-03 2.6e-03 -1.2e-03 2.8e-04 -4.1e-05 -1.4e-03 5.8e-04 -1.2e-03 -

8.7e-04 1.3e-03 1.4e-03 4.1e-03 -1.2e-03 -4.7e-04 -2.0e-04 -8.1e-04 -5.0e-04

4.0e-03 7.2e-04 -8.6e-04 -1.2e-03 -1.6e-03 -9.4e-04 8.2e-04 2.3e-04 -7.9e-04

【０１２０】

【表１６】

共分散行列　C（続き）

〔15行目〕

2.5e-02　2.7e-04　-1.5e-03　6.6e-04　-7.3e-04　-1.7e-03　1.4e-03　-1.5e-03　1.7e-03

-8.3e-04　2.4e-03　-1.6e-03　-1.5e-03　9.5e-04　-8.4e-05　1.0e-03　4.1e-03　3.0e-04

3.8e-03　2.3e-04　-1.3e-03　1.9e-04　-1.7e-03　2.3e-03　-1.1e-03　-4.0e-04

〔16行目〕

2.1e-02　-1.7e-03　1.8e-03　1.1e-03　2.4e-03　5.7e-03　-1.7e-03　5.5e-04　-1.0e-03

2.8e-03　-1.4e-03　-1.7e-03　-1.4e-03　1.7e-03　4.5e-03　2.4e-03　1.5e-04　6.7e-04

8.9e-05　5.1e-06　2.1e-03　-2.0e-04　1.0e-03　2.7e-03　6.4e-04

〔17行目〕

3.4e-02　-4.8e-04　-8.2e-04　-7.4e-04　-9.2e-04　-1.3e-03　-1.4e-03　-1.7e-03　-1.7e-03

-1.7e-03　-1.3e-03　-1.4e-03　-1.4e-03　-1.7e-03　-1.7e-03　-1.4e-03　-1.6e-03　-8.9e-04

-1.3e-03　-1.7e-03　-1.4e-03　-1.5e-03　-1.3e-03　-1.1e-03

〔18行目〕

2.3e-02　-1.0e-05　-6.5e-04　6.0e-03　-1.2e-03　1.6e-04　-1.2e-03　1.5e-03　-1.3e-03　-

1.2e-03　-7.2e-04　4.5e-03　1.6e-04　1.3e-03　2.3e-03　-6.8e-04　-7.0e-04　-6.3e-04

4.4e-04　-1.3e-03　-3.0e-04　1.6e-03　1.2e-03

〔20行目〕

2.6e-02　5.4e-04　-8.9e-04　-1.5e-03　2.5e-03　8.5e-04　-4.8e-04　6.6e-03　-1.5e-03　-

9.4e-04　1.0e-03　4.9e-04　2.9e-03　-1.0e-03　2.3e-03　3.9e-03　-6.0e-04　4.0e-03

1.7e-03　-1.1e-03　3.8e-03　-9.2e-04

〔21行目〕

2.5e-02　1.5e-03　-1.5e-03　-9.0e-04　2.3e-03　-8.8e-04　-1.3e-03　-1.5e-03　-1.4e-04

-1.6e-03　4.1e-03　4.1e-04　-4.3e-04　-2.9e-04　-4.5e-04　1.1e-03　6.3e-03　3.8e-03　-

3.2e-04　-1.5e-03　-1.3e-03

【０１２１】

【表１７】

共分散行列　C（続き）

〔22行目〕

2.1e-02 -1.6e-03 5.5e-05 1.4e-03 1.7e-03 2.7e-03 -1.6e-03 -4.8e-04 1.1e-03 -
6.1e-06 4.0e-03 4.0e-03 3.9e-03 -1.0e-03 -1.1e-03 1.3e-03 1.9e-03 -8.9e-04
1.6e-05 1.6e-03

〔23行目〕

3.4e-02 -6.4e-04 -1.0e-03 -9.9e-04 -9.7e-04 -1.3e-03 -1.4e-03 -1.4e-03 -1.7e-03
-1.7e-03 -1.4e-03 -1.6e-03 -8.9e-04 -1.3e-03 -1.7e-03 -1.4e-03 -1.5e-03 -1.3e-03
-1.1e-03

〔24行目〕

2.5e-02 -2.0e-04 2.0e-03 2.4e-03 -1.4e-03 2.5e-03 6.6e-04 1.2e-03 -1.8e-03
1.8e-03 8.4e-04 1.2e-03 1.9e-03 6.0e-04 4.9e-03 2.9e-03 -3.3e-04 1.7e-03

〔25行目〕

2.8e-02 2.1e-03 2.7e-03 -1.7e-03 2.2e-03 1.1e-03 4.8e-03 3.7e-03 2.4e-03
1.7e-03 -5.7e-04 1.9e-04 -1.3e-03 6.9e-03 -3.7e-04 1.8e-03 -1.5e-03

〔26行目〕

2.7e-02 5.3e-03 -1.7e-03 5.8e-04 -3.8e-04 2.0e-03 2.7e-03 1.5e-04 1.1e-03
6.6e-04 5.1e-06 -3.6e-04 3.8e-03 1.7e-03 9.4e-04 3.1e-03

〔27行目〕

2.8e-02 -1.7e-03 -1.0e-04 -5.7e-04 1.3e-03 -8.3e-04 -5.3e-04 1.1e-03 8.2e-04
9.6e-04 -1.5e-03 2.7e-03 -1.6e-03 -8.9e-04 -1.5e-03

〔27行目〕

3.4e-02 -6.6e-04 -6.9e-04 -1.0e-03 -9.9e-04 -1.4e-03 -1.6e-03 -8.9e-04 -1.3e-03
-1.7e-03 -1.4e-03 -1.5e-03 -1.3e-03 -1.1e-03

〔28行目〕

2.5e-02 -2.3e-04 9.8e-04 1.6e-03 5.5e-03 2.2e-03 -6.8e-04 1.5e-03 5.4e-03
2.2e-03 5.1e-03 2.8e-03 7.9e-04

【０１２２】
【表１８】

共分散行列　Ｃ（続き）

〔29行目〕

2.5e-02　1.9e-03　3.8e-03　3.4e-04　4.9e-04　-2.7e-04　6.0e-04　-1.6e-03　4.1e-03

6.8e-03　9.0e-04　1.3e-03

〔30行目〕

2.7e-02　2.5e-04　1.3e-04　9.2e-04　1.6e-03　-1.1e-03　4.0e-03　-6.6e-04　1.5e-03　-

1.2e-03　-1.5e-03

〔31行目〕

2.9e-02　4.6e-03　4.0e-03　9.5e-04　1.0e-03　1.2e-03　6.6e-04　-1.5e-03　-9.2e-04　-

6.5e-04

〔32行目〕

2.0e-02　2.8e-03　2.5e-03　-8.8e-04　9.2e-04　-1.1e-04　3.9e-03　6.7e-05　2.9e-04

〔33行目〕

2.7e-02　1.1e-03　3.0e-03　2.6e-03　-4.0e-04　-2.4e-04　-1.3e-03　-8.2e-06

〔34行目〕

1.7e-02　3.5e-03　9.8e-04　-1.0e-03　-9.3e-04　-9.3e-04　-7.1e-04

〔35行目〕

2.4e-02　7.0e-04　5.7e-04　-1.4e-03　5.3e-04　-1.1e-03

【０１２３】

【表１９】

（続き）

〔36行目〕

2.6e-02　-1.4e-04　-4.2e-04　1.8e-03　1.4e-03

〔37行目〕

2.6e-02　-1.4e-03　1.1e-03　-1.1e-04

〔38行目〕

2.6e-02　2.4e-03　1.2e-03

〔39行目〕

2.0e-02　3.5e-03

〔40行目〕

2.3e-02

【０１２４】

上記の逐次アップデートで計算された共分散行列Ｃ’を用いて特異値分解し、最初の１０個の特異値と最大の特異値に対する特異ベクトルを計算した結果を、図２８に示す。また、特異ベクトル(最大の特異値に対するもの)を表２０に示す。

【０１２５】
【表２０】
〔第１特異値に対する特異ベクトル〕（Ｎ＝４０次元）

-5.2960e-01 -3.8319e-01 -5.2864e-02 -3.2729e-02 -3.1732e-02

-2.6187e-01 5.2893e-01 3.8302e-01 3.6178e-02 5.7374e-02

2.6109e-01 3.1860e-05 4.9484e-05 3.5469e-05 4.4623e-05

5.2818e-05 3.1851e-05 3.6387e-05 4.8000e-05 4.3140e-05

5.1465e-05 3.1915e-05 4.2917e-05 5.5013e-05 5.4694e-05

5.1524e-05 3.1898e-05 4.3605e-05 4.3651e-05 5.4658e-05

5.5368e-05 4.3439e-05 5.1774e-05 2.6056e-05 3.7219e-05

5.1642e-05 4.4973e-05 4.7018e-05 3.9580e-05 3.2341e-05

【０１２６】
比較例３として、従来の手法（あらかじめ文書総数、キーワード総数がわかっている場合）で表５～表１０のようなデータが与えられた場合に、直接共分散行列を計算し、これを特異値分解して特異値（大きい方から１０個）と最大特異値に対する特異ベクトルを求めた結果を図２９に示す。
【０１２７】
また、第１特異値に対する特異ベクトル（Ｎ＝４０次元）を下記表２１に示す
【０１２８】
【表２１】
〔第１特異値に対する特異ベクトル〕(Ｎ＝４０次元)

-5.2960e-01 -3.8319e-01 -2.6187e-01 -5.2864e-02 -3.2729e-02

-3.1732e-02 5.2893e-01 2.6109e-01 3.8302e-01 3.6178e-02

5.7374e-02 3.1860e-05 3.1851e-05 3.1915e-05 3.1898e-05

4.2917e-05 4.3605e-05 4.9484e-05 3.6387e-05 5.1774e-05

3.5469e-05 4.8000e-05 5.5013e-05 4.4623e-05 4.3651e-05

5.1642e-05 4.7018e-05 4.3140e-05 5.4694e-05 5.4658e-05

3.2341e-05 2.6056e-05 5.1465e-05 3.9580e-05 5.2818e-05

4.3439e-05 5.1524e-05 5.5368e-05 3.7219e-05 4.4973e-05

【０１２９】
上述した実施例３と、比較例３とを比較した結果、本発明の方法で得られる特異値と、全体のデータから共分散行列を作って、これを特異値分解して得られる特異値は、一致していることが示された。
【０１３０】
また、最大特異値についてみても、たとえば、従来手法で得られる最大特異値に対する特異ベクトルの第3要素は-2.6187e-01であり、これは本発明で得られる最大特異値に対する特異ベクトルの第6要素に対応していることが示された。すなわち、最大特異値に対する特異ベクトルは、順列(permutation)を除いて、一致していることが示された。
【０１３１】
（実施例４）

＜ダウンデート例＞
実施例４として、実施例１で使用したデータから日付が２００１年７月１日（ｄａｔａ１）と２日（ｄａｔａ２）のデータを削除することにより、データベースからのドキュメントのダウンデートについて検討した。ダウンデートのため、まず最初の処理は、ＳＵＭ－ＭＥＡＮ（Ｍ）$_1$と、ＳＵＭ（Ｍ）$_1$とを更新した。このために、ｄａｔａ１とｄａｔａ２とをまず走査して、削除分のＳＵＭ－ＭＥＡＮ（Ｄ）$_1$と、ＳＵＭ（Ｄ）$_1$とをまず作成した。ここでＤは、削除する文書数である。ＳＵＭ－ＭＥＡＮ（Ｄ）$_1$と、ＳＵＭ（Ｄ）$_1$とは、それぞれ、削除する文書の平均ベクトル（Ｎ次元）と削除される文書だけからなる積和行列成分(Ｎ×Ｎの対称行列)となる。

【０１３２】
具体的には、ＳＵＭ－ＭＥＡＮ（Ｄ）$_1$と、ＳＵＭ（Ｄ）$_1$とを求めた後、新らたなＳＵＭ－ＭＥＡＮ（Ｍ）$_1$と、ＳＵＭ（Ｍ）$_1$とを下記式を使用して算出した。

【０１３３】
【数１６】

$$\text{SUM-MEAN}(M)_1 = \text{SUM-MEAN}(M)_1 - \text{SUM-MEAN}(D)_1$$

$$\text{SUM}(M)_1 = \text{SUM}(M)_1 - \text{SUM}(D)_1$$

【０１３４】
上記式は、Ｎ次元ベクトルの引き算であり、上記式は、Ｎ×Ｎの対称行列の引き算（要素ごと）である。これで、ＳＵＭ－ＭＥＡＮ（Ｍ）$_1$と、ＳＵＭ（Ｍ）$_1$との更新の第１ステップを終了する。

【０１３５】
次いで、削除することによってキーワード・ハッシュ・テーブルに保持する必要がなくなるキーワードが出てくる可能性があるので、これをＤ個の文書に関してもう一度走査して調査した。実施例３では、ｄａｔａ１とｄａｔａ２とを削除するので、これに伴い図３０に示すキーワードが削除対象として検出された。

【０１３６】
ＳＵＭ－ＭＥＡＮ（Ｍ）$_1$に関しては、上記のキーワード各々に対応するインデックスの位置の要素を削除した（これを（i_1、i_2、i_3、i_4、i_5、i_6）と表記する。）。この結果、キーワード６個分だけ減少した３４次元のベクトルが得られた。ＳＵＭ（Ｍ）$_1$に関しては、この６つのインデックス各々に対応する行列の縦方向と横方向の要素を削除した。たとえば、”AlGore”に対するインデックスi_1に関しては、ＳＵＭ（Ｍ）$_1$の中で行番号または列番号がi_1のものをすべて削除した。この時点でＮ×Ｎ行列は、（Ｎ－１）×（Ｎ－１）行列として生成され、最終的には、（Ｎ－６）×（Ｎ－６）の行列として、ＳＵＭ（Ｍ）$_1$を生成した。

【０１３７】
上述した削除を、”Japan”に対応するインデックスi_6まで繰り返した。このプロセスを終了した時点で、ＳＵＭ－ＭＥＡＮ（Ｍ）$_1$とＳＵＭ（Ｍ）$_1$との更新を終了した。この結果として、元の４０次元から６次元少ない３４次元ベクトルのＳＵＭ－ＭＥＡＮ（Ｍ）$_1$とＳＵＭ（Ｍ）$_1$とを得た。

【０１３８】
最後にＳＵＭ（Ｍ）$_2$をアップデートの場合と同じ式で計算し、共分散行列を求め、これを特異値分解して特異値（大きい方から１０個）と最大特異値に対する特異ベクトルを求めた。その結果を、図３１に示す。また、表２２には、第１特異値に対する特異ベクトル（Ｎ＝３４次元）を示す

【０１３９】
【表２２】

〔第１特異値に対する特異ベクトル〕（Ｎ＝３４次元）

6.6945e-01 1.7290e-01 2.6265e-02 6.7805e-02 6.6945e-01 -2.1384e-02

-5.6159e-02 -3.5503e-02 -4.6750e-02 -5.8561e-02 -2.1388e-02 -3.9300e-02

-5.4406e-02 -4.5086e-02 -5.8817e-02 -2.1630e-02 -4.9441e-02 -6.3746e-02

-6.3778e-02 -5.6538e-02 -2.1591e-02 -5.0169e-02 -4.8973e-02 -6.2525e-02

-6.6006e-02 -4.9784e-02 -5.9844e-02 -2.6616e-02 -3.7788e-02 -5.7980e-02

-5.1621e-02 -5.1754e-02 -4.1644e-02 -3.2110e-02

【０１４０】

また、比較例４としてｄａｔａ３からｄａｔａ７までをまとめた全体の行列を表５～表１０と同様なフォーマットとして一度に共分散行列を計算してＳＶＤを実行させることにより得た特異値（大きい方から１０個）を図３２に示す。また、最大特異値に対する特異ベクトルを求めた結果を表２３に示す。

【０１４１】

【表２３】

〔第1特異値に対する特異ベクトル〕（Ｎ＝３４次元）

6.6945e-01 6.6945e-01 1.7290e-01 2.6265e-02 6.7805e-02 -2.1384e-02

-2.1388e-02 -2.1630e-02 -2.1591e-02 -4.9441e-02 -5.0169e-02 -5.6159e-02

-3.9300e-02 -5.9844e-02 -3.5503e-02 -5.4406e-02 -6.3746e-02 -4.6750e-02

-4.8973e-02 -5.7980e-02 -5.1754e-02 -4.5086e-02 -6.3778e-02 -6.2525e-02

-3.2110e-02 -2.6616e-02 -5.8817e-02 -4.1644e-02 -5.8561e-02 -4.9784e-02

-5.6538e-02 -6.6006e-02 -3.7788e-02 -5.1621e-02

【０１４２】

実施例４および比較例４の結果を比較すると、最大から１０個の特異値に関しては、従来手法と一致していることが示される。また、本手法で得られる最大特異値に対する特異ベクトルと従来手法で得られる特異ベクトルとを比較すると、順序を除き、それぞれの特異ベクトルは、ほぼ一致していることが示された。たとえば、実施例４により得られた特異ベクトルの２番目のベクトル要素1.7290e-01は、従来手法（一括手法）の３番目のベクトルの要素に対応していることが示される。

【０１４３】

本発明の上述した情報検索を実行させるためのプログラムは、これまで知られたいかなる言語を使用しても記述することができ、例えばＣ言語、Ｃ＋＋言語、Ｊａｖａ（登録商標）といった言語を使用して本発明の方法を実行させるためのプログラムとすることができる。また、本発明の方法を実行させるためのプログラムは、フロッピー（登録商標）ディスク、ハードディスク、磁気テープといったコンピュータ可読な磁気記録媒体、光磁気ディスク、ＣＤ－ＲＯＭ、ＤＶＤといったコンピュータ可読な記録媒体に記憶させることができる。

【０１４４】

上述したように、本発明によれば、ドキュメントが逐次アップデートされていく大規模なデータベースにおける以前に計算された次元削減に関する結果を使用して、アップデートされたドキュメントを含むドキュメント－アトリビュート行列の特異値分解を効率的に実行することが可能となる。また、本発明により得られた特異ベクトルを使用してアップデートされたドキュメント－アトリビュート行列の次元削減を実行させた後、情報検索を行

うことにより、効率的で、かつ高精度の情報検索を実行することが可能となる。

【０１４５】

これまで本発明を図面に示した実施の形態に基づいて詳細に説明してきたが、本発明は、図面に示した実施の形態に限定されるものではなく、種々の変更、別の実施の形態を採用することが可能である。例えば、本発明においては、ドキュメントを文書ドキュメントとして説明してきたが、本発明において使用することができるドキュメントは、文書ドキュメントに限定されるものではなく、オーディオ、グラフィックス、動画といったドキュメントを含むことができる。

【図面の簡単な説明】

【図１】　本発明において使用するデータベースの概略構成を示した図。

【図２】　副行列Ａ１が加えられた時の更新を示した図。

【図３】　本発明のＱＲ分解方を用いた情報検索方法のフローチャート。

【図４】　一般的な行列Ｄに対するＱＲ分解の実行のための擬似コードを示した図。

【図５】　行列ＡのＱＲ分解により得られる行列の構成要素を示した概略図。

【図６】　特異値分解してｋ次元への次元削減を示した図。

【図７】　本発明の第２の実施形態の情報検索方法のフローチャートを示した図。

【図８】　図７に示したステップＳ１１において使用する共分散行列生成のための擬似コードを示した図。

【図９】　アトリビュート・ハッシュ・テーブルを変更するプロセスを示した概略図。

【図１０】　本発明において使用するアトリビュート・ハッシュ・テーブルを示した図。

【図１１】　アトリビュートの削除を実行する場合の概略図。

【図１２】　アトリビュートの削除と、生成される行列との関係を示した図。

【図１３】　アトリビュートの追加と、生成される行列との関係を示した図。

【図１４】　アトリビュートの追加と、生成される行列との関係を示した図。

【図１５】　本発明の情報検索システムを示した概略図。

【図１６】　タイム・スタンプとデータ・ファイルメイトからなるデータ例を示した図。

【図１７】　図１６で説明した個々のファイルを示した図。

【図１８】　図１６で説明した個々のファイルを示した図。

【図１９】　図１６で説明した個々のファイルを示した図。

【図２０】　図１６で説明した個々のファイルを示した図。

【図２１】　図１６で説明した個々のファイルを示した図。

【図２２】　図１６で説明した個々のファイルを示した図。

【図２３】　図１６で説明した個々のファイルを示した図。

【図２４】　本発明において使用したキーワードを例示した図。

【図２５】　ｄａｔａ１のみを処理した段階でのＳＵＭ－ＭＥＡＮ（Ｍ）$_1$と、ＳＵＭ（Ｍ）$_1$とを示した図。

【図２６】　ｄａｔａ２をアップデートしてデータを追加して得られた結果を示した図。

【図２７】　ｄａｔａ７までアップデートして得られたＳＵＭ－ＭＥＡＮ（Ｍ）$_1$を示した図。

【図２８】　アップデートされた共分散行列Ｃ’を使用して得られた特異ベクトルを示した図。

【図２９】　はじめから表５～表１０のデータとして与えられた行列を直接特異値分解して得た特異値を示した図。

【図３０】　削除の対象となったキーワードを示した図。

【図３１】　本発明によりダウンデートされた共分散行列により得られた特異値を示した図。

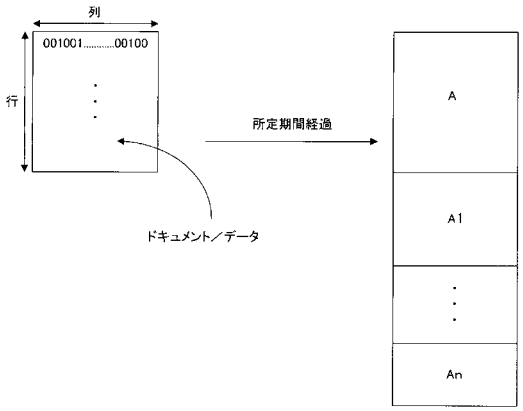
【図３２】　表５～表１０のデータから予めデータをダウンデートして直接共分散行列を生成して得られた特異値を示した図。

【符号の説明】

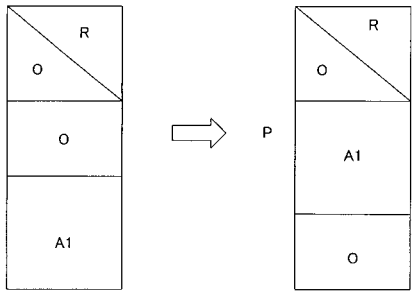
１０…コンピュータ

１２…データベース
１４…ネットワーク
１６…コンピュータ

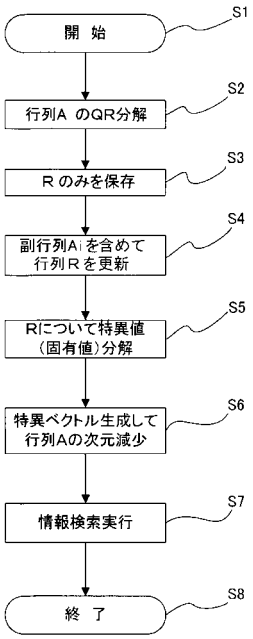
【図１】



【図２】



【図３】



【図４】

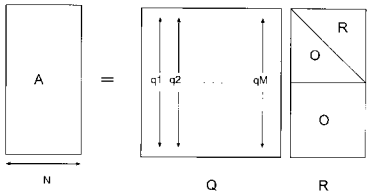
Pseudo Code for the QR-decomposition of $D \in R^{m \times n}$

$$D = \begin{bmatrix} \mathbf{d_1^T} \\ \mathbf{d_2^T} \\ \vdots \\ \mathbf{d_m^T} \end{bmatrix} = \begin{bmatrix} d_{;1} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{m1} & d_{m2} & \cdots & d_{mn} \end{bmatrix}$$

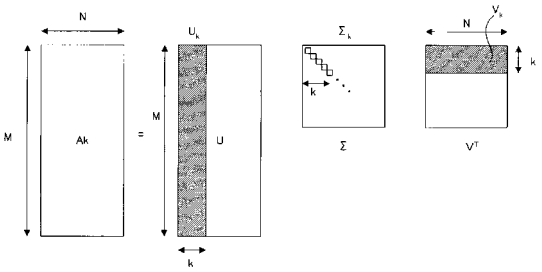
(when information about Q does not need to be stored).

for $k = 1, 2, \ldots, n$
  determine a reflector $Q_k = I - \gamma_k u^{(k)} u^{(k)T}$ such that
  $Q_k [d_{kk} \ \cdots \ d_{mk}]^T = [-\sigma_k \ 0 \ \cdots \ 0]^T$.
  set $d_{kk} = -\sigma_k$
  set $[d_{k+1k} \ \cdots \ d_{mk}]^T$ to zero
  for $j = k+1, k+2, \ldots, n$
    $[d_{kj} \ \cdots \ d_{mj}]^T = Q_k [d_{kj} \ \cdots \ d_{mj}]^T$

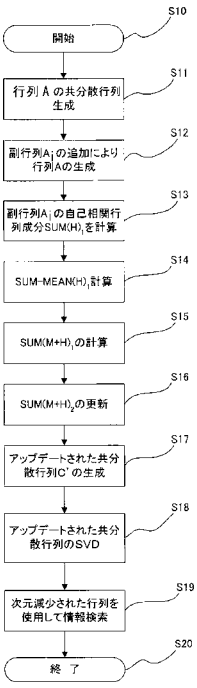
【図５】



【図６】



【図７】



【図８】

Pseudo Code for Constructing Covariance Matrix (basic case, no up/downdating)

Code to determine the covariance matrix for a set of n-dimensional vectors

$$\{d_i \quad d_i = (d(i,1), d(i,2), \ldots, d(i,N)\}^T \in R^N\}_{i=1}^M,$$

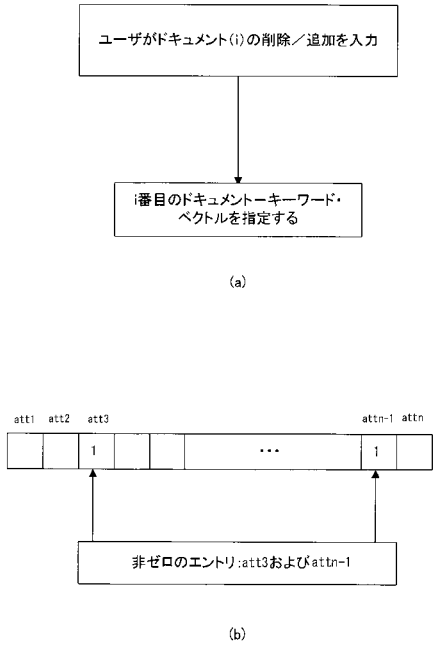
where:
  M is the cardinality of the set $\{d_i\}$ (unknown - to be determined),
  $D(\cdot)$ is the component-wise sum of the document vectors,
  $C(\cdot,\cdot)$ is an N-by-N covariance matrix,
  temp is a parameter for storing intermediate results during computations.

routine for initialization
$M = 0$
for $i = 1$ to N
  $D(i) = 0$
  for $j = 1$ to N
    $C(i,j) = 0$

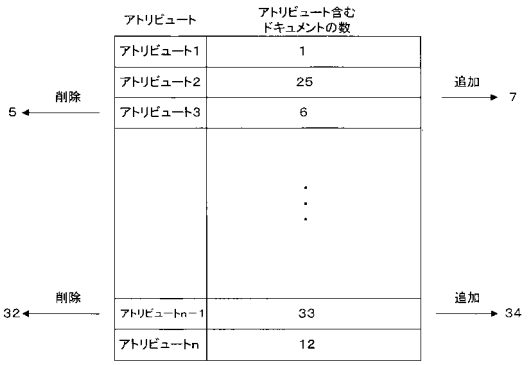
routine to compute component-wise sum of the document vectors D and cardinality M
for $i = 1$ to end-of-stack
for $j = 1$ to N
    $D(j) = D(j) + d(i,j)$
  $M = M + 1$

routine to compute $M \times C(\cdot,\cdot)$
for $i = 1$ to N
  for $j = 1$ to N
    temp = 0
    for $k = 1$ to M
      temp = temp + $d(k,i) \cdot d(k,j)$
    $C(i,j) = [\text{temp} - D(i) \cdot D(j)/M]/M$

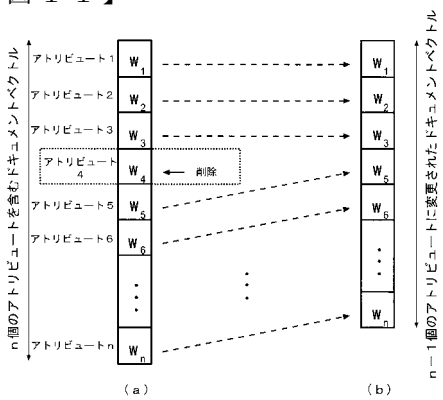
【図９】



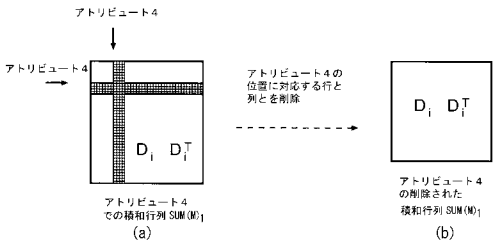
【図１０】



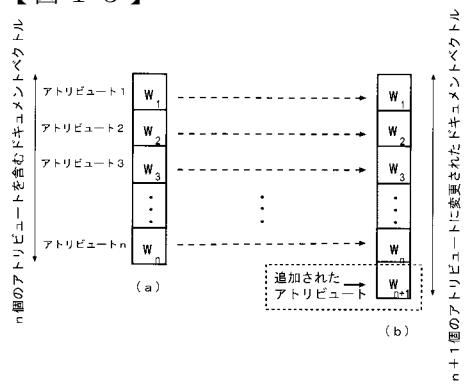
【図１１】



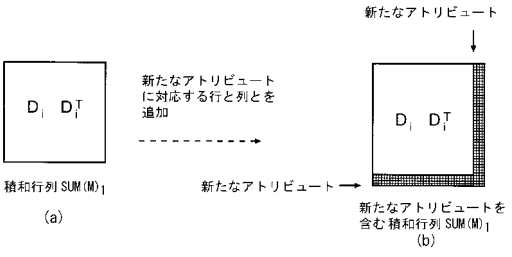
【図１２】



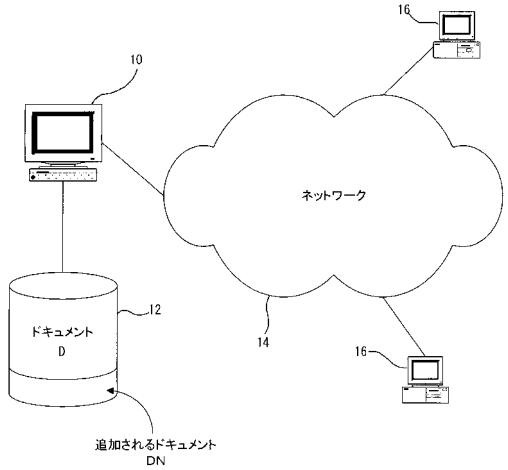
【図１３】



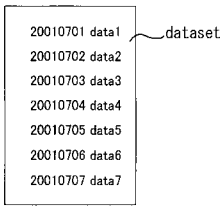
【図１４】



【図１５】



【図１６】



【図１７】

```
1 Clinton 1.0 AlGore 1.0 Chelsea 0.4 ↓
2 Clinton 1.0 AlGore 1.0 Bush 0.3 ↓
3 Clinton 1.0 AlGore 1.0 Japan 0.1 ↓
4 Clinton 1.0 AlGore 1.0 Chelsea 0.3 ↓
5 Clinton 1.0 AlGore 1.0 Japan 0.2 ↓
6 Clinton 1.0 AlGore 1.0 Chelsea 0.2 ↓
7 Clinton 1.0 AlGore 1.0 Japan 0.3 ↓
8 Clinton 1.0 AlGore 1.0 Chelsea 0.1 ↓
9 Clinton 1.0 AlGore 1.0 Japan 0.4 ↓
10 Clinton 1.0 AlGore 1.0 ↓
11 Clinton 1.0 AlGore 1.0 Hillary 0.5 Bush 0.3 ↓
12 Clinton 1.0 AlGore 1.0 Hillary 0.5 Chelsea 0.4 ↓
13 Clinton 1.0 AlGore 1.0 Hillary 0.5 Japan 0.1 ↓
14 Clinton 1.0 AlGore 1.0 Hillary 0.5 ↓
15 Clinton 1.0 AlGore 1.0 Hillary 0.5 Bush 0.3 ↓
16 Clinton 1.0 AlGore 0.5 Hillary 1.0 Chelsea 0.1 ↓
17 Clinton 1.0 AlGore 0.5 Hillary 1.0 Chelsea 0.4 ↓
18 Clinton 1.0 AlGore 0.5 Hillary 1.0 Japan 0.1 ↓
19 Clinton 1.0 AlGore 0.5 Hillary 1.0 ↓
20 Clinton 1.0 AlGore 0.5 Hillary 1.0 Bush 0.3 Japan 0.2 ↓
```

【図１８】

```
1 Clinton 1.0 Hillary 1.0 Japan 0.1 ↓
2 Clinton 1.0 Hillary 1.0 Chelsea 0.2 ↓
3 Clinton 1.0 Hillary 1.0 ↓
4 Clinton 1.0 Hillary 1.0 Bush 0.3 ↓
5 Clinton 1.0 Hillary 1.0 Chelsea 0.4 ↓
6 Java 1.0 JSP 1.0 apache 0.1 servlet 0.4 ↓
7 Java 1.0 JSP 1.0 servlet 0.3 ↓
8 Java 1.0 JSP 1.0 apache 0.2 ↓
9 Java 1.0 JSP 1.0 ↓
10 Java 1.0 JSP 1.0 apache 0.1 ↓
11 Java 1.0 JSP 1.0 servlet 0.2 ↓
12 Java 1.0 JSP 1.0 apache 0.3 ↓
13 Java 1.0 JSP 1.0 servlet 0.4 ↓
14 Java 1.0 JSP 1.0 servlet 0.1 ↓
15 Java 1.0 JSP 1.0 apache 0.1 ↓
16 Java 1.0 applet 0.5 JSP 1.0 servlet 0.2 ↓
17 Java 1.0 applet 0.5 JSP 1.0 apache 0.3↓
18 Java 1.0 applet 0.5 JSP 1.0 apache 0.2 ↓
19 Java 1.0 applet 0.5 JSP 1.0 servlet 0.1 ↓
20 Java 1.0 applet 0.5 JSP 1.0 ↓
```

【図１９】

```
1 Java 1.0 applet 1.0 JSP 0.5 apache 0.1 ↓
2 Java 1.0 applet 1.0 JSP 0.5 servlet 0.3 ↓
3 Java 1.0 applet 1.0 JSP 0.5 servlet 0.2 ↓
4 Java 1.0 applet 1.0 JSP 0.5 servlet 0.1 ↓
5 Java 1.0 applet 1.0 JSP 0.5 ↓
6 Java 1.0 applet 1.0 apache 0.1 ↓
7 Java 1.0 applet 1.0 ↓
8 Java 1.0 applet 1.0 servlet 0.3 ↓
9 Java 1.0 applet 1.0 apache 0.2 ↓
10 Java 1.0 applet 1.0 servlet 0.1 ↓
11 Bluetooth 1.0 dream 0.1 ↓
12 Bluetooth 1.0 cat 0.1 ↓
13 Bluetooth 1.0 god 0.1 ↓
14 Bluetooth 1.0 museum 0.1 ↓
15 Bluetooth 1.0 ↓
16 soccer 1.0 center 0.1 ↓
17 soccer 1.0 spring 0.1 ↓
18 soccer 1.0 silk 0.1 ↓
19 soccer 1.0 lecture 0.1 ↓
20 soccer 1.0 ↓
```

【図２０】

```
1 matrix 1.0 heat 0.1 ↓
2 matrix 1.0 power 0.1 ↓
3 matrix 1.0 magnet 0.1 ↓
4 matrix 1.0 gourmet 0.1 ↓
5 matrix 1.0 ↓
6 DNA 1.0 music 0.1 ↓
7 DNA 1.0 shark 0.1 ↓
8 DNA 1.0 snail 0.1 ↓
9 DNA 1.0 montage 0.1 ↓
10 DNA 1.0 ↓
11 cat 0.3 magnet 0.4 lecture 0.2 gourmet 1.0 ↓
12 center 0.2 silk 0.3 magnet 0.2 lecture 0.3 pattern 0.2 montage 1.0 ↓
13 whisper 0.2 saga 0.2 museum 0.2 torch 1.0 ↓
14 heat 0.3 spring 0.4 power 0.2 stock 0.1 silk 0.3 flag 1.0 ↓
15 heat 1.0 dream 0.2 flag 0.3 ↓
16 music 1.0 stock 0.2 insect 0.4 flag 0.1 ↓
17 heat 0.2 dream 1.0 spring 0.3 power 0.4 god 0.2 insect 0.2 magnet 0.4 pavillion 0.3 pattern 0.2 ↓
18 center 1.0 spring 0.1 shark 0.3 magnet 0.4 lecture 0.3 pavillion 0.4 museum 0.4 ↓
19 whisper 1.0 spring 0.1 god 0.3 shark 0.3 ↓
20 cat 1.0 magnet 0.3 lecture 0.2 museum 0.2 pattern 0.2 ↓
```

【図２１】

```
1 spring 1.0 stock 0.4 pavil ion 0.4 museun 0.2 ↓
2 power 1.0 snail 0.3 gourmet 0.3 ↓
3 whisper 0.3 god 1.0 ↓
4 spring 0.3 shark 1.0 snail 0.2 montage 0.4 ↓
5 spring 0.1 stock 1.0 magnet 0.2 latitude 0.2 pavillion 0.3 pattern 0.3 ↓
6 dream 0.2 shark 0.3 stock 0.1 insect 1.0 pavillion 0.3 museum 0.3 ↓
7 insect 0.2 silk 1.0 ↓
8 music 0.3 power 0.4 magnet 1.0 gourmet 0.4 flag 0.4 ↓
9 dream 0.4 god 0.3 insect 0.3 magnet 0.3 snai  1.0 gourmet 0.2 ↓
10 heat 0.3 music 0.2 shark 0.3 insect 0.3 magnet 0.2 latitude 1.0 lecture 0.3 muse um 0.3 ↓
11 whisper 0.2 shark 0.1 snail 0.3 saga 1.0 pattern 0.4 ↓
12 center 0.2 whisper 0.3 stock 0.4 silk 0.2  ecture 1.0 museum 0.4 ↓
13 music 0.3 pavillion 1.0 ↓
14 cat 0.4 snail 0.3 pavillion 0.3 museum 1.0 montage 0.4 ↓
15 music 0.4 whisper 0.4 cat 0.3 power 0.4 insect 0.4 pattern 1.0 ↓
16 dream 0.3 spring 0.2 magnet 0.2 gourmet 1.0 torch 0.2 ↓
17 dream 0.2 power 0.3 god 0.3 magnet 0.2 saga 0.3 pattern 0.2 montage 1.0 ↓
18 dream 0.2 silk 0.3 torch 1.0 ↓
19 flag 1.0 ↓
20 heat 1.0 dream 0.3 whisper 0.2 stock 0.3 sna l 0.3 gourmet 0.3 torch 0.3 ↓
```

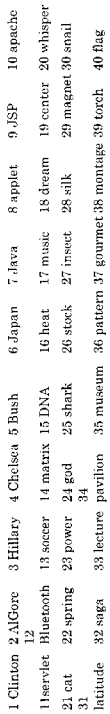
【図２２】

```
1 music 1.0 whisper 0.4 god 0.2 stock 0.2 insect 0.2 snai  0.4 pattern 0.4 montage 0.4 ↓
2 dream 1.0 center 0.3 snail 0.2 ↓
3 center 1.0 shark 0.2 latitude 0.3 lecture 0.3 pattern 0.4 ↓
4 dream 0.2 whisper 1.0 spring 0.4 power 0.3 lecture 0.3 gourmet 0.4 montage 0.4 f lag 0.2 ↓
5 center 0.2 cat 1.0 spring 0.2 shark 0.1 gourmet 0.2 ↓
6 spring 1.0 silk 0.2 saga 0.4 gourmet 0.4 ↓
7 cat 0.3 power 1.0 pavillion 0.3 flag 0.2 ↓
8 god 1.0 ↓
9 center 0.3 god 0.1 shark 1.0 insect 0.4 snail 0.3 ↓
10 whisper 0.4 stock 1.0 silk 0.4 snail 0.4 museum 0.2 ↓
11 god 0.3 shark 0.1 insect 1.0 magnet 0.3 ↓
12 stock 0.3 silk 1.0 snail 0.3 museum 0.4 flag 0.4 ↓
13 heat 0.4 whisper 0.4 god 0.3 magnet 1.0 saga 0.2 lecture 0.1 museum 0.4 ↓
14 power 0.4 snail 1.0 museum 0.3 ↓
15 latitude 1.0 pavillion 0.4 ↓
16 stock 0.3 saga 1.0 torch 0.4 ↓
17 lecture 1.0 museum 0.2 pattern 0.2 gourmet 0.2 montage 0.2 ↓
18 music 0.2 spring 0.2 insect 0.2 latitude 0.2 pavillion 1.0 ↓
19 dream 0.2 museum 1.0 ↓
20 silk 0.1 lecture 0.3 museum 0.2 pattern 1.0 montage 0.3 ↓
```

【図２３】

```
1 gourmet 1.0 ↓
2 dream 0.4 center 0.2 whisper 0.3 spring 0.3 power 0.3 montage 1.0 torch 0.3 ↓
3 whisper 0.3 magnet 0.2 pavillion 0.2 torch 1.0 ↓
4 heat 0.3 music 0.3 power 0.3 shark 0.4 lecture 0.4 pavillion 0.3 pattern 0.2 gou rmet 0.4 torch 0.2 flag 1.0 ↓
5 heat 1.0 music 0.4 center 0.2 god 0.3 insect 0.2 latitude 0.1 saga 0.1 pattern 0 .3 torch 0.1 ↓
6 music 1.0 power 0.2 stock 0.4 silk 0.2 torch 0.3 ↓
7 dream 1.0 insect 0.2 lecture 0.3 museum 0.2 ↓
8 center 1.0 god 0.2 stock 0.2 snail 0.1 lecture 0.2 ↓
9 whisper 1.0 cat 0.2 latitude 0.2 lecture 0.2 ↓
10 cat 1.0 pattern 0.2 gourmet 0.3 ↓
11 heat 0.4 spring 1.0 snail 0.3 saga 0.3 museum 0.2 gourmet 0.4 ↓
12 power 1.0 silk 0.3 snail 0.3 lecture 0.3 flag 0.2 ↓
13 god 1.0 lecture 0.4 museum 0.2 montage 0.4 ↓
14 power 0.3 shark 1.0 museum 0.2 montage 0.4 torch 0.2 flag 0.4 ↓
15 music 0.2 dream 0.3 spring 0.3 god 0.2 stock 1.0 montage 0.4 ↓
16 heat 0.3 dream 0.1 cat 0.2 shark 0.3 insect 1.0 pattern 0.2 ↓
17 power 0.2 stock 0.2 silk 1.0 ↓
18 magnet 1.0 latitude 0.4 montage 0.3 flag 0.4 ↓
19 dream 0.1 stock 0.2 silk 0.3 magnet 0.3 snail 1.0 lecture 0.2 museum 0.1 ↓
20 god 0.1 stock 0.2 latitude 1.0 ↓
```

【図２４】



【図２５】

```
document 1: new keyword = Clinton
document 1: new keyword = AlGore
document 1: new keyword = Chelsea
document 2: new keyword = Bush
document 3: new keyword = Japan
document 11: new keyword = Hillary
```

data1で追加される新しいキーワード。
これらがハッシュテーブルに追加される。

SUM-MEAN(M)$_1$ (N = 6)
[20.0 17.5 1.9 1.2 1.4 7.5]

data1 処理後のSUM-MEAN(M)$_1$

SUM(M)$_1$ (N = 6)

```
20.00 17.50 1.90 1.20 1.40 7.50
      16.25 1.65 1.05 1.25 5.00
            0.63 0.00 0.00 0.70
                  0.36 0.06 0.60
                        0.36 0.35
                              6.25
```

data1 が処理された後のSUM(M)$_1$。
対称行列なので半分だけ書き出した。

【図２６】

```
document 26: new keyword = Java
document 26: new keyword = JSP
document 26: new keyword = apache
document 26: new keyword = servlet
document 36: new keyword = applet
```

data2で追加された新しいキーワード。
これら5つがハッシュテーブルに追加され，
合計で11個のキーワードとなる。

SUM-MEAN(M)$_1$ (N = 11)
[25.0 17.5 2.5 1.5 1.5 12.5 15.0 15.0 1.3 1.7 2.5]

data2 処理後のSUM-MEAN(M)$_1$

SUM(M)$_1$ (N = 11)

```
25.00 17.50 2.50 1.50 1.50 12.50 0.00 0.00 0.00 0.00 0.00
      16.25 1.65 1.05 1.25 5.00 0.00 0.00 0.00 0.00 0.00
            0.83 0.00 0.00 1.30 0.00 0.00 0.00 0.00 0.00
                  0.45 0.06 0.90 0.00 0.00 0.00 0.00 0.00
                        0.37 0.45 0.00 0.00 0.00 0.00 0.00
                              11.25 0.00 0.00 0.00 0.00 0.00
                                    15.00 15.00 1.30 1.70 2.50
                                          15.00 1.30 1.70 2.50
                                                0.29 0.04 0.25
                                                      0.51 0.15
                                                            1.25
```

data2 が処理された後のSUM(M)$_1$。
11x11 の対称行列となる。

【図２７】

SUM-MEAN(M)$_1$(N=40)
[25.0 17.5 2.5 1.5 1.5 12.5 25.0 17.5 1.72.7 12.5 5.0 62 4.8 5.9 6.7 5.0 4.7 6.0
5.7 6.4 5.0 5.3 6.8 6.7 6.6 5.0 5.4 5.5 6.8 6.7 5.4 6.4 3.5 5.0 6.5 5.6 6.0 5.2 4.4]

data7 処理後のSUM-MEAN(M)$_1$

【図２８】

特異値 (Largest 10)

```
0.3161665995009
0.2324900894427
0.0590495015116
0.0589660332610
0.0487117233259
0.0404724327529
0.0383981484518
0.0367731637155
0.0359210981307
0.0358693365225
```

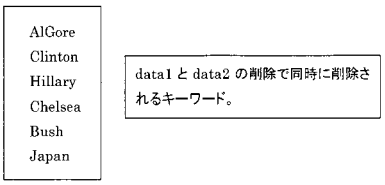
data2で追加された後のSUM(M)$_1$。
11x11 の対称行列となる。

【図２９】

特異値 (Largest 10)

0.3161665995009
0.2324900894427
0.0590495015116
0.0589660332610
0.0487117233259
0.0404724327529
0.0383981484518
0.0367731637155
0.0359210981307
0.0358693365225

【図３０】



【図３１】

特異値 (Largest 10)

0.1987144620508
0.0690170884491
0.0566614854733
0.0537575326572
0.0514825460617
0.0502895403078
0.0502170789069
0.0497783735464
0.0480041373927
0.0467372731528

【図３２】

特異値 (Largest 10)

0.1987144620508
0.0690170884491
0.0566614854733
0.0537575326572
0.0514825460617
0.0502895403078
0.0502170789069
0.0497783735463
0.0480041373927
0.0467372731528

フロントページの続き

(72)発明者　寒川　光
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
(72)発明者　小林　メイ
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
(72)発明者　青野　雅樹
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
(72)発明者　竹内　広宜
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　深津　始

(56)参考文献　北研二、ほか，意味および感性に基づくマルチメディア・クロス・コンテンツ検索に関する研究，財団法人電気通信普及財団　調査研究報告書，第１９号，４１１－４１８頁

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F 17/30