Title	文書クラスタリングの技法 : 文献レビュー
Sub Title	Techniques of document clustering : a review
Author	岸田, 和明(Kishida, Kazuaki)
Publisher	三田図書館・情報学会
Publication year	2003
Jtitle	Library and information science No.49 (2003.) ,p.33- 75
JaLC DOI	
Abstract	The document clustering technique is widely recognized as a useful tool for informationretrieval， organizing web documents， text mining and so on． The purpose of this paper is toreview various document clustering techniques， and to discuss research issues for enhancingeffectiveness or efficiency of the clustering methods． We explore extensive literature onnon - hierarchical methods （ single - pass methods ） ， hierarchical methods （ single - link ， completelink ， etc． ） ， dimensional reduction methods （ LSI ， principal component analysis ， etc． ） ， probabilisticmethods ， data mining techniques ， and so on． In particular ， this paper focuses on typicaltechniques ， such as the k - means algorithm ， the leader - follower algorithm ， self - organizing map （ SOM ） ， single一 or complete - link methods ， bisecting k - means methods ， latent semantic indexing （ LSI ） ， Gaussian - Mjxture model and so on． After reviewing the techniques and algorithms ， wediscuss research issues on document clustering ； computational complexity ， feature extraction （ selection of words ） ， methods for defining term weights and similarity ， and evaluation of results．
Notes	展望論文
Genre	Journal Article
URL	https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=AN00003152-00000049-0033

Library and Information Science No. 49 2003

展望論文

文書クラスタリングの技法：文献レビュー

Techniques of Document Clustering: A Review

岸　田　和　明
Kazuaki KISHIDA

Résumé

The document clustering technique is widely recognized as a useful tool for information retrieval, organizing web documents, text mining and so on. The purpose of this paper is to review various document clustering techniques, and to discuss research issues for enhancing effectiveness or efficiency of the clustering methods. We explore extensive literature on non-hierarchical methods (single-pass methods), hierarchical methods (single-link, complete-link, etc.), dimensional reduction methods (LSI, principal component analysis, etc.), probabilistic methods, data mining techniques, and so on. In particular, this paper focuses on typical techniques, such as the k-means algorithm, the leader-follower algorithm, self-organizing map (SOM), single- or complete-link methods, bisecting k-means methods, latent semantic indexing (LSI), Gaussian-Mixture model and so on. After reviewing the techniques and algorithms, we discuss research issues on document clustering; computational complexity, feature extraction (selection of words), methods for defining term weights and similarity, and evaluation of results.

岸田和明：駿河台大学文化情報学部，埼玉県飯能市阿須698
Kazuaki Kishida: Surugadai University, 698 Azu, Hanno, Saitama
e-mail: kishida@surugadai.ac.jp
受付日：2004年2月25日　改訂稿受付日：2004年6月28日　受理日：2004年8月19日

I. はじめに

情報検索の分野では，図書や雑誌論文などの文書(documents)の集合を内容的に均質ないくつかの群に分けるための，文書クラスタリング(document clustering)の研究が，長年にわたって試みられてきた。その応用目的としては以下のものが挙げられる（岸田(2003)[46]）。

1. 従来の情報検索にクラスタリングの結果を直接適用することによって検索性能を向上させる。
2. 検索結果としての文書集合をグループ化してわかりやすく提示する。
3. キーワード検索とは異なった，ブラウジングに基づく検索様式を提供する。

このほかにも，検索の処理における作業効率の改善のために，文書クラスタリングが利用されることもある(Salton と McGill (1983)[73])。

本稿では，以上のような，情報検索を応用目的とした文書クラスタリングの代表的な手法やアルゴリズムを概観し，整理することを試みる。現在，この種の手法・アルゴリズムは多岐にわたっており，それらを整理して，その特徴を把握することには意義があろう。このため，本稿は，単に文献を紹介するだけではなく，ある程度，その技法を具体的に記述し，その特徴を論じていく。

なお，本稿は文書データに対するクラスタリング一般について網羅することを目的とするものではない。文書のクラスタリング自体は，WWWの組織化やテキストマイニング（あるいはデータマイニング）など，現在，さまざまな領域で研究が進められており，本稿では，それらに関しては一部を論じるのみである。特に，データマイニングにおけるクラスタリング手法については，いくつかの研究例を除き（第III章E節参照），本稿の対象外とする。これらについては，Jain ら(1999)[39]や Kolatch (2001)[51]，神嶌(2003)[42]などのレビュー論文がある。また，現在盛んに研究されているテキスト分類(text categorization)もまた本稿の範囲外である。テキスト分類はいわば「教師付きの(supervised)分類」であり，正解付きのデータが学習用に与えられているという条件の下での分類である。それに対して，本稿で扱う文書クラスタリングは，「教師なしの(unsupervised)分類」に相当し，この点で，機械学習を応用したテキスト分類の技法とは一線を画している（図書館・情報学分野におけるテキスト分類に関する過去の研究例については岸田(2001)[45]などを参照)。

以下，本稿では，第II章で文書クラスタリングの特徴と類型とを議論したあと，第III章にて，これまで提案されてきた文書クラスタリングの技法・アルゴリズムを概観する。その結果に基づいて，第IV章では，文書クラスタリングの研究における諸問題を整理する。

なお，“document”に対する訳語としては，図書館・情報学の分野では伝統的に「文献」が当てられてきた。しかし，“document clustering”の

場合，最近では，いわゆるWeb文書や電子文書をも対象にすることが多く，本稿では，"document"という概念に対して，統一的に「文書」という用語を用いておく。

また，本稿では，全体を通して，数学的な記号をなるべく1つの意味で用いるよう努めるが，場合によっては，同一の記号が異なる意味で使われることもある。そのような場合には，混乱が生じないよう，その旨説明を付けることとする。なお，本稿において，統一的に使用される記号の主なものを以下に示す（ここで，i, j, k は添字とする）。

- x_{ij}: 文書 d_i における語 t_j の出現延べ回数
- n_j: 文書集合全体における語 t_j の出現文書数
- $\tilde{n}_k$: クラスタ C_k に含まれる文書総数
- w_{ij}: 文書ベクトル $\mathbf{d}_i$ における語 t_j についての要素（重み）
- $\tilde{w}_{kj}$: クラスタベクトル $\mathbf{c}_k$ における語 t_j についての要素（重み）
- $|A|$: 集合 A に含まれる要素数
- $\|\mathbf{d}_i\|$: ベクトル $\mathbf{d}_i$ のノルム

II. 文書クラスタリングの特徴と類型

A. 文書クラスタリングの一般的特徴

N 件の文書 d_i $(i=1,\ldots,N)$ を要素とする集合 $D=\{d_1, d_2, \ldots, d_N\}$ を，いくつかのクラスタ $C_1, C_2, \ldots, C_L$ に分割することを考える。すなわち，

$$D=C_1\cup C_2\cup\cdots\cup C_L=\{C_k\}_{k=1}^{L}$$

である。この際に，1件の文書が唯一のクラスタに属するように分割する場合，

$$C_k\cap C_h=\phi \quad (k\neq h)$$

と，複数のクラスタに含まれることを許す場合，

$$C_k\cap C_h\neq\phi \quad (k\neq h)$$

とがある。前者は「排他的」，後者は「非排他的」なクラスタリングである。

文書クラスタリングを実行する場合，一般に，文書は，各語の重みから構成されるベクトル

$$\mathbf{d}_i=(w_{i1}, w_{i2}, \ldots, w_{iM})^T \qquad (1)$$

として表現される。ここで w_{ij} は，i 番目の文書における語 t_j の重みであり $(j=1,\ldots,M)$，T は転置記号を意味する。なお，M は文書集合 D に含まれる語の異なり数とする。

例えば，ベクトル空間モデル[73]に基づけば，2件の文書間の類似度は，それらのベクトルの成す角度の余弦として定義される。すなわち，

$$s(\mathbf{d}_i, \mathbf{d}_h)=\frac{\sum_{j=1}^{M} w_{ij}w_{hj}}{\sqrt{\sum_{j=1}^{M} w_{ij}^2}\sqrt{\sum_{j=1}^{M} w_{hj}^2}} \qquad (2)$$

であり，この類似度 s に基づいて，クラスタリングを実行できる。

もちろん，文書ベクトルに依拠しないクラスタリングも可能である。例えば，計量書誌学的に共引用や書誌結合などを測定すれば，文書間の類似度を規定できる。共引用を使う場合には，類似度を計算する対象である2つの文書を両方ともに引用している文書数を c として，類似度を $c/\sqrt{ab}$ などとすればよい。ここで，a は2つの文書のうちの一方の文書を引用している文書総数，b は他方の文書を引用している文書総数である。また，WWWのリンク構造を利用すれば，同様な方法を応用した，Web文書のクラスタリングも可能であろう。ただし，情報検索を応用目的とした文書クラスタリングの場合，文書ベクトルを用いずに類似度を計算することは現時点では少なく，本稿では，この方法についてはこれ以上言及しない。計量書誌学的な観点からの文書のクラスタリングについては，Small (1997, 1999)[78),79)] が参考になる。また，Webのリンク構造を応用したクラスタリングに関しては，最近では，WangとKitsuregawa (2002)[89)] の研究などがある。

一般に，クラスタリングの対象文書数 N はかなり大きい。ある1件の検索質問に対する出力文書集合をクラスタに分割する場合などは例外としても，あるデータベースをクラスタリング技法を使って構造化しようとするようなときには，N は非常に大きな数となる。このことは，単連結法や完全連結法などの階層的クラスタ分析法を適用しようとする際に大きな障壁となる。

また，一般に，文書ベクトルの次元数 M も大きなものになる。M はすでに述べたように，文書集合 D における異なり語数（総数）であり，実際の状況にもよるが，多くの場合，$M>N$ が成立すると考えられる。したがって，文書×語の重み行列

$$\mathbf{W}=(w_{ij}) \quad i=1,\ldots,N;\ \ j=1,\ldots,M \qquad (3)$$

を考えると，これは横長となり，しかもその要素の多くが0である，疎(sparse)な行列である。この状況は，通常の主成分分析や因子分析などの多変量解析法が想定しているデータ行列とは異なっている。したがって，文書クラスタリングにこの種の手法の適用を試みる場合には，通常，行と列を入れ替えたり（すなわち$\mathbf{W}^T$），何らかの方法によって語数を減らすなどの工夫を加えなければならない。また，特にMの大きさが問題にならなくとも，一般的に，語の中には専門的なものとそうでないものとがあり，非専門用語を除くことによって，処理の効率化を実現できる。このように，文書クラスタリングにおいては，文書ベクトルを構成する特徴（すなわち語）をいかに取捨選択するか，という問題が重要になる（第IV章参照）。

さらに，(2)式が示すように，ベクトル間の類似度は，ユークリッド距離でなく，余弦係数（またはそれに類した尺度）に基づいて計算されることが多い。これもまた，文書クラスタリングの特徴のひとつである。ただし，もし文書ベクトルが標準化され，その長さが1，すなわち，$\tilde{\mathbf{d}}_i=\mathbf{d}_i/\|\mathbf{d}_i\|$であるならば，「順位付け」という点では平方ユークリッド距離と余弦係数とは同一の結果を与える(Schutze と Silverstein (1997)[74])。例えば，2つの標準化された文書ベクトル$\tilde{\mathbf{d}}_i$と$\tilde{\mathbf{d}}_h$との間の平方ユークリッド距離は，

$$\begin{aligned}\|\tilde{\mathbf{d}}_i-\tilde{\mathbf{d}}_h\|^2&=\Sigma(w_{ij}-w_{hj})^2\\&=\Sigma w_{ij}^2-2\Sigma w_{ij}w_{hj}+\Sigma w_{hj}^2\\&=1-2\Sigma w_{ij}w_{hj}+1\\&=2(1-\Sigma w_{ij}w_{hj})\end{aligned}$$

であり，一方，式(2)については，文書ベクトルの長さが1に標準化されているのでその分母は1となるから，$\Sigma w_{ij}w_{hj}$である。したがって，「順位付け」という点では，両者は同一の結果を与える。実際に，第III章で概観するように，類似度ではなく，距離（非類似度）によって文書集合のクラスタ化を試みている研究例も少なくない。

以上のように，文書クラスタリングには，

1. クラスタリングの対象数Nがかなり大きいこと（検索結果集合をクラスタリングする場合を除く）
2. 通常，特徴の数Mがクラスタリングの対象数Nよりも大きくなること
3. 余弦係数などによる類似度に基づく場合が多いこと

などの特徴がある。このような特徴のために，文書クラスタリングには一種独特な工夫が必要となり，その結果，1つの研究領域が形成されるに至ったと捉えることもできよう。

B. 文書クラスタリング技法の類型

一般に，クラスタリングの方法は，階層的なものと非階層的なものとに大別できる[39]。すでに述べたように，文書クラスタリングの場合には，その分類対象の数が大きいので，階層的な方法（単連結法，完全連結法，群平均法など）の実行は難しい。なぜなら，階層的な方法の場合，N件の文書の各ペア（組）の類似度を求めなければならず，その計算量は，少なくとも，$O(N^2)$になる。さらに，クラスタを階層的に構成するために，類似度データから，適切な文書の組の類似度を探索するのにも一定の計算が必要となる。

このため，情報検索の分野では早い時期（1960年代）から非階層的な方法の適用が探究されてきた。その例としては，Dattolaの方法やRocchioの方法などがあり(Yu (1974)[97])，これらは，文書が記録されたファイルを1度走査するだけで，クラスタを構成しようと試みるので，単一パス・アルゴリズム (single-pass algorithm) と呼ばれることがある。この結果，その計算量は基本的には$O(N)$程度に抑えられ（詳細は第IV章A節参照），この点では，大規模文書集合のクラスタリングに適しているといえる。ただし，単一パス・アルゴリズムには，その結果が文書の処理の順序に大きく依存するという欠点がある。なお，実際には，クラスタリングの結果を洗練するために，ファイルが複数回走査される場合もあるが，本稿では，このような非階層的な方法を，一括して「単一パス・アルゴリズム」と呼んでおく。一般的には，非階層的な方法として，k-means法がよく

知られており，後述するように，この技法は文書クラスタリングにとっても重要である。

もちろん，大規模文書集合に対して階層的な方法を適用する研究も試みられている（第III章B節参照)。単一パス・アルゴリズムでは，文書集合Dがいくつかの部分集合に「平面的に」分割されるだけであるが，階層的な方法の場合には，さらに，クラスタが上位・下位に構造化されるわけであり，このことは，情報の組織化・検索の観点からはより望ましいと考えられる。

階層的なクラスタリングには，凝集型と分割型とがある。一般的な単連結法などの階層的クラスタ分析法は凝集型であり，個々の対象（ここでは文書）から出発して，類似度行列を使って，それらを次第に大きなクラスタに組み上げていく。一方，分割型の場合には，全文書集合Dから出発して，その分割を再帰的に繰り返すことによって，階層を構成する。

また，すでに述べたように，行列$\mathbf{W}$（または$\mathbf{W}^T$）に対して主成分分析のような次元縮約の方法を適用して，その結果から文書をクラスタに分割することも可能である。特に，情報検索の分野では，LSI (latent semantic indexing) (Deerwester ほか (1990)[20]) において特異値分解 (singular value decomposition: SVD) を利用した次元縮約の方法が利用されているという背景もある（詳細は第III章C節参照)。なお，LSIは，語数を減らすための「特徴抽出」の方法としても重要である。

そのほか，実例は少ないが，確率的なモデルに基づく方法も文書クラスタリングに利用されている。確率的な方法は，パターン認識の分野などではむしろ中心的な役割を果たしており，理論的に，より洗練されている。ただし，本論が対象とする「教師なしの分類」の場合には，確率分布のパラメータを推定するために，EMアルゴリズムなどの近似的な方法を使わざるを得ないことや，文書集合における語の分布を，数学的に扱いやすい確率分布（正規分布など）で近似しなければならないなどの問題がある（第III章D節参照)。

以上，文書クラスタリングの方法は，まず，

1. 単一パス・アルゴリズム
2. 階層的クラスタ分析法
3. 主成分分析などの次元縮約法の利用
4. 確率モデルに基づく方法

に類型化される。そのほか，関連ルール (association rule) のようなデータマイニングの方法を適用した例がある（第III章E節参照)。また，近年，情報検索システムにおけるGUIのひとつとして，文書集合の構造を可視化する技術が盛んに研究されているが，その中には，文書集合のクラスタリングを伴うものがある（第III章F節)。

一般的にいえば，クラスタリングの技法には，グラフ理論に基づく方法など，上記の1.〜4.以外のものもある（グラフ理論については，文書クラスタリングへの若干の応用例はある。例えば第III章B節のSTCアルゴリズムなどを参照)。特に，近年のパターン認識やデータマイニングの発展によって，一般的なクラスタリングの方法は多様化・高度化しているといえる。例えば，2次元または3次元での空間的なデータのクラスタリングは，空間中に複雑な形状で分布しているクラスタを識別する必要から，従来の文書クラスタリングとは異なる方法が種々研究されている（詳細は，Jain ほか (1999)[39] や Kolatch (2001)[51]，神嶌 (2003)[42] のレビューを参照)。これらの方法は現時点では文書クラスタリングの方法として明示的に列挙することはできないが，もちろん将来的には，そのひとつとして数える必要が生じるかもしれない。

次章では，具体的な文書クラスタリングの方法・アルゴリズムを上記の類型に従って，概観していく。

III. 文書クラスタリングの技法

A. 単一パス・アルゴリズム

1. k-means 法の適用

k-means 法は，一般的には，非階層的なクラスタリングにおける標準的なアルゴリズムであり，文書クラスタリングへの直接的な応用例もいくつかある。

例えば，分散型検索 (distributed retrieval) の実現を目的として，Xu と Croft (2000)[95] は k-means 法を利用している。この研究では，文書とクラスタ間の類似度は，

$$s(d_i, C_k) = \sum_{j: x_{ij} \neq 0} \frac{x_{ij}}{l_i} \log \frac{x_{ij}/l_i}{(x_{ij}+\tilde{x}_{kj})/(l_i+\tilde{l}_k)}$$

で定義されている。ここで，x_{ij} は文書 d_i における語 t_j の出現回数，$\tilde{x}_{kj}$ はクラスタ C_k における語 t_j の出現回数である。また l_i は文書 d_i の長さ（延べ語数），すなわち，$l_i = \sum_{j=1}^{M} x_{ij}$ であり，$\tilde{l}_k$ は同じく，クラスタ C_k の長さ（延べ語数）を意味する。この式は，Kullback-Leibler の距離に若干の修正を加えたものである。

一般的な k-means 法は，以下の手順で実行される（Duda ほか (2001)[22]）。

一般的な k-means アルゴリズム

(1) クラスタの個数 L を決め，初期的なクラスタのベクトルを L 個生成する ($\mathbf{c}_1, \ldots, \mathbf{c}_L$)。
(2) N 件の分類対象を，それぞれ，最も近いベクトル $\mathbf{c}_k$ に従って分類し（そのクラスタに割り当てて），ベクトル $\mathbf{c}_k$ を更新する。
(3) もしベクトル $\mathbf{c}_k$ が変化しなくなれば処理を終了し，そうでなければ (2) に戻る。

一般的な k-means 法では，クラスタベクトル $\mathbf{c}_k$ が安定するまで，分類対象のクラスタへの割り当てとクラスタベクトルの更新とが反復的に繰り返される。この反復回数を r とすれば，一般的な k-means 法の計算量は $O(NMLr)$ となる (Duda ほか (2001)[22])。ここで N はこれまでどおり文書数（標本の大きさ），M は語数（ベクトルの次元），L はクラスタ数である。なお，「k-means」の「k」はクラスタの個数を意味するが，本稿では，クラスタの個数には，一貫して，記号 L を使用する。

階層的クラスタ分析の計算量が N^2 に比例してしまうのに対して，k-means 法の計算量は少ない。これは大きな利点である。しかし，クラスタ個数 L を先験的に与える必要があり，また，同時に，それらのクラスタの核となる種子点 (seeds) を初期的に設定しなければならない。これらは，教師なしの分類である文書クラスタリングの状況においては，望ましい条件ではない。このため，後述するように，単一パス・アルゴリズムとして計算量の少ない k-means 法を基本的に採用しつつ，クラスタ個数や種子点の設定などに工夫を加える試みがいくつかなされている。

2. Willett のアルゴリズム

Willett (1980)[91] によるアルゴリズムは，一種の k-means 法であるが，転置索引ファイルを使って初期的なクラスタを設定するため，先験的にクラスタ個数や種子点を与える必要はない。

Willett のアルゴリズム

(1) 転置索引ファイルを使って，初期的なクラスタベクトルを M 個生成する ($\mathbf{c}_1, \ldots, \mathbf{c}_M$)。クラスタ個数 L を M に設定する ($L \leftarrow M$)。
(2) 以下の手順 (a) と (b) を一定回数，反復的に繰り返す。
(2-a) 各文書 $d_1, \ldots, d_N$ に対して，文書ベクトル $\mathbf{d}_i$ と，クラスタのベクトル $\mathbf{c}_k$ との類似度 $s(\mathbf{d}_i, \mathbf{c}_k)$ を計算し ($k = 1, \ldots, L$)，その値が最大のクラスタに文書 d_i を割り当てる（同点の場合には，複数のクラスタに割り当てる）。$\mathbf{c}_k$ を更新する。
(2-b) 文書がまったく割り当てられなかったクラスタを削除する。残ったクラスタの個数を L として，それに対応するクラスタベクトルを $\mathbf{c}_1, \ldots, \mathbf{c}_L$ とする。

段階 (1) における初期的なクラスタのベクトルの生成には，転置索引ファイルが利用される。すなわち，ある 1 つの語を共有する文書群を 1 つのクラスタと見なし，初期時点では，t_j ごとに，合計 M 個のクラスタが存在すると考える。そして，上記の手順 (2-a) と (2-b) の反復計算の過程の中で，次第にクラスタ個数が減少していき，最終的に残った L 個のクラスタが結果として出力されるというしくみである。

クラスタベクトルは，そのクラスタに属する文書のうちの一定件数以上に出現する語を抜き出し

て構成する。すなわち，$\tilde{n}_{j|k}$ をクラスタ C_k に割り当てられた文書の中で語 t_j を含んでいる文書の数として，そのクラスタにおける語 t_j の重み $\tilde{w}_{kj}$ を

$$\tilde{w}_{kj}=\begin{cases}1, & \tilde{n}_{j|k}>\tau \text{ の場合}\\ 0, & \text{それ以外}\end{cases} \tag{4}$$

とする。ここで，τ は閾値である。τ としては，クラスタ C_k に含まれる文書数 $\tilde{n}_k$ を使って，$\log_2 \tilde{n}_k$，$\sqrt{\tilde{n}_k}$ などのように設定することも考えられるが，Willett (1980)[91] では，$\tilde{n}_k/3$ が採用されている。式 (4) より，当然，クラスタのベクトル $\mathbf{c}_k$ は 2 値ベクトルとなる。文書ベクトルもまた，2 値ベクトルであり，両者の類似度の計算には Dice 係数，

$$s(\mathbf{d}_i, \mathbf{c}_k)=\frac{2\sum_{j=1}^{M} w_{ij}\tilde{w}_{kj}}{\sum_{j=1}^{M} w_{ij}+\sum_{j=1}^{M}\tilde{w}_{kj}} \tag{5}$$

が用いられる。

この方法では先験的にクラスタ個数 L を与える必要はないが，反復計算によって L が一定値に収束する保証はない（最終的に単一のクラスタにまとまってしまう可能性もある）。したがって，上記の手順 (2) の反復回数，または，L やクラスタベクトルの収束条件を前もって設定する必要がある。

3. 平均クラスタリング・アルゴリズム

クラスタリングの結果に対する「良し悪し」を評価する何らかの基準関数を設定し，その基準に照らして最適な分割 $\{C_k\}_{k=1}^{L}$ を求めることを考える。例えば，基準関数としては，

$$J_e(\{C_k\}_{k=1}^{L})=\sum_{k=1}^{L}\sum_{i:\, d_i\in C_k}\|\mathbf{d}_i-\mathbf{m}_k\|^2 \tag{6}$$

で定義される「平方誤差の総和 (sum-of-squared error)」などが考えられる。ここで，$\mathbf{m}_k$ は重心ベクトルで，

$$\mathbf{m}_k=\frac{1}{\tilde{n}_k}\sum_{i:\, d_i\in C_k}\mathbf{d}_i \tag{7}$$

である（$\tilde{n}_k$ はクラスタ C_k に属する文書数）。

この指標は，式 (1) によって定義されるベクトル間のユークリッド距離に基づいて，各クラスタがどれだけ密集しているかを表すものである。ベクトル間の距離が小さいほど，それらの文書は類似していると考えられるので，平方誤差の総和の値が小さいほど，各クラスタはまとまっており，クラスタリングは成功していると判断できる。しかし，平方誤差の総和のような基準に従って厳密にクラスタを決定するということは，莫大な数（おおよそ $L^N/L!$ 通り）の分割の候補から最適なものを見つけるという，非常に難しい問題であるので，実際には，反復的な計算によって近似的な最適解を求めるのが一般的である[22]。

Kogan らは，基準関数 (6) 式に基づいて，反復的に最適解を求める k-means 法を文書クラスタリングの問題に応用している (Kogan (2001)[48] および Dhillon ほか (2004)[21])。彼らのアルゴリズムは，基本的に，batch k-means 法（通常の k-means 法）と incremental k-means 法という 2 つの既存の方法を組み合わせたものであり，平均クラスタリング・アルゴリズム (means clustering algorithm) と名付けられている。

平方誤差の総和は，各文書を，その文書ベクトルが最もその重心に近いクラスタに割り当て直せば，小さくなることが期待できる。つまり，この再割り当てによって求め直されたクラスタを $\tilde{C}_k$ と表記すると ($k=1,\ldots,L$)，

$$\tilde{C}_k=\{d_i \mid k=\arg\min_{k'}\|\mathbf{d}_i-\mathbf{m}_{k'}\|\} \tag{8}$$

である（batch k-means 法）。

しかし，この手順を反復的に繰り返した場合，局所的な最適解に落ちてしまう可能性がある。これを防ぐには，ある 1 件の文書 d_i を別のクラスタに割り当て直してみたときに，その平方誤差の総和 J_e が最も減少するような，クラスタの分割を求めることが考えられる（incremental k-means 法）。この分割を $\{\hat{C}_k\}_{k=1}^{L}$ と表記する。

平均クラスタリング・アルゴリズムは，以上の 2 つの分割 $\{\tilde{C}_k\}_{k=1}^{L}$ と $\{\hat{C}_k\}_{k=1}^{L}$ とを交互に反復的に求めることによって，よりよいクラスタを得ようとする方法である。

平均クラスタリング・アルゴリズム

(1) 初期的なクラスタの分割 $\{C_k\}_{k=1}^{L}$ を生成する。また 2 つの閾値 θ_1 と θ_2 を設定する。

(2) 文書 $d_1, \ldots, d_N$ に対して，(8)式で定義される分割 $\{\tilde{C}_k\}_{k=1}^{L}$ を求める。もし，
$J_e(\{\tilde{C}_k\}_{k=1}^{L}) - J_e(\{C_k\}_{k=1}^{L}) < \theta_1$
ならば，分割を更新し，$\{C_k\}_{k=1}^{L} \leftarrow \{\tilde{C}_k\}_{k=1}^{L}$ とする。

(3) 文書 $d_1, \ldots, d_N$ に対して，分割 $\{\hat{C}_k\}_{k=1}^{L}$ を求める。もし，
$J_e(\{\hat{C}_k\}_{k=1}^{L}) - J_e(\{C_k\}_{k=1}^{L}) < \theta_2$
ならば，分割を更新し，$\{C_k\}_{k=1}^{L} \leftarrow \{\hat{C}_k\}_{k=1}^{L}$ とする。

(4) 終了条件が満たされれば，処理を終了する。そうでなければ，(2)に戻る。

4. SKWIC法

Frigui と Nasraoui (2004)[26] は，k-means法を拡張し，最適なクラスタリングとそのための最適な語の重みを同時に計算するアルゴリズム “simultaneous keyword identification and clustering of text document (SKWIC)” を提案した。この方法では，文書 d_i とクラスタ C_k の重心ベクトルとの非類似度を，

$$\hat{s}(\mathbf{d}_i, \mathbf{c}_k) = \sum_{j=1}^{M} \tilde{w}_{kj} \hat{w}_{ijk} \tag{9}$$

で定義する（つまり，この値が小さいほど，その文書とクラスタは類似していることになる）。ここで，

$$\hat{w}_{ijk} = \frac{1}{M} - x_{ij} m_{kj} \tag{10}$$

であり，この式中の m_{kj} は，重心ベクトル(7)式における j 番目の要素（語 t_j についての値）を意味する。SKWIC法では，これらの量を使って，次のような目的関数を定義し，これを最小にするような m_{kj} と $\tilde{w}_{kj}$ とを算出する。すなわち，

$$J_s = \sum_{k=1}^{L} \sum_{i:\, d_i \in C_k} \sum_{j=1}^{M} \tilde{w}_{kj} \hat{w}_{ijk} + \sum_{k=1}^{L} \left(\delta_k \sum_{j=1}^{M} \tilde{w}_{kj}^2 \right) \tag{11}$$

である。ただし，$\tilde{w}_{kj} \in [0, 1]$ $(k = 1, \ldots, L;\ j = 1, \ldots, M)$，かつ，

$$\sum_{j=1}^{M} \tilde{w}_{kj} = 1, \quad k = 1, \ldots, L$$

を条件として設定する。

この問題を Lagrange の未定係数法を使って解くと，

$$\tilde{w}_{kj} = \frac{1}{M} + \frac{1}{2\delta_k} \times \sum_{i:\, d_i \in C_k} \left[\left(\frac{1}{M} \sum_{j=1}^{M} \hat{w}_{ijk} \right) - \hat{w}_{ijk} \right] \tag{12}$$

を得る。なお，δ_k は，目的関数(11)式における第1項と第2項とのバランスを調整するためのパラメータで，実際には，クラスタリングの過程の中で反復的に値が算出される（後述）。

具体的な SKWIC 法の手順は以下のとおりである。

SKWIC法

(1) クラスタの個数 L を決め，L 件の文書を無作為抽出して，それらを各クラスタの重心とする（初期化）。また，$\tilde{w}_{kj} = 1/M$ のように初期設定する。

(2) $\hat{w}_{ijk}$ を式(10)に従って計算する $(k = 1, \ldots, L;\ j = 1, \ldots, M;\ i = 1, \ldots, N)$。

(3) 式(12)を使って，$\tilde{w}_{kj}$ を更新する。

(4) 式(9)を使って，$\hat{s}(\mathbf{d}_i, \mathbf{c}_k)$ を計算する $(k = 1, \ldots, L;\ i = 1, \ldots, N)$。

(5) $\hat{s}(\mathbf{d}_i, \mathbf{c}_k)$ に基づいて各文書 $d_1, \ldots, d_N$ を最も近いクラスタに再配分する。

(6) 新たに求められたクラスタに対して，重心ベクトル(7)式を計算する。ただし，$\tilde{w}_{kj} = 0$ ならば $m_{kj} = 0$ とする。

(7) もし再計算された重心ベクトルが前段階で求められたものと変わらなければ，クラスタリングの処理を終了する。もしそうでなければ，
$$\delta_k = K_\delta \frac{\sum_{i:\, d_i \in C_k} \sum_{j=1}^{M} \tilde{w}_{kj} \hat{w}_{ijk}}{\sum_{j=1}^{M} \tilde{w}_{kj}^2}$$
のように δ_k を再計算して（ここで K_δ はパラメータ），(2)に戻る。

以上の手法は，クラスタの個数 L を先験的に決め，そのうち，その重心に最も近いクラスタに文書を割り当てることから，基本的には k-means 法であるといえるが，特徴（すなわち語）の重み

を同時に最適化することに独創性がある。その重みを使えば，各クラスタを特徴づけるための「最適な」語の集合の特定が可能であり，この点，文書クラスタリングにとって興味深い方法といえる。なお，上記の手順では排他的なクラスタリングが得られるが，Frigui と Nasraoui (2004)[26] では，さらに，非排他的なアルゴリズムも提案されている。

5. Scatter/Gather のアルゴリズム

Scatter/Gather（Cutting ほか (1992)[19]）は，クラスタリングを活用することによって大規模な文書集合の通覧・検索を可能にするシステムである。このシステムは Xerox Palo Alto 研究所で開発され，その研究チームによる応用事例が，Hearst と Pedersen (1996)[34] など，いくつか報告されている。「Scatter」とは，1つまたは複数のトピックに関する文書集合をクラスタリングしてさらに詳細に分割することを意味し，それに対して「Gather」はその分割された集合から，関心のあるものをいくつか拾うことに相当する。この「Scatter」と「Gather」の2つを交互に繰り返していくことにより，大規模な文書集合が，次第に利用者の関心に密接な関連を持った，小規模な文書集合に絞られていくというしくみである。

そのクラスタリングの方法は基本的には k-means 法に基づく単一パス・アルゴリズムである。ただし，L 個のクラスタベクトルを設定するために，処理時間がより必要となる階層的クラスタ分析を使う点に特徴がある。具体的には，Buckshot と Fractionation という2つのアルゴリズムが提案されており（Cutting ほか (1992)[19]），これらは，ともに，

1. L 個の中心点（クラスタベクトル）を見つける
2. 各文書を1つの中心点に割り当てる（k-means 法）
3. 分割の精緻化 (refinement) を試みる

という手順で処理を進めていく。標準的な階層的クラスタ分析が適用されるのは，このうちの第1段階である。なお，ここでは，階層的クラスタ分析の計算量は，n 件の文書を処理する場合に $O(n^2)$ であると考えておく。

Buckshot では，文書集合全体から，$\sqrt{LN}$ 件の文書を無作為抽出して，それに対して階層的クラスタ分析を適用する。その結果は当然，抽出に依存して変化してしまうが，処理が速いという利点がある（計算量は $O(LN)$ になる）。Scatter/Gather における階層的クラスタリングの目的はあくまでクラスタベクトルの設定なので，無作為抽出された標本が母集団のようすを正しく反映していれば十分なわけである。

一方，Fractionation では，最初に文書集合全体を，それぞれ m 件の文書から成る小さな集合 (bukets) に機械的に分割して，それぞれに対して階層的クラスタ分析を適用する。そして，その結果得られた各クラスタを単一の文書とみなして，同一の手順を繰り返していく。Buckshot よりも処理時間を要する反面，一応すべての文書を対象にして分析するので，よりよい結果が得られると期待できる。

まず，文書集合全体 D を N/m 個の排他的部分集合に機械的に分割する。それぞれに対して，ρm 個のクラスタが得られるように，階層的クラスタ分析を適用する ($0<\rho<1$)。この結果，全部で $N/m \times \rho m = \rho N$ 個のクラスタ，すなわち，$C_{k,h}$ ($k=1, \ldots, N/m$; $h=1, \ldots, \rho m$) を得ることができる。ここで，$C_{k,h}$ は k 番目の部分集合における h 番目のクラスタとする。これらの $C_{k,h}$ を単一の文書と見なし，合計 ρN 件の文書に対して同様の手順を再び適用すれば（すなわち ρN 件の「文書 $C_{k,h}$」をいったん1つの集合にまとめてから，再度 m 件ずつに分割し，それぞれから ρm 個のクラスタを抽出），$\rho^2 N$ 個のクラスタが得られる。この処理を r 回繰り返して，$\rho^r N \leq L$ となった時点で手順を終了し，最後に残ったクラスタのベクトルを採用すればよい。計算量は，$O(m^2 \times N/m + m^2 \times \rho N/m + \cdots) = O(Nm(1+\rho+\rho^2+\cdots+\rho^r)) = O(Nm)$ である。

Scatter/Gather では，文書間の類似度および

文書とクラスタとの間の類似度を次のように計算する。まず，文書ベクトルの要素を$w_{ij}=\sqrt{x_{ij}}$で定義し，文書間の類似度は，一般的な余弦係数(2)式で求める。一方，クラスタC_kのベクトルは

$$\mathbf{c}_k=\sum_{i:d_i\in C_k}\mathbf{d}_i/\|\mathbf{d}_i\|$$

すなわち，

$$\tilde{w}_{kj}=\sum_{i:d_i\in C_k}\frac{w_{ij}}{\sqrt{\sum_{j=1}^{M}w_{ij}^2}}$$

とする。これがScatter/Gatherにおける具体的な「中心点」となる（この中心点は重心ベクトル(7)式とは異なることに注意)。文書とクラスタ間の類似度$s(\mathbf{d}_i, \mathbf{c}_k)$は同様に式(2)を使う。

L個の中心点が決まれば，あとはN件の文書を順に各中心点に割り当てていく。これは単純にL個のクラスタのベクトルと各文書ベクトルとの類似度を計算し，その値の最も大きなクラスタにその文書を配分する作業である。最後に，その結果として得られたL個のクラスタを精緻化する。この作業は，(i)分化(split)と(ii)結合(join)から成る。

「分化」ではその時点で得られている各クラスタに対してBuckshotを適用する（1つのクラスタに対して$L=2$でBuckshotを適用)。一方，「結合」は，2つのクラスタにおける「主要語」が共通している場合に，両者を併合する操作である。ここで，「主要語」とは，クラスタのベクトル中の重みによって語を並べたときの上位y個を指す。具体的には，2つのクラスタ間で共通する「主要語」の数を調べ，その共通語数がある閾値を超えていれば，両者を併合する。

以上がScatter/Gatherでのクラスタリングのアルゴリズムの概要である。このシステムでは，実際に，利用者に対して高速で応答しなければならない場合にはBuckshotを使い，そうでないとき（例えば，利用者の問合せ以前にクラスタリングを実行する場合など）には，Fractionationを用いている。

なお，階層的クラスタ分析の方法としては，凝集型の群平均法が利用されている。通常，この方法によって樹形図が出力されるが，この場合には，それは関係なく，群平均法の計算においてL個のクラスタが得られた時点で分析を打ち切る。そしてこれらのクラスタのベクトルを上述の方法で計算し，それらを中心点として利用する。

もし，文書集合（データベース）が前もって，階層的に構造化されていれば，Scatter/Gatherシステムを実際に利用者が使う際に，凝集型の階層的クラスタ分析を明示的に実行する必要はなく，その分，応答速度が短くなる。ただし，そのように前もって構成された階層構造が必ずしも利用者の情報要求に合うとは限らず，不要な文書もその結果に含まれてしまう可能性がある。この問題を解決するために，任意の文書集合に対して，既存の階層構造から必要なクラスタを抽出し，それに基づいて，クラスタリングを高速で（文書集合の大きさに対して線型の時間量で）実行するアルゴリズムも考案されている（Silverstein とPedersen (1997)[76])。

なお，江口ら(1999)[23]は，検索システムが利用者に返す検索結果集合をクラスタ化するためにFractionationを利用しているが，そこでは階層的クラスタ分析法として単連結法が使われている。また，この研究では，利用者による検索語の重みを文書ベクトルの重みに付加することによって，利用者の興味を反映した，適応型文書クラスタリング(adaptive document clustering)を実現しようとする試みがなされている。

さらに，第1段階で階層的クラスタ分析法である群平均法を用い，その結果得られたクラスタに対して，第2段階で各文書を割り当てる方法は，Stanford大学のDigital Libraries Testbedの構成要素であるSONIA (Service for Organizing Networked Information Autonomously) (Sahammi ほか(1998)[72])でも採用されている。もっともSONIAでの文書ベクトルや類似度の定義はScatter/Gatherのそれとは異なっている。

6. C^3MおよびC^2ICMアルゴリズム

k-means法では，通常，クラスタの重心ベクトルとの類似度（または非類似度）に基づいてクラスタリングが実行される。それに対して，クラス

タを1つの分類対象によって代表させ，それとの類似度（または非類似度）によって，対象を分類する方法もあり，一般に，k-medoid法と呼ばれる。CanとOzkarahan(1984,1985,1987,1990)[11)~14)]による C^3M (cover-coffecient-based concept clustering methodology)は，文書が他の文書に「覆われる」程度という独自の要因を導入することにより，クラスタ個数の自動決定を可能にした，k-medoid法の変種である。

この方法では，クラスタリングの対象になる N 件の文書と，それに含まれる M 個の語について，文書 d_i 中に語 t_j が出現するかどうかを示す変数 b_{ij} を考える。すなわち，出現すれば $b_{ij}=1$，出現しなければ $b_{ij}=0$ とする。そして，次の量を定義する。

$$\phi_{ij}=\frac{b_{ij}}{\sum_{j=1}^{M} b_{ij}},\quad i=1,\ldots,N;\ j=1,\ldots,M \tag{13}$$

$$\tilde{\phi}_{ij}=\frac{b_{ij}}{\sum_{i=1}^{N} b_{ij}},\quad i=1,\ldots,N;\ j=1,\ldots,M \tag{14}$$

ϕ_{ij} は文書 d_i における語 t_j の重要性を示している。例えば，ある文書に10個の語が含まれるならば，それらの語の ϕ_{ij} の値は0.1である。逆に，$\tilde{\phi}_{ij}$ は語 t_j に対する文書 d_i の重要性を表す。もしある語が100件の文書に出現するならば，$\tilde{\phi}_{ij}$ の値は0.01である。

次に，これらの量に基づいて，

$$\delta_{ih}=\sum_{j=1}^{M}\phi_{ij}\tilde{\phi}_{hj} \tag{15}$$

を定義する $(i,h=1,\ldots,N)$。すなわち，δ_{ih} は，文書 d_i 中に出現する各語に対して，文書 d_h の重要性を求め，ϕ_{ij} で重み付けして合計したものである。もし，文書 d_i と文書 d_h が語を共有しなければ，$\delta_{ih}=0$ となる $(i\neq h)$。さらに，もし文書 d_i が他のどの文書とも語を共有しなければ，定義より，$\delta_{ii}=1$ かつ $\delta_{ih}=0$ $(i\neq h)$ である。逆に，文書 d_i が他の文書と語を共有すればするほど，δ_{ii} の値は小さくなる。もし，文書 d_i に含まれる語がいずれも他のすべての文書中に出現するならば，$\delta_{ii}=1/N$ となり，これは δ_{ii} の最小値である。以上のことから，δ_{ii} は文書 d_i の「独自性(uniqueness)」を表すものと解釈できる。

一方，$i\neq h$ の場合の δ_{ih} は，文書 d_i が文書 d_h によって「覆われる(covered)」程度として解釈される。例えば，文書 d_1 と文書 d_2 が2つの語 t_1 と t_2 を共有し，t_1 も t_2 も10件の文書に出現していると仮定する。ここでもし文書 d_1 がこれらの語を含めて10個の語を持つのに対して，文書 d_2 がこの2つの語のみを含むとすれば，$\delta_{12}=0.02$，$\delta_{21}=0.1$ である。文書 d_2 が持つ語はいずれも文書 d_1 に含まれているという点で文書 d_2 は文書 d_1 に覆われているといえる（逆に，文書 d_1 が持つ語は文書 d_2 に含まれないものが多く，d_1 はそれほど d_2 には覆われていない)。また，もし t_1 と t_2 が5件の文書のみに出現すると仮定すると，$\delta_{21}=0.2$ となり，「覆われる」程度は倍増する。この例が示すように，一般には $\delta_{ih}\neq\delta_{hi}$ $(i\neq h)$ であり，さらに，式(13)と(14)を使えば，

$$\sum_{h=1}^{N}\delta_{ih}=\sum_{h=1}^{N}\sum_{j=1}^{M}\phi_{ij}\tilde{\phi}_{hj}=\sum_{j=1}^{M}\left(\phi_{ij}\sum_{h=1}^{N}\tilde{\phi}_{hj}\right)$$

$$=\sum_{j=1}^{M}(\phi_{ij}\times 1)=1,\quad i=1,\ldots,N$$

を得る。

もしすべての文書が他の文書とまったく語を共有しなければ，すべての文書の「独自性」δ_{ii} は1になる。これはこの文書集合の内部が非常に分離していることを意味している。この場合には，当然，最適なクラスタの個数は，$L=N=\sum_{i=1}^{N}\delta_{ii}$ にならざるをえない。一方，すでに述べたように，ある文書 d_i 中の語がすべての他の文書に含まれていれば，$\delta_{ii}=1/N$ であるが，もしすべての文書で含まれる語がまったく同一ならば，$\sum_{i=1}^{N}\delta_{ii}=\sum_{i=1}^{N}1/N=1$ となる。このような状況では，当然，すべての文書を1つのクラスタにまとめるべきであるから，この計算は直観と一致する。これらのことから，CanとOzkarahanは，$\sum_{i=1}^{N}\delta_{ii}$ をクラスタの最適個数の予測値として捉え，

$$L=\sum_{i=1}^{N}\delta_{ii} \tag{16}$$

とおいた。

C^3M の具体的な手順は以下のとおりである。

C³M に基づくアルゴリズム

(1) "cluster seed power" に基づいてクラスタの種子点を L 個選び，種子点の集合 D_s をつくる。$i=0$ と設定する。

(2) $i \leftarrow i+1$ とし，文書 d_i を読む。もし N 件の文書すべてを読み終わっていたならば，クラスタリングを終了する。そうでなければ，(3) に進む。

(3) もし d_i が種子点ならば ($d_i \in D_s$)，(2) に戻る。そうでなければ (4) に進む。

(4) d_i を最も覆っている種子点のクラスタに d_i を割り当てる。つまり，i を固定したときの最大の δ_{ih} を持つ d_h に文書 d_i を割り当てる ($d_h \in D_s$)。もし，そのような種子点が複数ある場合には，最も大きな "cluster seed power" を持つ d_h に割り当てる。そののち (2) に戻る。

ここで，"cluster seed power" とは，文書 d_i に対して，

$$p_i=\delta_{ii}(1-\delta_{ii})\sum_{j=1}^{M} b_{ij} \qquad (17)$$

によって定義されるもので ($i=1,\ldots,N$)，基本的には，「独自性」が中程度の文書の値が大きくなるように，$\delta_{ii}(1-\delta_{ii})$ を使い，さらに，文書に含まれる語数 $\sum_{j=1}^{M} b_{ij}$ を乗じている。この p_i は L 個の種子点を選ぶのにも使われる。つまり，最初にすべての文書に対して式 (17) を計算し，その上位 L 件を種子点とする。

なお，上記の (4) の段階で，特にクラスタを 1 つに絞らないことにすれば，重複を許すクラスタリングとなる。また，どの種子点にも割り当てられない文書（すべての $d_h \in D_s$ に対して $\delta_{ih}=0$ となる文書 d_i）は除外しておき，最後にそれらを 1 つにまとめて「その他」クラスタとする。

Can (1993)[9] では，C³M によって構成されたクラスタ集合に対して，新たな文書の追加または文書の削除を実行するための，"cover-coefficient-based incremental clustering methodology (C²ICM)" が提案されている（この方法を実際の OPAC システムに対して適用した例については Can ほか (1995)[10] を参照)。また，Ishikawa ら (2001)[37] は，この C²ICM アルゴリズムを拡張し，文書の価値が時間的に減衰していくことを組み込んだ手法である，F²ICM (forgetting-factor-based incremental clustering method) を考案している。この方法では，古い文書よりも新しい文書のほうがクラスタリングの結果に貢献するように，各文書に対して，指数関数的に減少する重み $w_i^t=\lambda^{t-T_i}$ を設定し ($i=1,\ldots,N$)，これを種々の計算に活用する。ここで，λ は定数（すなわち減衰の係数で，$0<\lambda<1$)，t は現在時間，T_i は文書 d_i が入手された時間である。

7. Leader-follower アルゴリズムの応用

単純な k-means 法では，クラスタ個数およびそれらのベクトルを先験的に与えなければならないが，それを回避するための単一パス・アルゴリズムとして，いわゆる leader-follower 法（Duda ほか (2001)[22]）がある。この方法では，クラスタ個数の代わりに，文書とクラスタとの類似度の閾値を前もって設定しておく。分類対象を順に処理していく中で，この閾値に基づいて自動的にクラスタが形成されていくというしくみである。この leader-follower アルゴリズムの最も単純な場合が，最近，岸田 (2003)[46] によって文書クラスタリングに応用されている。

その具体的な手順を以下に示す (Rasmussen (1992)[70])。

単純な leader-follower 法

(1) 閾値 θ_s を設定する。最初の文書 d_1 を読み，最初のクラスタとする ($\mathbf{c}_1 \leftarrow \mathbf{d}_1$)。この結果，クラスタの個数 L は 1 となる ($L \leftarrow 1$)。i を初期化する ($i \leftarrow 1$)。

(2) $i \leftarrow i+1$ とし，文書 d_i を読む。もし N 件の文書すべてを読み終わっていたならば，クラスタリングを終了する。そうでなければ，(3) に進む。

(3) 文書ベクトル $\mathbf{d}_i$ と，その時点で存在するすべてのクラスタベクトル $\mathbf{c}_k$ との類似度 $s(\mathbf{d}_i, \mathbf{c}_k)$ を計算する ($k=1,\ldots,L$)。もしそれが閾値を超えれば ($s(\mathbf{d}_i, \mathbf{c}_k)>\theta_s$)，$d_i$ をクラスタ C_k に加え，

ベクトル $\mathbf{c}_k$ を更新する（例えば，$\mathbf{c}_k \leftarrow \mathbf{c}_k + \eta \mathbf{d}_i$。ここで，$\eta$ は「学習率」で，0.0〜1.0 の実数とする）。もし文書がどのクラスタにも加わらなければ，その文書を新しいクラスタとする。すなわち，$L \leftarrow L+1$, $\mathbf{c}_L \leftarrow \mathbf{d}_i$。そののち段階 (2) に戻る。

なお，この手順では，クラスタの重複が許されており，1 件の文書が複数のクラスタに属することが可能となっている。もし，段階 (3) で，最も類似度の高いクラスタを 1 つだけ選んで，文書を追加することにすれば，排他的なクラスタリングとなる。

より具体的には，岸田 (2003)[46] では，情報検索における標準的なベクトル空間モデル（Buckley ほか (1994)[8]）に従って，

$$w_{ij} = (\log x_{ij} + 1) \log \frac{N}{n_j} \qquad (18)$$

と定義し，ベクトル $\mathbf{d}_i$ を求めている（ただし，$x_{ij}=0$ ならば $w_{ij}=0$ とする）。

一方，クラスタのベクトル $\mathbf{c}_k$ の重みについては，

$$\tilde{w}_{kj} = 1 + \log \sum_{i:\, d_i \in C_k} x_{ij}$$

としている（岸田 (2003)[46]）。この計算方法は，各クラスタをそれに含まれる文書を単純に合併した「巨大な文書」と考えていることになる。このベクトルと文書ベクトルと間の類似度の計算には，余弦係数 (2) 式が使われる[46]。以上の設定方法は，文書 d_i を「検索質問」，クラスタ C_k を「文書」として読み替えることによって，文書クラスタリングの問題を情報検索の状況に対応付けたものである。

この方法ならば，k-means 法とは異なり，あらかじめクラスタの個数を限定する必要はなく，閾値 θ_s に従い，文書集合 D の状況に応じて「自然」な個数のクラスタが求められることになる。

しかし，θ_s の設定という課題が依然として残るし，また，L が大きくなると，計算量の問題も生じる。上記の単純な leader-follower 法の計算量は，基本的には，反復計算しない場合（すなわち $r=1$ の場合）の k-means 法と同じで，特に工夫しなければ $O(NLM)$ である。ここで，もし計算の途中で L が大きくなり，N と同程度になってしまったら（例えば $L \cong N/2$），その計算量は $O(N^2M)$ に近づくことになる。また，L 個のクラスタベクトルの格納に必要な主記憶装置の領域もまた大きなものとなり，深刻な問題を引き起こす可能性がある。

8. Crouch のアルゴリズム

Crouch (1975)[17] による文書クラスタリングの方法は，一種の leader-follower アルゴリズムであるが，文書ファイルを 2 度走査して，最初に文書集合中に内在するクラスタを識別してから，次にそれらに文書を割り当てるという 2 段階の処理をおこなう点に特徴がある。すなわち，

(1) クラスタの設定（categorization の段階）
(2) クラスタへの文書の割り当て（classification の段階）

の 2 段階である。

段階 (1) では，ベクトル $\mathbf{c}_k$ を更新する際に，各語の変動係数を計算し，その値に基づいて，各クラスタを表現する語の集合 (core terms) を選別する。そして，選ばれた core terms の重みの平均に基づいて，ベクトル $\mathbf{c}_k$ を更新する。結果的に，処理が終了した時点で残った $\mathbf{c}_k$ ($k=1,\ldots,L$) が，クラスタを表現するベクトルとして採用されることになる。core terms を選別することによって，クラスタベクトルを格納する領域が少なくて済み，また，類似度 s の計算時間も短くなる。

なお，第 2 段階でのクラスタへの文書の割り当てにおいては，クラスタの重複が許され，1 件の文書が複数のクラスタに属することが可能である。また，類似度 s については，重複係数 (overlap coefficient),

$$s(\mathbf{d}_i, \mathbf{c}_k) = \frac{\sum_{j=1}^{M} \min(w_{ij}, \tilde{w}_{kj})}{\min(\sum_{j=1}^{M} w_{ij}, \sum_{j=1}^{M} \tilde{w}_{kj})} \qquad (19)$$

が使用されている。

9. CBC アルゴリズム

k-means 法の問題点の 1 つは，クラスタの重

心を再計算するときに，そのクラスタに部分的にしか属さないような分類対象（すなわち文書）が不当な影響を与えてしまうことである。その結果，各クラスタ内の結合がゆるくなり，1つのクラスタの内容が多様になってしまう可能性がある。この問題を解決するために，Pantel と Lin (2002)[68] は，Clustering By Committees (CBC) アルゴリズムを考案した。

CBC アルゴリズムは次の3つの段階（フェーズ）から構成される。

1. 段階Ⅰ：文書ごとに，類似度の高い20文書を求める。
2. 段階Ⅱ：段階Ⅰで計算された類似度から，committee のリストを求める。
3. 段階Ⅲ：各文書を最も類似した committee に割り当てる（この結果，各文書がクラスタに割り当てられることになる）。

ここで “committee” とは，各クラスタの核となるいわば「代表者（の集まり）」のようなもので，実際には，段階Ⅱにおいて，次の手順で求められる。

CBC アルゴリズムの段階Ⅱ

(1) 2つの閾値 θ_1 と θ_2 を設定し，3種類のリスト L_C, L_K, L_R を用意する。このうち，L_K を comittee のリストとする。また，処理が未完である文書のリストを E として，最初にすべての文書を E に含める。

(2) リスト L_C を空にし，E に含まれる文書ごとに次の処理をおこなう。

(2-1) その文書と類似した20文書に対して，群平均クラスタリングを適用する。

(2-2) その結果生成されたクラスタ C ごとに，得点 $|C| \times \mathrm{avgsim}(C)$ を計算する。ここで $|C|$ はそのクラスタに含まれる文書数，$\mathrm{avgsim}(C)$ は C 中の文書の組ごとの類似度の平均とする。

(2-3) 得点の最も高いクラスタを，リスト L_C に追加登録する。

(3) リスト L_C を得点の降順に並び替える。

(4) リスト L_K を空にする（初期化）。

(5) リスト L_C 中の上位から順に，各 $C(\in L_C)$ について次の処理をおこなう。

(5-1) クラスタ C の重心と，その時点でリスト L_K に含まれているすべての committee の重心との類似度を計算する。

(5-2) それらの類似度がすべて閾値 θ_1 を下回っている場合，C を comittee のリスト L_K に追加する。

(6) もしリスト L_K が空ならば処理を終了する。

(7) 再び，E に含まれる文書ごとに以下の処理をおこなう。

(7-1) もしその文書とすべての comittee との類似度が閾値 θ_2 を下回るならば，その文書をリスト L_R に追加する。

(8) もしリスト L_R が空ならば処理を終了する。そうでなければ，L_K の内容を保存し，L_R を E として，(2) に戻る（反復処理）。

この段階Ⅱの出力結果は，各反復計算における段階 (8) で保存されたリスト L_K の和集合である。段階Ⅲでは，この和集合に含まれる各 comittee の重心に対する各文書の類似度を計算し，各文書を，その類似度の最も高い comittee に割り当てることにより，最終的なクラスタを求める。

この手順から明らかなように，CBC アルゴリズムでは，最初に，各クラスタの核となるべき comittee を算出しておいて，それを中心とするクラスタを形成する。すなわち，ちょうど段階Ⅲの処理が，k-means 法にほぼ相当しており，その前段階のⅠ，Ⅱにおいて，各クラスタの核を committee として求めていることになる。これによって，「周辺的な」文書がクラスタの重心の計算に不当な影響を及ぼすことを防いでいるわけである。

10. TDT における単一パス・クラスタリング

Topic Detection and Tracking (TDT) は，何らかのイベント（できごと，事件など）についての情報を，ニュースの文書などから成るコーパスから自動識別する試みである。近年では，米国の

DARPA を中心に，TDT の研究が活発に行われている[64]。このうち，「Tracking」は，目標となるイベントに関するトピックがあらかじめ与えられ，それをいわば追跡するのに対して，「Detection」では，トピックは先験的には与えられず，その条件の下で文書集合をクラスタリングしなければならない。したがって，「Tracking」は教師付きの分類，「Detection」は教師なしのクラスタリングとみなすこともできる。

このような Detection のために，最近，単一パス・アルゴリズムが使用されている（Hatzivassiloglou ほか (2000)[32] や Franz ほか (2001)[25]）。なお，Hatzivassiloglou ら (2000)[32] は，単一パス・アルゴリズムのほか，伝統的な階層的クラスタ分析法（単連結法，完全連結法，群平均法）を TDT のコーパスに適用し，比較を試みている。

Franz ら (2001)[25] では，クラスタと文書間の類似度の計算に，確率型検索モデルである Okapi 方式が使用されている。まず，クラスタ C_k における文書 d_i と文書 d_h との類似度 $s(\mathbf{d}_i, \mathbf{d}_h; C_k)$ を次のように定義する。

$$s(\mathbf{d}_i, \mathbf{d}_h; C_k)=\sum_{j=1}^{M} w_{ij}w_{hj}\times idf(t_j, C_k)$$

ここで，w_{ij} は，文書 d_i における語 t_j の出現頻度を，$\sum_{j=1}^{M} w_{ij}=1$ となるように正規化したものである（w_{hj} も同様）。また，

$$idf(t_j, C_k)=\log\frac{N-n_j+0.5}{n_j+0.5}+\lambda\frac{2n_{j|k}}{n_j+\tilde{n}_k}$$

であり，$n_{j|k}$ はクラスタ C_k に属する文書の中で，語 t_j を含んでいる文書の数，λ は定数である。これらの式を使って，文書 d_i とクラスタ C_k の類似度は，最終的に，C_k に含まれる各文書との類似度の平均，

$$s(d_i, C_k)=\frac{1}{\tilde{n}_k}\sum_{h:\, d_h\in C_k} s(\mathbf{d}_i, \mathbf{d}_h; C_k)$$

で計算される。

また，Papka と Allan (2000)[69] では，TDT におけるオンライン・クラスタリング (on-line clustering) の方法が詳しく述べられている。これは，前もって文書集合がクラスタリングされているところに，新たな文書が 1 件追加された場合に，この文書を既存のクラスタに割り当てる（または，いずれにも割り当てられない場合には，新たなクラスタとして独立させる）ものである。この文献では，クラスタリングの方法としては階層的クラスタ分析が使用されているが，文書の到着順に 1 件ずつクラスタに割り当てていくという点で，ややかたちを変えた単一パス・アルゴリズムとして捉えることもできる。

Franz ら (2001)[25] と同様に，Papka と Allan (2000)[69] も，文書間の類似度の計算に，確率モデルを利用している。ただし，Okapi 方式ではなく，INQUERY における計算式が使用されている。既存の文書を d_h，新たに到着した文書を d_i とすると，それらのベクトル間の類似度は，INQUERY の sum 演算子によって，

$$s(\mathbf{d}_i, \mathbf{d}_h; d_h)=\frac{\sum_{j=1}^{M} w_{hj}w_{ij}}{\sum_{j=1}^{M} w_{hj}}$$

で求められる（この類似度は非対称である）。ここで，

$$w_{ij}=0.4+0.6\times\frac{x_{ij}}{x_{ij}+0.5+1.5l_i/\bar{l}}\times\frac{\log((N+0.5)/n_j)}{\log(N+1)}$$

であり，$\bar{l}$ は文書集合における文書長の平均，すなわち，$\bar{l}=N^{-1}\sum_{i=1}^{N} l_i$ を意味する（w_{hj} も同様に定義される）。

11. 自己組織化マップの応用

Kohonen による自己組織化マップ (self-organizing map: SOM)（Kohonen (1995)[49] や van Hulle (2000)[83] を参照）を応用して文書をグループ化する試みが 1990 年代前半にいくつかなされ (Lin ほか (1991)[57] や Chen ほか (1994)[15])，その後，Kohonen らのグループによって，SOM による文書クラスタリングの研究が精力的に展開された。Kohonen らのグループによるシステムは WEBSOM と呼ばれ，1990 年代後半から継続的に研究発表がなされている（初期のものとしては Honkela ほか (1996)[36] などがある）。WEBSOM 以外にも，文書クラスタリングに SOM を応用しようとする試みは数多い（Lin

(1997)[56]，Orwig ほか (1997)[67]，Roussinov と Chen (2001)[71]，Bote ほか (2002)[7] など)。

SOM は，これまで見てきた k-means 法や leader-follower 法とは異なる原理によって，非階層型の分割を与える方法で，基本的には，1 対の入力空間と出力層から構成される，一種のニューラルネットワークである。そのアルゴリズムは，(1) 競合段階と (2) 協調段階から成り，競合段階では，入力ベクトルに最も近い出力ノードが 1 つのみ特定される。

文書クラスタリングの場合には，入力ベクトルは文書ベクトル $\mathbf{d}_i$ そのものである。一方，SOM を例えば横 10，縦 20 のセルから成る矩形として出力しようとすれば，出力層は $10\times20=200$ 個のノードから構成されることになる。ここでは，出力ノードの総数を H と表記し，各出力ノードに対応づけられたベクトルを $\mathbf{v}_h\ (h=1,\ldots,H)$ とかく。

入力ベクトル $\mathbf{d}_i$ と第 h 番目の出力ノードのベクトル $\mathbf{v}_h$ との「近さ」をユークリッド距離の 2 乗で測ることにすれば (Orwig ほか (1997)[67])，各文書 d_i に対して，

$$h^*=\arg\min_h \|\mathbf{d}_i-\mathbf{v}_h\|^2$$
$$=\arg\min_h \sum_{j=1}^{M}(w_{ij}-v_{hj})^2 \qquad (20)$$

を求め，h^* 番目のノードを，入力ベクトル $\mathbf{d}_i$ に最も近いという意味での「勝者」として定義できる (ここで，$\mathbf{v}_h=(v_{h1},\ldots,v_{hM})^T$)。これが競合段階である。

協調段階では，「勝者」ノードとその周辺のノードの重みを調整する。例えば，「勝者」ノードとその近傍の添字の集合を $\mathcal{N}_{h^*}$ と定義し，$j=1,\ldots,M$ について，反復計算によって，

$$v_{hj}^{(r+1)}=\begin{cases} v_{hj}^{(r)}+\eta^{(r)}(w_{ij}-v_{hj}^{(r)}), & h\in\mathcal{N}_{h^*}\text{ の場合} \\ v_{hj}^{(r)}, & \text{それ以外}\end{cases} \qquad (21)$$

を求める (Orwig ほか (1997)[67])。ここで，$v_{hj}^{(r)}$ における r は反復回数を示している。また，$\eta^{(r)}$ は学習率であり $(0\le\eta^{(r)}\le1)$，r の増加に伴って減少していく。

SOM による文書クラスタリングの具体的な手順の例は次のようになる (Orwig ほか (1997)[67])。

SOM による文書クラスタリングの例

(1) 乱数を発生させ，出力ベクトル $\mathbf{v}_h$ を初期化する $(h=1,\ldots,H)$。

(2) 文書 $d_1,\ldots,d_N$ を順に入力し，(20) 式と (21) 式によって，出力ノードを調整する。この過程を G 回反復する。

(3) 文書 $d_1,\ldots,d_N$ を順に入力し，最も距離の近い出力ノードにその文書を割り当てていく。

(4) 語 $t_1,\ldots,t_M$ を同様に出力ノードに割り当てていく (1 つの語のベクトルを，その語のみを 1，他の語は 0 とした M 次元ベクトルとして考えれば，上の段階 (3) と同じ手順で語を出力ノードに割り当てることができる)。これによって，SOM 上の各ノードにラベルを付与できる。

実際には，類似したノードをまとめて区域 (region) とし，それぞれの区域をクラスタとみなせばよい。SOM による文書クラスタリングの最大の長所は，その結果を 2 次元平面上に美しく表示できる点にある。地図全体は長方形であり，そのセルを区域に分割するので，区域ごとに色分けすれば，非常に見やすい地図となる。また，上記の段階 (4) で割り当てた語をラベルとして表示することもできる (なお，各区域に割り当てられた文書中の語の出現頻度に基づいてラベルを決める方法も考案されている (Lagus と Kaski (1999)[54])。一般に，各クラスタに対するラベル付けは重要な問題であり，SOM 以外の場合にも多くの研究があるが，本稿では，これに関して，SOM 以外については特に言及しない)。その反面，出力ノードの学習のためにはかなりの数の反復が必要であり，大規模なデータに対しては多くの計算量が必要になる。

Kohonen らによる WEBSOM では，大規模な文書集合に対して SOM を適用できるように，いくつかの工夫が加えられている。その結果，最近では，600 万件を超える特許の抄録に対する SOM の作成に成功している (Kohonen ほか

(2000)[50])。WEBSOMでは，出力ノードのベクトルの修正に，式(21)ではなく，

$$v_{hj}^{(r+1)}=v_{hj}^{(r)}+f_{c(\mathbf{d}_i),h}(r)(w_{ij}-v_{hj}^{(r)}) \qquad (22)$$

を使う。ここで，$c(\mathbf{d}_i)$は$\mathbf{d}_i$に対する勝者ベクトルの添字を返す関数であり，また，

$$f_{c(\mathbf{d}_i),h}(r)=\alpha(r)\exp\left(-\frac{\|\mathbf{t}_h-\mathbf{t}_{c(\mathbf{d}_i)}\|^2}{2\sigma^2(r)}\right)$$

である。ここで，$\mathbf{t}_h$は出力領域中での位置を示す2次元ベクトルである。この関数によって，出力領域中の勝者ノードとの距離が離れるにつれて，次第に各ノードの重みの修正幅が小さくなる（したがって，近傍集合$\mathcal{N}_{h^*}$を明示的に規定する必要はない）。なお，$\alpha(r)$はr回目の反復における学習率，$\sigma^2(r)$は，関数$f_{c(\mathbf{d}_i),h}(r)$の拡がりを調整するパラメータである。

WEBSOMでは，式(22)に投入される文書ベクトル$\mathbf{d}_i$に対する工夫もなされている。例えば，各文書ベクトルの次元数を減らすために，$m\times M$の行列$\mathbf{B}$を使って（ただし，$m\ll M$），

$$\tilde{\mathbf{d}}_i=\mathbf{B}\mathbf{d}_i \qquad (23)$$

のように文書ベクトルを変換する。この結果，新しい文書ベクトル$\tilde{\mathbf{d}}_i$では次元数がかなり減るので，計算量が少なくてすむ。変換行列$\mathbf{B}$を求めるには，LSIを使うこともできるが（本章C節参照），Kohonenらはこの行列を無作為に生成している（Kohonenほか(2000)[50]）。具体的には，各列の要素が正規分布に従い，なおかつ長さが1になるように，$\mathbf{B}$を計算する。この方法は，LSIよりも簡便であり，さらに文書間の類似度の計算という目的からは十分という結果が報告されている(Kaski(1998)[43])。なお，この方法を使用した場合のラベル付け（地図中の各区域を表すキーワードの抽出方法）は，AzcarragaとYap(2001)[4]によって議論されている。

さらには，複合語の処理方法についても独自の工夫が研究されており(Kaski(1999)[44])，また，SOMをクラスタ型の情報検索に応用する試みなどもある(Lagus(2000)[53])。

12. 遺伝的アルゴリズムの応用

Jonesら(1995)[41]によるクラスタリング手法は，遺伝的アルゴリズムを使って，ある基準に基づく「最適な」クラスタリングを近似的に求めようとするものである。

まず，文書ベクトルは2値として，文書d_iに語t_jが含まれれば$w_{ij}=1$，そうでなければ$w_{ij}=0$とする。次に，ある1つのクラスタのベクトルを，そのクラスタ中の一定数以上の文書に出現する語によって定義する。すなわち，Willettによるアルゴリズムで説明した式(4)を使う。そのうえで，クラスタに含まれる各文書のベクトルとクラスタベクトルとの類似度を，

$$s(\mathbf{d}_i,\mathbf{c}_k)=\frac{\sum_{j=1}^{M}w_{ij}\tilde{w}_{kj}}{\sum_{j=1}^{M}w_{ij}+\sum_{j=1}^{M}\tilde{w}_{kj}-\sum_{j=1}^{M}w_{ij}\tilde{w}_{kj}} \qquad (24)$$

で定義する（ここまでの説明から明らかなようにベクトルは両者とも2値である）。

クラスタが適したものであるかどうかは，そのクラスタ内での式(24)の平均，

$$J_g=\frac{1}{\tilde{n}_k}\sum_{i:\,d_i\in C_k}s(\mathbf{d}_i,\mathbf{c}_k)$$

によって測定される。そして，これをクラスタリングの成功の程度を測定する基準として使い，遺伝的アルゴリズムを適用する。この場合の染色体はNの長さを持ち，各文書がそのクラスタに属すれば1，そうでなければ0とする。あとは，ほぼ標準的な遺伝的アルゴリズムを使って，最適な分割（クラスタリング）を近似的に求めることになる。この際に，パラメータとして，上記のτのほかに，遺伝子の数，交配率などを与える必要がある。

13. Lightweightアルゴリズム

このアルゴリズム（Weissほか(2000)[90]）では，CBCアルゴリズムと同様に，最初に，文書$d_1,\ldots,d_N$それぞれに対して，最も類似したn件の文書（すなわち類似度の高い順にn件の文書）から成る集合を作成する。文書d_iに対するこの文書集合を$\tilde{D}_i$とし，

$$\tilde{D}_i=\{d_{i(1)},d_{i(2)},\ldots,d_{i(n)}\}$$

と表記する($i=1,\ldots,N$)。また，文書d_iに対し

て，それが属するクラスタの番号を返す関数を $g(d_i)$ と定義する。もし d_i がどのクラスタにも属さない場合には，この関数は 0 を返すものとする。

Lightweight アルゴリズム

(1) 閾値 θ を決め，$i \leftarrow 1$ と設定して，文書 d_i を読む。$h \leftarrow 1$ とする。

(2) 文書 $d_{i(h)}$ を読み，もし類似度 $s(\mathbf{d}_i, \mathbf{d}_{i(h)})$ の値が閾値 θ を超えていれば (2-a) に進む。そうでなければ (3) へ跳ぶ。

(2-a) もし d_i と $d_{i(h)}$ の両方がどのクラスタにも属していなければ ($g(d_i)=g(d_{i(h)})=0$)，これらを併せて 1 つのクラスタを構成し，(3) へ跳ぶ。そうでなければ (2-b) に進む。

(2-b) もし $d_{i(h)}$ のみがどのクラスタにも属していなければ，d_i の属するクラスタに $d_{i(h)}$ を割り当てたのち，(3) へ跳ぶ。そうでなければ，(2-c) に進む。

(2-c) もし d_i と $d_{i(h)}$ とが同一クラスタに属していれば ($g(d_i)=g(d_{i(h)})$)，何もせずに (3) へ跳ぶ。そうでなければ，(2-d) に進む。

(2-d) d_i の属するクラスタと $d_{i(h)}$ が属するクラスタとを併合し，(3) へ進む。

(3) 文書集合 $\tilde{D}_i$ 中の次の文書に処理を移すために，$h \leftarrow h+1$ とする。もし $h \leq n$ ならば (処理する文書が残っていれば)，(2) に戻る。そうでなければ，(4) に進む。

(4) 次の文書に処理を移すために，$i \leftarrow i+1$ とする。もし，$i \leq N$ ならば (処理する文書が残っていれば)，$h \leftarrow 1$ としたうえで (2) に戻る。そうでなければ，クラスタリングの処理を終了する。

Lightweight アルゴリズムは，最終的には，文書 d_1 から d_N まで順に処理していく中で，クラスタを計算するため，本稿ではこの節に含めたが，実際には，このアルゴリズムを本当に単一パス・アルゴリズムに分類してよいかどうかは微妙である。このアルゴリズムでは，基本的に，文書の各組についての $N \times N$ の類似度行列が出発点となっている。この点は階層的クラスタ分析法と同様であるが，Lightweight アルゴリズムは，この類似度行列から，文書ごとに一部のデータ（n 個の類似度）を取り出して，断片的に使用していく点に特徴がある。計算量の観点では，$\tilde{D}_1, \ldots, \tilde{D}_N$ を得るのに，類似度行列を求めるための $O(N^2)$ に加えて，N 回のソートの実行が必要になる。そして，単一パス・アルゴリズム的にクラスタを形成する部分で，$N \times n$ 回の処理を要することになる。

なお，2 つの文書の類似度 $s(\mathbf{d}_i, \mathbf{d}_h)$ の計算には，tf-idf が利用される。ただし，tf については，2 値とする。すなわち，語 t_j が文書 d_i に出現すれば $b_{ij}=1$，出現しなければ $b_{ij}=0$ と定義して，文書ベクトルの要素を，

$$w_{ij}=b_{ij}\left(1+\frac{1}{n_j}\right)$$

とする。そして，2 つのベクトル $\mathbf{d}_i$ と $\mathbf{d}_h$ との内積 $\sum_{j=1}^{M} w_{ij}w_{hj}$ によって，類似度 $s(\mathbf{d}_i, \mathbf{d}_h)$ が計算される。Lightweight アルゴリズムでは，1 つの文書ベクトルに使用する語数の上限を決めて，計算量を減らす工夫が加えられている。具体的には，1 件の文書中の語をその出現頻度の順に並べ，上位 m 語までを文書ベクトルに使用する。

以上の説明から明らかなように，このアルゴリズムでは，θ, n, m の 3 つのパラメータを先験的に与えておく必要がある。

B. 階層的クラスタリング

1. 階層的クラスタ分析法の応用

1970 年代に，文書集合に階層的クラスタ分析法を適用し，その結果を検索に活用する方法が考案され，その後いくつかの実験が試みられた。これは文書集合をあらかじめクラスタリングしておいて，検索質問との類似度の高いクラスタを出力する方法である。通常の検索では，検索質問と各文書との類似度（適合度）が計算されるのに対して，単一の文書ではなくクラスタを計算対象として扱う点にこの方法の特徴がある。基本的には，クラスタリングの手法は階層型でも非階層型でも構わないが，階層型の場合には，樹形図の階層を上下することによって出力文書数の調整が可能と

いう利点がある。

階層的クラスタ分析法の中でも，初期の研究（Jardin と van Rijsbergen (1971)[40]，van Rijsbergen (1974)[84]，van Rijsbergen と Croft (1975)[86] など）では，主として，単連結法が利用されていた。単連結法には，データを処理する順序に結果が依存しないという利点がある（宮本 (1999)[60]）。具体的には，次のような手順で検索が実行される (van Rijsbergen (1974)[84])。

1. 文書間の類似度を定義し，類似度行列を計算する。
2. 類似度行列に対して単連結法アルゴリズムを適用し，樹形図を得る。
3. 樹形図の最上層から出発し，枝を下に降りていく。その際，各水準で最も検索質問と一致するクラスタを選択する。次に進んだときに，もしその一致度が減少したら処理を終え，一致度の最も高かったクラスタを出力する。

このうち，第3段階については，さまざまな方法が提案されており，上の説明はそのなかの1つの例にすぎない（例えば，樹形図の下から上昇していく方法もある）。

類似度の計算方法としては，初期の研究では，文書ベクトルの要素を2値として，余弦係数，Dice 係数，Jaccard 係数，重複係数などが使われた。a を一方の文書に含まれる語数，b を他方の文書に含まれる語数，c を両者に共通して含まれる語数とすれば，これらの係数はそれぞれ，

$$\frac{c}{\sqrt{a}\sqrt{b}},\quad \frac{2c}{a+b},\quad \frac{c}{a+b-c},\quad \frac{c}{min(a,b)}$$

で求められる。検索質問とクラスタとの一致度も，基本的には，同様な方法で計算される。

以上のような階層的クラスタ分析法を使った検索の妥当性を保証するのが，いわゆるクラスタ仮説（Jardin と van Rijsbergen (1971)[40] や van Rijsbergen (1979)[85]）である。これは，2つの文書が類似している場合，それらは同一の検索質問にともに適合している可能性が高く，それに対して，類似していない場合には，同一の検索質問に適合する可能性は低いという仮説である。この仮説が成立する文書集合に対しては，クラスタリングに基づく検索は成功する可能性が高い。実際に，このクラスタ仮説が成立するかどうかを実証的に調べた研究（van Rijsbergen と Sparck Jones (1973)[87] や El-Hamdouchi と Willett (1987)[24]）もある。

一般に単連結法は「偏った」樹形図を出力しやすく，これは「chain effect」などと呼ばれている。実際，情報検索の場合でも，そのような状況が観察されており (Murtagh (1984)[62])，完全連結法や群平均法などの適用も試みられている (Murtagh (1984)[62] や Griffith ほか (1984)[29])。

以上のような検索への応用は主として1970年代から80年代に盛んに研究されたが（Willett (1988)[93] を参照），それ以外にも，階層的クラスタ分析法を応用した事例がいくつかある。例えば，すでに述べたように，Scatter/Gather のアルゴリズムでは，クラスタの中心点を発見するのに，階層的クラスタ分析法を使っている。また，WebCluster システム（Muresan と Harper (2001)[61]）では，完全連結法や群平均法による階層的クラスタを活用して，利用者の検索を支援する工夫が取り入れられている。さらに，TDT の研究においても，階層的クラスタ分析の適用例がある（Hatzivassiloglou ほか (2000)[32] や Papka と Allan (2000)[69]）。

2. 転置索引ファイルの利用

階層的クラスタ分析法を大規模な文書集合に適用する場合の最大の問題点はその計算量にある。N 件の文書に対して，それぞれの組ごとに類似度を計算する必要があり，その組数は全部で $N(N-1)/2$ となる。したがって，計算量はこの部分だけで $O(N^2)$ である。

類似度行列が大きければ，それを保存する領域についても注意を払う必要がある。Anderberg による古典的な教科書 (Anderberg (1973)[1]) では，階層的クラスタ分析法を実行する主な方法として，(1) 行列内蔵法，(2) データ内蔵法，(3) 分

類行列法の3つが挙げられている。このうち，(1)が主記憶に類似度行列が収まる場合であり，(2)と(3)はそうでない場合の対処法である。(2)は，結果として計算される類似度行列よりも，その元データのほうが小さい場合を想定したもので，元データを主記憶に格納しておき，クラスタリングの途中で，必要に応じて類似度を計算する方法である。また(3)では，類似度行列をいったん外部記憶装置に出力して，その上でソートを実行する。そして，その結果を順次読み取って，クラスタリングを実行していく。一般的には，階層的クラスタ分析法のアルゴリズムの研究において，例えばSLINK (Sibson (1973)[75]) などの効率的な方法も考案されている。

一方，類似度行列の計算量の問題は，「共有する語がなければ，それらの文書間の類似度は自動的に0になり，しかもそのようなケースが頻発する」という文書クラスタリングの特殊事情を考慮すれば，いくぶん緩和される可能性がある。例えば，Croft (1977)[16] は，転置索引ファイルを使って，効率的に単連結法を実行する方法を提案している。転置索引ファイルでは，索引語ごとに，それを含む文書集合が記録されており，この集合中の文書は必ず1語以上の語を共有している。ここで，転置索引ファイルに M 個の索引語が登録されていたとする。もしそれらに対応した M 個の文書集合それぞれに対して，個別に単連結法を適用したとすれば，語を共有しない文書間の計算は絶対になされない。したがって，計算量は $O(N^2)$ よりも少なくなる可能性がある。結果として得られる M 個の樹形図には，同一の文書が重複して含まれることになるが，上で述べたクラスタによる検索に応用目的を限定すれば大きな問題は生じない。ただし，1件の文書中に含まれる語の数が多く，各文書が M 個の集合中に繰り返し出現するような場合には，逆に計算量が増えてしまう可能性がある（Harding と Willett (1980)[31]）。結局，Croft (1977)[16] の方法は文書長がかなり短い場合（例えば，文書の標題のみから成るデータなど）に，適した方法であるといえる。

Willett (1981)[92] は，この問題を解決するために，類似度行列を計算する前に，転置索引ファイルを使って，各文書が他の文書と語を何語共有するかを記録したファイルを作っておく方法を提案した。例えば，3つの語 t_1, t_2, t_3 があったときに，転置索引ファイルから，

t_1: 1, 1, 0, 1, 0, 1, 1, ...

t_2: 0, 1, 0, 0, 0, 1, 1, ...

t_3: 0, 1, 0, 0, 0, 0, 1, ...

という情報が読み取れたと仮定する。「t_1: 1, 1, 0, ...」は，文書 d_1 と d_2 がこの語を含み，d_3 は含んでいないことを意味している。ここで，ある文書 d_i が，語 t_1, t_2, t_3 の3語のみを含んでいたとする。この場合，上のリストを上から順に足し合わせて，

V_i: 1, 3, 0, 1, 0, 2, 3, ...

というベクトルを作成したとすれば，これは，文書 d_i が各文書とそれぞれ共有する語数，すなわち，上記の余弦係数などの分子 c のリストに相当することになる（例えば，d_1 とは1語，d_2 とは3語を共有し，d_3 とは語を共有しないことが読み取れる）。したがって，各 $V_i (i=1, \ldots, N)$ と，各文書の語の総数（a や b に相当）のリストをB木やハッシュ法などを使って実装しておけば，余弦係数などの類似度の計算が高速になる。

岸田 (2003)[46] は，このWillettの方法を若干修正し，さらに，上で説明したAnderberg (1973)[1] における分類行列法を組み合わせることによって，大規模な文書集合への単連結法の適用を試みている。この研究では，文書間の類似度は余弦係数で計算され，また，その重み w_{ij} は，標準的なベクトル空間モデルに基づいて，式(18)が使用されている。この値 $w_{ij} (i=1, \ldots, N)$ は，もし転置索引ファイルにtf（すなわち x_{ij}）の情報が含まれていたならば，上記のWillettの方法によって計算可能である（すなわち，上記の「1」の代わりに x_{ij} の値が得られることになる）。具体的な手順は以下のとおりである。

1. 1件の文書 d_i のレコードを読み，それに含まれる各語について，それぞれ転置索引ファイルを探索する。

2. 転置索引ファイルには x_{ij} および出現文書数 n_j の値が記録されているので，文書 d_i と1語以上を共有する他の文書に関して，式(2)の分子の値を計算できる。この値をファイルに書き出しておく。
3. また，この転置索引ファイルの探索によって，文書 d_i 自体のノルムも計算できるので，その値を，上記2.とは別のファイルに書き出しておく。
4. 上記の手順をすべての文書に対して繰り返す。

以上の作業によって作成された2つのファイルを再度，読み出すことによって，1語以上を共有するすべての文書ペアに対する類似度を計算でき，次に，それを降順にソートすれば，Anderberg (1973)[1] における分類行列法の適用が可能になる[46]。

3. Voorheesによる単連結法アルゴリズム

Voorhees (1986)[88] は，1980年代半ばに，転置索引ファイルを利用した階層的クラスタ分析のアルゴリズムを応用し，約12,000件の文書集合のクラスタリングを試みている。この研究では，単連結法，完全連結法，群平均法が取り上げられているが，ここでは，そのうち単連結法のアルゴリズムのみを示す。このアルゴリズムでは，分析対象の N 件の文書に対して，大きさ N の配列を4つ使用する（したがって $O(N)$ の領域となる）。これらはそれぞれ

(a) *nn*[ ]: 各文書に最も近い文書の番号（添字）を記録しておくための配列
(b) *sim*[ ]: 各文書に最も近い文書に対する類似度を格納するための配列
(c) *InHier*[ ]: 各文書が樹形図に取り込まれたかどうかを記録するための配列
(d) *sim2*[ ]: ある1つの文書に対する他の文書の類似度を計算した結果を一時的に保存するための配列

である。

なお，文書番号を表現する添字は1から N まで動くものとし，0は「未定義」を意味すると約束しておく。また，次の変数を用意する。

(i) *CurrId*: 処理対象となる1件の文書の番号（添字）を記録するための変数
(ii) *MaxSim*: 類似度の最大値を見つけるための記録用変数
(iii) *NextId*: 次の処理対象となる1件の文書の番号（添字）を記録するための変数

Voorheesによる単連結法

(1) $i=2,\ldots,N$ について *sim*[*i*], *InHier*[*i*], *nn*[*i*] を初期化する。すなわち，*sim*[*i*]←0.0, *InHier*[*i*]←0, *nn*[*i*]←0 ($i=2,\ldots,N$)。また，第1番目の文書を樹形図に取り込むとともに，*CurrId* をこの番号に初期化する (*CurrId*←1)。
(2) もし，*CurrId* が0ならばすべての処理を終了する。そうでなければ，*CurrId* が樹形図にすでに取り込まれていることを記録する (*InHier*[*CurrId*]←1)。
(3) *CurrId* に対する他の文書の類似度を計算し，その結果を *sim2*[ ] に記録する。
(4) *MaxSim*←0.0, *NextId*←0, および *i*←1 として，(4-a) に進む。
(4-a) もし *i* 番目の文書がまだ樹形図に取り込まれていないならば（すなわち *InHier*[*i*] が0ならば），(4-b) に進む。そうでなければ，何もせず (4-d) に跳ぶ。
(4-b) もし *sim2*[*i*]>*sim*[*i*] ならば，*CurrId* の文書が，*i* 番目の文書にその時点で最も近いことになるので，*nn*[*i*] と *sim*[*i*] を更新する。すなわち，*nn*[*i*]←*CurrId*, *sim*[*i*]←*sim2*[*i*]。
(4-c) *sim*[*i*] が *MaxSim* よりも大きければ，*MaxSim* を更新し (*MaxSim*←*sim*[*i*])，その文書の番号を *NextId* に仮に記録しておく (*NextId*←*i*)。
(4-d) *i*←*i*+1 とする。この結果もし *i* が N を超えれば (5) に進む。そうでなければ (4-a) に戻る。
(5) もし *NextId*≠0 ならば，*NextId* の文書を樹形図に追加する。
(6) *CurrId*←*NextId* のように更新し，(2) に戻

る。

Voorheesのアルゴリズムでは，各時点でまだ樹形図に取り込まれていない文書のうち，すでに取り込まれている文書との類似度が最大のものを探していく（上の手順の(4-c)）。新たな文書が樹形図に取り込まれることに伴って，クラスタ間の類似度を更新しなければならないが，その作業は，新たな文書が取り込まれた次のループ（(2)から始まるループ）での手順(4-b)でなされることになる。この結果，最終的に，単連結法による樹形図を描くことができる。

なお，手順(3)における *sim*2[ ]の計算は，*CurrId* が更新されるたびに実行されるので，この部分の計算量は合計で $O(N^2)$ とならざるを得ない。一方，*sim*2[ ]は大きさ N の配列なので，主記憶の容量は $O(N)$ に節約されている。したがって，*sim*2[ ]の計算に対して，転置索引ファイルを利用した処理時間の短縮を工夫することが重要となる。

4. Suffix Tree Clustering (STC)

Suffix Tree Clustering (STC) は，ZamirとEtzioni (1998)[98] によって提案された，文書クラスタリングの方法であり，接尾辞木 (suffix tree) を利用する点に特徴がある。さらに，この方法を用いて，Web文書に対する検索結果を動的にクラスタリングするためのインタフェースであるGrouperも開発されている（ZamirとEtzioni (1999)[99]）。

STCでは，まず最初に文書集合全体に対して，接尾辞木を構成する。接尾辞木の各ノードには，それに該当するいくつかの文書を割り当てることができるため，各ノードをそれぞれ1つのクラスタとみなすことが可能になる。これらのクラスタは「基本クラスタ (base clusters)」と呼ばれる。これはちょうど，転置索引ファイルにおける1つの索引語を含む集合を1つのクラスタと考えることに似ている。

この場合，当然，1件の文書が複数の基本クラスタに重複して属することになるが，次の段階では，この重複の程度を利用して，基本クラスタのうち，類似したものを併合していく。まず，クラスタ C_k と C_h に含まれる文書数をそれぞれ $\tilde{n}_k, \tilde{n}_h$ とし，さらに両者に重複して含まれる文書数を $\tilde{n}_{kh}$ とおく。ここで，これらの2つのクラスタの類似度 $s(C_k, C_h)$ を，

$$s(C_k, C_h)=\begin{cases}1, & \tilde{n}_{kh}/\tilde{n}_k>0.5 \text{ かつ} \\ & \tilde{n}_{kh}/\tilde{n}_h>0.5 \text{ の場合} \\ 0, & \text{それ以外}\end{cases}$$

で定義する。そして，この2値の類似度を使って，基本クラスタのグラフを構成する。つまり，類似度が1である2つのクラスタを連結していき，連結された1つのグラフを1つのクラスタとみなす。この方法は，原理的には単連結法に等しいが，類似度0の基本クラスタは連結されないので，最終的に1つの大きなクラスタにまとまる前に停止する。

5. Scatter/Gatherにおける群平均法

何度か述べているように，Scatter/Gatherのアルゴリズムでは，階層的クラスタ分析法（群平均法）が使われている（Cuttingほか (1992)[19]）。したがって，やはり階層的クラスタリングの処理速度がここでも問題とならざるを得ないが，この計算が高速となるようBuckshotおよびFractionationといった工夫が導入されている（本章のA節参照）。

また，Scatter/Gatherでは，この部分の処理の効率を高めるために，文書ベクトルの要素のうち，その重みが大きい上位 m 個のみを使用している（それ以外は重み0とする）。この点に関しては，その後，詳しく調査され，わずか20個程度の要素に限定した場合，その処理時間は大幅に減少するにもかかわらず，クラスタリングの品質はそれほど損なわれないことが報告されている（SchutzeとSilverstein (1997)[74]）。なお，この研究では，同時に，LSIによる次元縮約を利用して，文書ベクトルの要素を減らす方法についての分析も試みられている。

6. 情報ボトルネック法に基づくクラスタリング

Slonim と Tishby (2000)[77] は，情報ボトルネック法 (information bottleneck method) に基づいて，文書（または文書クラスタ）の分布と語（または語クラスタ）の分布との間の相互情報量 (mutural information metrics) を使った，新しい類似度を提案している。この相互情報量は，例えば，

$$M_I(C;\Omega)=\sum_{k,j:\,C_k\in C,\,t_j\in\Omega}P(C_k)P(t_j|C_k)\times\log\frac{P(t_j|C_k)}{P(t_j)} \qquad (25)$$

のように定義される。ここで C は文書クラスタの集合（すなわち，ある1つのクラスタリングの結果），Ω は語の集合である。具体的には，この式に含まれる確率は，

$$P(t_j|C_k)=\frac{1}{P(C_k)}\sum_{i:\,d_i\in C_k}P(d_i)P(t_j|d_i)$$

$$P(t_j|d_i)=\frac{x_{ij}}{\sum_{j=1}^{M}x_{ij}},\quad P(C_k)=\sum_{i:\,d_i\in C_k}P(d_i)$$

$$P(d_i)=\frac{1}{|D|}=\frac{1}{N},\quad P(t_j)=\frac{1}{|\Omega|}=\frac{1}{M}$$

のように操作的に定義される。式 (25) は，語の集合 Ω が与えられたときの，文書クラスタの集合 C の情報量であり，これを使えば，凝集型の階層的クラスタリングが可能になる。つまり，併合前と併合後の $M_I(C;\Omega)$ の値の差が小さい順に（併合によって失われる情報量の小さい順に），クラスタを併合していけばよい（最初の段階では，各 d_i を1つのクラスタ，すなわち singleton と考える）。

さらに，Slonim と Tishby (2000)[77] では，最初に語のクラスタリングを実行し，それによって構成された語のクラスタを用いて，文書クラスタリングを試みる，“double-clustering” が提案されている。すなわち，式 (25) と同様に，語のクラスタ（これを $\tilde{T}=\{\tilde{T}_1,\ldots,\tilde{T}_L\}$ と表記する）と個別文書との相互情報量 $M_I(\tilde{T};D)$ を定義できるわけであり，これを使ってまず，語のクラスタを構成する。そして，語そのものではなく，そのクラスタを使って，文書の表現をよりコンパクトなものにしておいてから，$M_I(C;\tilde{T})$ に基づいて，階層的なクラスタリングを実行する。なお，$\tilde{T}_h(\in\tilde{T})$ に対して，

$$P(\tilde{T}_h|d_i)=\frac{\sum_{j:\,t_j\in\tilde{T}_h}x_{ij}}{\sum_{h=1}^{L}\sum_{j:\,t_j\in\tilde{T}_h}x_{ij}}$$

と定義する。

7. 分割型の階層的クラスタリング

階層的クラスタ分析法は一般的には凝集型 (agglomerative) であり，個々の対象から出発して，それらを段階的に併合していく。このようなボトムアップ型のアプローチに対して，トップダウン的なアプローチを採用することも可能である。すなわち，文書集合全体 D から出発して，それを逐次的に分割していけばよい。もし各段階で常に2つのクラスタに分けていくならば，結果的に2分木が構成され，これは凝集型アルゴリズムによる樹形図と形式的には等しくなる。

例えば，Karypis を中心とするグループは，k-means 法を使って，各段階での文書集合（クラスタ）の分割を試みている（Steinbach ほか (2000)[81]，Zhao と Karypis (2002)[100] など）。この方法は，bisecting k-means 法と呼ばれるが，実際の手順は以下のとおりである。

基本的な bisecting k-means 法

(1) 分割の対象となる文書集合を $\mathscr{H}$ と表記する。初期設定として，$\mathscr{H}=D$ とおく。
(2) $\mathscr{H}$ の中から，何らかの方法によって2つの文書を選び，それを種子点とする。
(3) 2つの種子点を使って，k-means 法を実行し，新たに2つのクラスタを生成する。
(4) その時点でまだ分割されていないクラスタの中から，何らかの基準を使って，1つのクラスタを選び，それを $\mathscr{H}$ として (2) に戻る。もし，この際，1つもクラスタが選ばれない（基準を満たすクラスタがない，もしくは処理の終了条件に到達した）場合には，処理を終了する。

具体的には，段階 (2) では単純無作為抽出によって2件の文書を選ぶこととし，段階 (4) では，最大のクラスタを選択することにすれば，とりあえずは，上記の手順を実行できる。このような分割型

のk-means法については，例えば石岡(2000)[38]など，一般的にも研究されている。

分割型の場合，その計算量は，単純な凝集型に比べて，減少することが期待できる。つまりN個の対象を2分木に構成する場合，もし，この2分木が「完全に平衡」ならば，各段階でのk-means法で走査される文書数の総計は，

$$\sum_{h=0}^{\log_2 N-1} 2^h \times \frac{N}{2^h} = N + N(\log_2 N - 1)$$
$$= N \log_2 N$$

であり，すなわち，計算量は$O(N \log N)$程度となる。

上記の手順をより洗練させるには，クラスタの「質」に関する何らかの基準関数を設定し，それに基づいて，段階(3)でのk-means法や，段階(4)でのクラスタ選択を精緻化することが考えられる。その基準関数としては，次のようなものが提案されている（Zhao と Karypis (2002)[100]）。

1. クラスタ内の基準関数：以下のJ_1とJ_2のいずれかを最大にする。

$$J_1 = \sum_{k=1}^{L} \tilde{n}_k \left(\frac{1}{\tilde{n}_k^2} \sum_{i,h:\, d_i, d_h \in C_k} s(\mathbf{d}_i, \mathbf{d}_h) \right) \qquad (26)$$

または

$$J_2 = \sum_{k=1}^{L} \sum_{i:\, d_i \in C_k} s(\mathbf{d}_i, \mathbf{m}_k) \qquad (27)$$

2. クラスタ間の基準関数：以下のJ_3を最小にする。

$$J_3 = \|\mathbf{D}\| \sum_{k=1}^{L} \tilde{n}_k s(\mathbf{m}_k, \mathbf{m})$$

ここで，$\mathbf{D}$は文書集合全体をクラスタとして考えた場合のベクトルであり，$\mathbf{m}$はその重心である（実際には，$\|\mathbf{D}\|$は定数で，式中で約分されて消える）。

3. 混合的な基準関数：以下のJ_4とJ_5のいずれかを最大にする。

$$J_4 = \frac{J_1}{J_3} \quad \text{または} \quad J_5 = \frac{J_2}{J_3}$$

この方法はincremental k-means法の一種である。また，上記手順の段階(4)におけるクラスタ選択の際にも，クラスタを1つ増やしたときに，上記の基準を最も改善するようなクラスタを選ぶことにすれば，より望ましい結果を得ることができるかもしれない。例えば，単に大きなクラスタを選択すると，本来的に大きなクラスタを無理に分割してしまうことがありうるが，上記の基準を適用すれば，このような分割を避けることができる。以上の手法のほかに，Boley ら(1999)[6]によって考案されたPDDPアルゴリズムもまた，分割型の階層的クラスタリングを実現する方法として位置づけられる。この方法は，主成分分析を利用して，文書集合を2分割していくものであり，その詳細については，次節で述べることとする。

8. 制限された凝集型クラスタリング

bisecting k-means法による文書クラスタリングを試みたKarypisのグループは，同時に，「制限された凝集型クラスタリング(constrained agglomerative clustering)」を提案している（Zhao と Karypis (2002)[100]）。これは，第1段階としてまずk-means法を適用し，いくつかのグループに分割しておいてから，個々のグループごとに凝集型の階層的クラスタリングを実行する方法である。

上記のJ_1～J_5の基準を適用したk-means法は，文書集合全体に関する情報を利用しているのに対して，凝集型のアルゴリズムの場合，クラスタの構成は非常に局所的な情報のみによってなされる。このため，凝集型の方法では，小さくまとまったクラスタを識別しやすいのに対して，処理の最初の段階で不当な併合が生じると，その影響が歯止めなしに広がって，本来，分割されるべきクラスタがまとまってしまう可能性がある。そこで，第1段階のk-means法で，ある程度の均質なグループにまとめて，そのような影響の波及を防ぐようにすれば，クラスタリングの質が向上するという着想である。

さらに，文書集合全体に対して凝集型のアルゴリズムを適用するよりも，計算量がかなり減少するという利点がある。すなわち，第1段階でのk-means法によって，L'個のグループが生成され，

それぞれ，$\tilde{n}_1, \ldots, \tilde{n}_{L'}$ 件の文書が含まれるとすれば，

$$O(N^2) > \sum_{k=1}^{L'} O(\tilde{n}_k^2)$$

となることが期待できる。この点でも，この方法は有望であるといえよう。

同様に，Smeaton ら (1998)[80] もまた，大規模文書集合（約 34,000 件の新聞記事）をクラスタリングするために，第 1 段階で文書集合をグループに分割し，次に，各グループに対して，凝集型アルゴリズム（完全連結法）を適用するというかたちをとっている。ただし，第 1 段階では，k-means 法を用いるのではなく，各文書を検索質問とみなして，それを除いた文書集合に対して検索を実行し，上位 n 件の部分集合を 1 つのグループとしている。つまり，N 回の検索が繰り返され，N 個のグループが設定されるわけである（当然，これらのグループは重複する可能性がある）。n については，$n=30, n=40$ などと設定されるため，これらに対する完全連結法にはそれほど時間がかからない（ただし，完全連結法の実行が N 回繰り返される）。一方，第 1 段階での N 回の検索実行にはかなり時間を要するようである。

C. 次元縮約法に基づくクラスタリング

1. PDDP アルゴリズム

PDDP (principal direction divisive partitioning) アルゴリズムは，主成分分析（主成分変換）を利用して，文書集合を逐次的に分割していくアルゴリズムである（Boley ほか (1999)[6]）。このアルゴリズムを適用すると，2 分木としてクラスタが階層的に構成されることになる。つまり，PDDP は，分割型の階層的アルゴリズムの一種であるといえるが，行列 $\mathbf{W}$ に対して主成分分析を応用する方法の代表例として，この節で議論することにする。

まず，行列 (3) 式の各要素を，

$$w_{ij} = \frac{x_{ij}}{\sqrt{\sum_{j=1}^{M} x_{ij}^2}} \qquad (28)$$

で定義する。これは，行列 $\mathbf{W}$ における各文書の長さ（ノルム）を 1 に標準化することを意味する。

その上で，$\mathbf{W}$ に対する転置行列（すなわち，語×文書での重み行列）の共分散行列（$N \times N$ 行列），

$$\mathbf{S} = (\mathbf{W}^T - \mathbf{u}\mathbf{e}^T)^T(\mathbf{W}^T - \mathbf{u}\mathbf{e}^T) \qquad (29)$$

を対角化する行列 $\mathbf{G}^T\mathbf{S}\mathbf{G} = \Lambda$ を求める。ここで，$\mathbf{u}$ は M 次元の平均ベクトルで，その j 番目の要素は，$u_j = N^{-1} \sum_{i=1}^{N} w_{ij}$ である ($j=1, \ldots, M$)。また，$\mathbf{e}$ はその要素がすべて 1 である N 次元ベクトル $\mathbf{e} = (1, 1, 1, \ldots, 1)^T$ とする。

この結果として計算された対角行列 Λ の各要素は行列 $\mathbf{S}$ の固有値であり，そのうちの最大の固有値に対応する，行列 $\mathbf{G}$ 中の N 次元列ベクトルを $\mathbf{g}_1$ と表記する。これは，語×文書の重み行列に対して主成分分析を実行し，「寄与率」の最も大きな第 1 主成分を求めたことにほかならない。

固有ベクトル $\mathbf{g}_1$ の第 i 番目の要素は，文書ベクトル $\mathbf{d}_i$ を第 1 主成分へ射影した値なので，この値が正であるグループと負であるグループの 2 つに，文書集合を分割できる。PDDP アルゴリズムでは，この結果分割された 2 つの文書集合のそれぞれに対して同様な処理を反復的に繰り返す。これによって，文書集合に関する 2 分木を構成することが可能となる。

なお，PDDP アルゴリズムでは，固有値および固有ベクトルの算出には，“Lanczos” 型の反復法 (Golub と van Loan (1996)[28]) が使用されている。これは疎な行列に適したアルゴリズムである[6]。

PDDP アルゴリズムを改良した NGPDDP (non-greedy version of PDDP) アルゴリズムも考案されている (Nilsson (2002)[63])。PDDP アルゴリズムが各分割の段階で常に第 1 主成分を使うのに対して，NGPDDP では，第 2 位以下の主成分による分割も認められる（第 1 主成分による分割が必ずしも最適な結果を与えるとは限らないため）。分割の各段階で，何番目の主成分を採用するかについては，各クラスタの散らばりの合計が小さくなるよう決められる（詳細は元の文献を参照）。

2. LSI の応用と IRR アルゴリズム

LSI は，情報検索の性能向上のために，行列 $\mathbf{W}$（または $\mathbf{W}^T$）に対して特異値分解 (SVD) を施し，

文書空間の次元を圧縮する手法である（Deerwester ほか (1990)[20]）。具体的に，語×文書の行列 $\mathbf{W}^T$ に対する特異値分解は

$$\mathbf{W}^T=\mathbf{UQV}^T$$

と書ける（ただし，$M>N$ を仮定）。ここで，$\mathbf{U}$ は $M\times N$ の直交行列，$\mathbf{Q}$ は $N\times N$ の対角行列，$\mathbf{V}$ は $N\times N$ の直交行列である。ただし，$\mathbf{W}$ のランクを $r(\leq N)$ とすると，$\mathbf{Q}$ における $N-r$ 個の対角要素は 0 である（この節では記号 r は計算の反復回数でなく，行列のランクを示すものとする）。

ここで，$\mathbf{Q}$ における 0 でない r 個の対角要素に対応する $\mathbf{V}$ の列ベクトルを取り出して，

$$\mathbf{v}_{(1)}, \mathbf{v}_{(2)}, \ldots, \mathbf{v}_{(r)}$$

と表記する。ここで，各 $\mathbf{v}_{(k)}$ は N 次元ベクトルであり ($k=1,\ldots,r$)，SVD によって抽出された r 個の次元に対する各文書の値を示している。したがって，ある閾値を超えた文書のみがその次元に「属する」と仮定すれば，$\mathbf{V}$ によって非排他的な r 個のクラスタが得られることになる。また，この値を利用すれば，本来の語数 M よりも少ない r 次元での文書ベクトルを構成することができ，これを一種の特徴抽出として利用することも可能である。

以上の LSI を，情報検索でなく，文書クラスタリングに応用する際の問題点は，規模の小さなクラスタがうまく抽出されないことにある。本来，LSI は，全体的に影響力の小さい語を一種のノイズとして排除することによって検索性能を向上させる効果を持つが，同時に，他の文書とあまり類似しないような文書も除かれる傾向にある。その結果，相対的に小さなクラスタが抽出されない可能性がある。

この点を補うために，Ando によって IRR (iterative residual rescaling) アルゴリズムが考案された（Ando (2000)[2], Ando と Lee (2001)[3]）。その基本的なアイデアは，元の行列から次元を逐次的に抽出することによって，それより前の段階で取り出された次元に関連しない部分を説明するような次元の選択を可能にすることにある。このためには，前段階で取り出された次元を差し引いた「剰余 (residual)」を計算し，そこから次の次元を抽出するようにすればよい。なお，処理を施す行列をここでは $\mathbf{R}$ と表記し（$M\times N$ 行列），その各列を $\mathbf{r}_i$ と書く ($i=1,\ldots,N$)。また，ある係数 q を使った，$\mathbf{R}$ のスケール変換を

$$\mathbf{R}_s=(\|\mathbf{r}_1\|^q\mathbf{r}_1, \|\mathbf{r}_2\|^q\mathbf{r}_2, \ldots, \|\mathbf{r}_N\|^q\mathbf{r}_N) \qquad (30)$$

で定義する。なお，これまで同様，$M>N$ を仮定しておく。

IRR アルゴリズム

(1) 初期設定として，$\mathbf{R}=\mathbf{W}^T$ とする。また q をある値に定め，$k\leftarrow 1$ とする。前もってクラスタ個数 L を決めておく。
(2) $\mathbf{R}$ に対して，式 (30) を計算し，$\mathbf{R}_s$ を求める。
(3) $\mathbf{R}_s\mathbf{R}_s^T$ の固有ベクトルを計算し，そのうち，最大の固有値に相当するベクトルを $\mathbf{b}_k$ とする（M 次元ベクトル）。
(4) $\mathbf{R}\leftarrow\mathbf{R}-\mathbf{b}_k\mathbf{b}_k^T\mathbf{R}$ として，$\mathbf{R}$ を更新する。
(5) $k\leftarrow k+1$ とする。$k>L$ ならば，L 個のベクトル $\mathbf{b}_k$ が求められたことになるので，処理を終了する。そうでなければ (2) に戻る。

この結果，順次得られたベクトル $\mathbf{b}_1,\ldots,\mathbf{b}_L$ を使って，文書ベクトル $\mathbf{b}_i$ を，

$$\bar{\mathbf{d}}_i=(\mathbf{b}_1,\ldots,\mathbf{b}_L)^T\mathbf{d}_i$$

のように，より次元数の少ない，L 次元ベクトルに変換することができる。この変換後のベクトルを使えば，上で述べたような方法を用いて，クラスタリングや特徴抽出を実行することが可能である。

3. 主成分分析の応用

上記の LSI や IRR アルゴリズムの実際的な問題点として，N が大きい場合に特異値分解や固有値の計算が難しくなることが挙げられる。IRR アルゴリズムでは，手順の段階 (3) で $M\times M$ の行列の固有値を計算する必要があり，$M>N$ を仮定しているので，N が大きくなれば，当然，この計算は困難になる。仮に，語の選別を実施し，$M<N$ にした場合には，IRR アルゴリズムに入力するデータ行列を $N\times M$ の行列 $\mathbf{W}$ としなければならず，この結果，段階 (3) で固有値を計算する行列

は$N\times N$となるため，依然，問題は解決しない。

それに対して，主成分分析の場合には，語を選別した後の$N>M$のデータに対する，$M\times M$の共分散行列の固有ベクトルを求めればよいので，その実行の可能性は高くなる。KobayashiとAono (2004)[47]は，この着想に基づいて，IRRアルゴリズムに改良を加えた，COV-rescaleアルゴリズムを考案した（詳細は元の文献を参照）。実際に，この研究では，このアルゴリズムをTREC[82]の10万件を超える文書集合に適用し，良好な結果を得ている（選別された語の数は約1万語）。

4. NMFに基づくクラスタリング

Xuら(2003)[96]は，行列$\mathbf{W}^T$をSVDではなく，non-negative matrix factorization (NMF)によって分解し，それに基づいて文書集合をクラスタに分割することを試みている。NMFの場合，$\mathbf{W}$を，目的関数

$$J_n=\frac{1}{2}\|\mathbf{W}^T-\mathbf{U}\mathbf{V}^T\| \tag{31}$$

が最小になるような，$M\times L$行列$\mathbf{U}$と$N\times L$行列$\mathbf{V}$とに分解する（記号の節約のため，SVDと同じ記号を使って，行列を表現しておく）。ここで，$\|\cdot\|$は，行列に含まれるすべての要素の2乗の合計を意味している。

この最小化問題は，

$$u_{ij}\leftarrow u_{ij}\frac{(\mathbf{W}^T\mathbf{V})_{ij}}{(\mathbf{U}\mathbf{V}^T\mathbf{V})_{ij}}$$

$$v_{ij}\leftarrow v_{ij}\frac{(\mathbf{W}\mathbf{U})_{ij}}{(\mathbf{V}\mathbf{U}^T\mathbf{U})_{ij}}$$

の反復計算によって解くことができる（Xuほか(2003)[96]）。ここで，u_{ij}, v_{ij}はそれぞれ行列$\mathbf{U}$, $\mathbf{V}$の第i行，第j列の要素であり，同様に，$(\cdot)_{ij}$は任意の行列の第(i,j)要素を意味するものとする。

この結果として計算されたv_{ij}を，i番目の文書がj番目のクラスタに属する程度として解釈すれば，この値を使ってクラスタを構成できる。例えば，文書d_iをそのv_{ij}の値が最も大きなクラスタC_jに割り当てることにすればよい（ただし，v_{ij}は標準化された値とする。標準化の方法については元の文献を参照）。

NMFの場合，SVDとは異なり，$\mathbf{U}$, $\mathbf{V}$は直交行列ではない。つまり，SVDに基づくLSIが，各語によって張られる文書空間を，より次元数の少ない直交空間として再構成しようとするのに対して，NMFに基づく方法では，語の出現パターンの類似した文書の集合（すなわちクラスタ）に，より直接的に対応するように軸が抽出されることになる。なお，NMFを計算する際には，Lの値（クラスタの個数に対応）を先験的に与えておく必要がある。

D. 確率モデルに基づくクラスタリング

文書クラスタリングでの適用例は多くはないが，何らかの確率分布を仮定して，1つの文書ベクトル$\mathbf{d}_i$が与えられたときのクラスタC_kの確率$P(C_k|\mathbf{d}_i,\Theta)$を求めることによって，文書集合をクラスタに分割することができる。ここで，Θは，確率分布のパラメータ（のベクトル）である。すなわち，文書d_iに対する各クラスタの$P(C_k|\mathbf{d}_i,\Theta)$ $(k=1,\ldots,L)$を推計し，その値が最大であるクラスタにその文書を含めることにすればよい（ベイズ型の分類）。

教師付きのテキスト分類の場合には，学習用データからΘを推定できる。一方，本稿が議論している「教師なし」の状況ではこの方法は使えない。しかし，適当な初期値を設定できれば，EMアルゴリズムを使って，Θを求めることは可能である(Hofmann (1999)[35])。例えば，Liuら(2002)[58]は確率分布として多次元ガウス分布を仮定し，EMアルゴリズムを用いて，そのパラメータ群を推計している。

実際には，Liuら(2002)[58]によって提案された文書クラスタリングの方法は，次のような特徴を持っている。

1. 「通常の語」，「固有名詞」，および，「統計的に求められた関連語の組」の3種類から構成される語句の集合を使った特異値分解によって，より少数の「特徴」f_j $(j=1,\ldots,m)$を抽出する。
2. ガウス型混合モデル (Gaussian Mixture

Model: GMM) に基づく EM アルゴリズムによって，初期的なクラスタ群を構成する。

3. 初期的なクラスタ群から，クラスタをよりよく識別する特徴を抽出し，それに基づいて，クラスタを再編成する。

この方法の特徴は，確率的なクラスタリングの結果を，事後的に，上記 3. での反復計算によって精緻化している点にある。事後的な精緻化が必要となるのは，学習用データのない状況において確率的な方法を適用することの難しさによるものと推察される。前段階で特異値分解を適用するのも，おそらく同様の理由からであろう。

具体的には，各文書ベクトルは，L 個のクラスタから構成されるモデル $\mathcal{M}$ から確率的に生成されると考える。すなわち，

$$P(\mathbf{d}_i|\mathcal{M})=\sum_{k=1}^{L} P(C_k)P(\mathbf{d}_i|C_k),\quad i=1,\ldots,N \tag{32}$$

とする。ここで，$P(\mathbf{d}_i|C_k)$ は，重心ベクトル $\mathbf{m}_k$ と共分散行列 Σ_k をパラメータとする，M 次元ガウス分布

$$P(\mathbf{d}_i|C_k)=\frac{1}{(2\pi)^{M/2}|\Sigma_k|^{(1/2)}}\times\exp\left(-\frac{1}{2}(\mathbf{d}_i-\mathbf{m}_k)^T\Sigma_k^{-1}(\mathbf{d}_i-\mathbf{m}_k)\right) \tag{33}$$

である（すなわち，GMM）。

(32) 式が最大になるような，パラメータ群 $\mathbf{m}_1,\ldots,\mathbf{m}_L,\Sigma_1,\ldots,\Sigma_L$ を求めるために，EM アルゴリズムを使う（Liu ほか (2002)[58]）。この際，L 個の重心ベクトルの初期値は，$\mathbf{m}_0=N^{-1}\sum_{i=1}^{N}\mathbf{d}_i$ および $\Sigma_0=N^{-1}\sum_{i=1}^{N}(\mathbf{d}_i-\mathbf{m}_0)(\mathbf{d}_i-\mathbf{m}_0)^T$ をパラメータとする正規分布からの無作為抽出によって設定する。また，Σ_k については，すべて等しい初期値とし，Σ_0 を使う。EM アルゴリズムを適用して，$\mathbf{m}_1,\ldots,\mathbf{m}_L,\Sigma_1,\ldots,\Sigma_L$ が求められたならば，各文書の属するクラスタを，式 (33) を使って決定する。

次に，このクラスタリングの結果に基づいて，クラスタをより識別する特徴 f_j を特定する。その基本的なアイデアは，ある特定のクラスタのみに出現し，他のクラスタには出現しないような特徴を「識別力のある」ものとして選ぶことにある。実際には，ある特徴 f_j がクラスタ C_k に出現する回数に基づくある指標を定義して，それによって識別力を持つ特徴を選別する（ここではその指標を $\beta(\mathrm{f}_j)$ と表記しておく。この指標に関する詳細は元の文献を参照)。選択された特徴の集合を Γ と書く。

Liu ら (2002)[58] のアルゴリズムの詳細は以下のとおりである。

GMM+EM アルゴリズムに基づく方法

(1) 初期設定として，クラスタの個数 L と，$\beta(\mathrm{f}_j)$ に対する閾値とを設定する。

(2) GMM+EM アルゴリズム（上述）によって，初期的なクラスタ群 $C_1, C_2, \ldots, C_L$ を生成する。

(3) 与えられたクラスタ群に対する各特徴の $\beta(\mathrm{f}_j)$ の値を計算し，閾値を超える特徴の集合 Γ を求める。

(4) すべての文書 $d_1,\ldots,d_N$ に対して，新しいクラスタを割り当てる。具体的には，各文書に含まれる「識別力のある特徴」$\mathrm{f}_j(\in\Gamma)$ に対して，それが最頻出するクラスタをそれぞれ調べ，その中で，最頻出のクラスタとして最も多く挙げられたものを，その文書が属する新たなクラスタとする。

(5) 新たに得られたクラスタ群とその前段階のクラスタ群とを比較し，変化がなければ処理を終了する。そうでなければ，(3) に戻る（反復計算)。

なお，このアルゴリズムでは，クラスタの個数 L を先験的に与えなければならないが，Liu ら (2002)[58] では，モデル選択の手法を使って，最適な L を見つける方法も考案されている。

E. データマイニングの手法の応用

1. 関連ルールの応用

データマイニングにおける基本的な問題として，関連ルール (association rule) の発見がある

(Han と Kamber (2001)[30])。例えば，ある店舗における売上記録（すなわちトランザクション）を分析したところ，ある商品 A と商品 B（例えば，パーソナルコンピュータと財務会計ソフト）とが同時に購入される傾向が強いということが明らかになったならば，$A \rightarrow B$ または $B \rightarrow A$ という関連ルールがこれらの商品の間に存在すると考える。このときもし売上レコードの全体で商品 A と商品 B とが同時に購入される割合が 10% であり，なおかつ，商品 A が購入されている売上レコードのうちの 50% が商品 B を購入しているとすると，関連ルール $A \rightarrow B$ の "support" は 10%，"confidence" は 50% として定義される。

Boley ら (1999)[6] は，各「売上レコード」を語，「商品」を文書とみなすことにより関連ルールの手法を文書クラスタリングに応用した。この場合，関連ルールは，「語を共有する」という観点からの文書間の関連を表していることになる。

実際には，Boley ら (1999)[6] の方法は一種の "multi-level hypergraph partitioning" アルゴリズムであり，関連ルールに基づいて構成されたハイパーグラフを，そのエッジの重みの減少が最小になるよう，逐次的に分割していく（グラフの節点を文書，エッジを文書間の関連と考える）。この際，エッジの重みは，各部分グラフに含まれる，関連ルールの "confidence" の平均として定義される（詳細は元の文献を参照）。

2. FTC および HFTC アルゴリズム

同様な着想は Beil ら (2002)[5] によっても試されている。彼らの方法は，FTC (frequency term-based clustering) と呼ばれ，文書集合中で頻出する語（の集合）に基づいてクラスタを構成する。

文書集合中に出現するすべての語の集合を Ω と表記し，その任意の部分集合 $F(\subseteq\Omega)$ に対して，F に含まれる語がすべて出現する文書の集合を $\mathrm{cov}(F)$ と書く（単語だけでなく，それらの組み合わせもまた F の要素であることに注意）。ある比率 θ_n を決め $(0<\theta_n<1)$，$|\mathrm{cov}(F)|>N\theta_n$ である F を「頻出語集合」と定義し，その集合を Ω_F と表記する。つまり，

$$\Omega_F=\{F\subseteq\Omega \mid |\mathrm{cov}(F)|>N\theta_n\} \quad (34)$$

であり，ある一定数以上（$\theta_n\times100\%$ 以上）の文書に出現する語集合を要素とする集合が Ω_F ということになる。

この $\mathrm{cov}(F)$ こそが 1 つの文書クラスタであり，FTC では，文書クラスタを構成するのに適した語集合 F を Ω_F の中から，反復的に選択していく。これによって選択された F の集合を Φ_F と書くことにする。

実際に Φ_F を求めるために，Ω_F の任意の部分集合 R に対して，その要素（すなわち語集合）が，各文書にいくつ出現するかを数えるための関数 $f_x(R, d_i)$ を用意する。そして，各 $\mathrm{cov}(F)$ に対して，

$$\mathrm{overlap}(F|R) = \sum_{i:\, d_i\in\mathrm{cov}(F)} -\frac{1}{f_x(R,d_i)}\log\frac{1}{f_x(R,d_i)} \quad (35)$$

を計算する。これは，ある語集合 F によって構成されたクラスタが，他のクラスタと重複する程度を表す指標である。例えば，$R=\{F_1, F_2, \ldots, F_m\}$ として，このうちの F_1 によって構成された $\mathrm{cov}(F_1)$ が他のクラスタ $\mathrm{cov}(F_2), \ldots, \mathrm{cov}(F_m)$ とまったく重複しないならば，$\mathrm{cov}(F_1)$ 中の文書の $f_x(R, d_i)$ は常に 1 なので（F_1 自身のみがカウントされる），結果的に，$\mathrm{overlap}(F_1|R)$ は 0 になる。つまり，$\mathrm{overlap}(F|R)$ の値が小さいほど，F は，文書クラスタを構成する語集合として望ましいと解釈できる。

FTC アルゴリズム

(1) $\Phi_F=\phi$ と初期化し，θ_n を決めて，Ω_F を算出し，これを R に代入する $(R\leftarrow\Omega_F)$。

(2) Φ_F 中の各 F が構成するクラスタ $\mathrm{cov}(F)$ のいずれかに，すべての文書が含まれれば，処理を終了する。そうでなければ次に進む。

(3) R に含まれるすべての F に対して，$\mathrm{overlap}(F|R)$ を計算し，その値が最小である F を求める（これを F_b とする）。

(4) F_b を Φ_F に加えるとともに，F_b を R から削除する（Φ_F と R の更新）。

(5) F_b が構成するクラスタ $\mathrm{cov}(F_b)$ に含まれる

すべての文書を，R に含まれるすべての F が構成するクラスタ $\mathrm{cov}(F)$ から削除し，(2) に戻る。

以上の処理によって，重複しないクラスタ群 $\mathrm{cov}(F_k)$ $(k=1,\ldots,L)$ が得られるとともに，その内容の記述が F_k として与えられることになる。ただし，段階 (5) で，F_b 中の文書を削除しなければ，非排他的なクラスタリングとなる。

Ω_F には，1 つの語から構成される F のほかに，複数の語から構成される F も含まれている。したがって，仮に $F_1=\{t_1\}$, $F_2=\{t_1, t_2\}$ とすれば，$\mathrm{cov}(F_2)\subseteq\mathrm{cov}(F_1)$ という関係が成立することになる。Ω_F の要素におけるこの種の関係を利用して階層的なクラスタリングを構成する方法は，HFTC (hierarchical frequency term-based clustering) と呼ばれる (Beil ほか (2002)[5])。なお，この場合には，$\mathrm{cov}(F_1)$ が親クラスタ，$\mathrm{cov}(F_2)$ が子クラスタということになる。また，もし，$F_3=\{t_1, t_3\}$ ならば，$\mathrm{cov}(F_3)$ は $\mathrm{cov}(F_1)$ の子で，なおかつ $\mathrm{cov}(F_2)$ の兄弟に相当する。

3. FIHC アルゴリズム

Fung ら (2003)[27] による FIHC (frequent itemset-based hierarchical clustering) もまた，同様な考えに基づいたアルゴリズムである。このアルゴリズムでは，基本的に，“cluster support” と “global support” との差によって，各文書が各クラスタに属する程度 $s(d_i, C_k)$ を定義する。ここで，“cluster support” とは，$F(\in\Omega_F)$ 中の要素 (すなわち 1 つの語 t_j) が，あるクラスタ C_k 中の文書中に出現する割合を意味する。これを cs_j と表記すれば，

$$\mathrm{cs}_j(C_k)=\frac{1}{\tilde{n}_k}\sum_{i:\,d_i\in C_k} b_{ij} \tag{36}$$

である。ここで，b_{ij} は文書 d_i 中に語 t_j が出現すれば 1，そうでなければ 0 を示す変数である。一方，$t_j\in F$ についての “global support” を gs_j と書くことにすると，

$$\mathrm{gs}_j=\frac{|\mathrm{cov}(F)|}{N} \tag{37}$$

である。

閾値 θ_n を定めて F を規定したのと同様に，“cluster support” に対しても，閾値 θ_c を決めて，それを超える語を “cluster frequent” として取り扱う。ある 1 つの文書 d_i における “cluster frequent” の集合を γ_i と表記する。一方，いずれかの F $(\in\Omega_F)$ に含まれる語の集合を $\Omega'(\subseteq\Omega)$ と書く。結局，各文書が各クラスタに属する程度 $s(d_i, C_k)$ は，

$$s(d_i, C_k)=\sum_{j:\,t_j\in\gamma_i\cap\Omega'} x_{ij}\times\mathrm{cs}_j(C_k) - \sum_{j:\,t_j\in\overline{\gamma_i}\cap\Omega'} x_{ij}\times\mathrm{gs}_j \tag{38}$$

で定義される。ここで，$\overline{\gamma_i}$ は，文書 d_i 中での，“cluster frequent” でない語の集合を意味する。つまり，文書 d_i 中に含まれる語が，そのクラスタ中の他の文書にも出現するほど，$s(d_i, C_k)$ の値が大きくなる。また，右辺第 2 項により，文書 d_i における “cluster frequent” でない語 t_j $(\in\Omega')$ に対しては，一種のペナルティが課せられる。

FTC と同様に，FIHC でもまず，$\mathrm{cov}(F)$ として，初期的なクラスタ群を求め，次に，文書ごとに各クラスタの $s(d_i, C_k)$ の値を計算し，その最も高いクラスタのみにその文書を所属させる。これによって排他的なクラスタリングが得られる。そして，やはり，FTC と同様に，各 F が単数または複数の語から構成されることを利用して，クラスタ群を階層的に構成する。なお，FIHC アルゴリズムでは，同一のトピックが異なるクラスタに分散することを防ぐため，事後的に，クラスタの併合が試みられる（詳細は元の文献を参照）。

F. 文書構造の視覚化のための技法

1. VIBE

VIBE (VIsualization By Example) システム (Olsen ほか (1993)[66] および Korfhage (1997)[52]) は，利用者の関心に合わせて，文書集合を視覚的に提示するためのシステムである。文書クラスタリングがその主目的ではないが，利用者が入力した主題概念に応じて，結果的に文書がクラスタ化されることになる。

例えば，ある利用者が “document retrieval”, “scientific visualization”, “virtual reality” の3つの概念に関心を持ったとする。これらをそれぞれ POI (point of interest) と呼ぶ。そして，利用者が各 POI の位置をコンピュータの画面上に指定すると，それぞれの POI の座標が決まる。もし，ある文書が1つの POI（例えば “document retrieval”）に含まれる語のみしか持たないならば，その文書はその POI の位置に重ね合わせて表示される。

一方，POI の2つの語（例えば，“document” と “virtual”）を持つ語の場合には，文書中でのそれらの重みが反映されるように，2つの POI を結ぶ線上に，その文書の位置が決められる。POI の語が3つ以上出現する場合でも，同様な方法で，その文書の表示座標が計算される。

2. GUI への SOM の応用

すでに詳しく説明した，Kohonen の SOM は，文書構造を視覚化するための有用なツールである。その特徴は，長方形の出力領域に，主題ごとに矩形の区域を設定できることにある（Lin (1997)[56]）。Java アプレットなどを使って，各区域を色分けすれば，美しい「文書の地図」を表示することが可能である。

3. SPIRE プロジェクト

SPIRE (spatial paradigm for information retrieval and exploration) は，テキストを視覚化する方法の研究を目的として，1994年に開始されたプロジェクトである (Wise (1999)[94])。同年には最初のソフトウェアである Galaxies が作成され，続いて，ThemeSpaces が発表された。Galaxies は，宇宙空間に散らばる星を模倣して，文書を提示するシステムであり，一方，ThemeSpaces では，地形の立体的な景観図 (landscape) として文書構造が表現される。

これらのシステムの処理手順の概要は以下のとおりである (Wise (1999)[94])。

(1) 文書集合に対して，前処理を施し，各文書をベクトルとして表す。
(2) ベクトルを標準化し，高次元空間中で文書をクラスタ化する。
(3) 高次元空間におけるベクトルとクラスタの重心とを2次元平面上に射影する。
(4) Galaxies と ThemeSpaces のそれぞれの方法に従って，段階 (3) の結果を表示する。

段階 (1) では，ストップワードの除去や語幹抽出が行われる。そしてさらに，文書集合に含まれる語数は一般にかなり多いため，次元を減らす工夫が加えられる。プロジェクトの初期の頃は，ニューラルネットワークが使用されていたようであるが，その後，語の出現頻度と CCV (condensed clustering value) に基づく独自の方法が開発された (Wise (1999)[94])。

まず，低頻度および高頻度の語を除去し，次に CCV を計算する。CCV は語の出現のでたらめさ (randomness) を測定する指標であり，この値が小さいほど，主題を表す語であると解釈される。Wise (1999)[94] では，CCV の値がある閾値よりも小さい語のみを残すことによって，語の数を約20,000から500程度にまで減らした実験が紹介されている。

以上の手順により絞り込まれた語を基本として各文書のベクトルが構成され，この結果，高次元空間に文書が布置されることになる。次に，この空間中で文書のクラスタリングを行う（上記の段階 (2)）。文書はユークリッド空間中に布置されているので，k-means 法や階層的クラスタ分析法を適用できる。また，SPIRE では，“fast divisive clustering” という独自の方法も検討されている。これは，あらかじめ決められた数の種子点を空間中に無作為に分布させ，その種子点を中心とする超球を互いに重複しないように反復的に構成していく方法であり，最終的に，各超球に含まれる文書群をそれぞれクラスタとみなす。

高次元空間でのクラスタ化が終了した後，それを2次元空間に布置する（上記の段階 (3)）。もし文書数が少ないならば，これには一般的な多次元尺度構成法 (MDS) を利用することができる。また，文書数が多い場合のために，“anchored least stress (ALS)” という独自の方法も開発されてい

る。

最後に，MDS または ALS の結果を表示する。Galaxies の場合には，単に，各文書を緑色の点，クラスタの重心をオレンジ色の点で表すだけである。ThemeSpaces の場合には，Galaxies の平面的な表現を基本として，各クラスタを表現する語の選択・表示と，立体的な地形として文書数を表現する工夫とが加えられる。

4. NIRVE

NIRVE (The NIST information retrieval visualization engine) (Cugini ほか (1997)[18]) は，米国の NIST で開発されたインタフェースであり，検索結果としての文書集合を 3 次元で視覚的に表示する機能を持つ。その特徴は，検索質問に含まれる「概念 (concept)」を表現する「語（keyword)」を利用者に特定してもらい，それらの概念に基づいて，文書集合をクラスタに分割する点にある（いわゆる適応型クラスタリング)。

利用者が m 個の概念を指定したとすれば，各文書はまず，その概念に対する重みに基づいて，m 次元空間中に布置される。ある概念に対して利用者が指定した語 t_j が文書 d_i 中に出現する頻度を x_{ij} として，d_i 中の t_j の重みは，$w_{ij}=\sqrt{x_{ij}/l_i}$ で定義される。ここで，l_i は文書 d_i の長さ（延べ語数）である。この w_{ij} に基づいて各概念の重みが計算され，その値によって，m 次元概念空間中での各文書の位置が決まる。

概念空間中に布置された文書間の類似度は通常のユークリッド距離によって測定できる。この距離に基づいて，文書のクラスタリングを行うが，その方法は単純である。すなわち，1 つの文書から出発し，最も近い文書をつないでいく。もし，その距離が閾値を超えれば，そこまでつながった文書を 1 つのクラスタとみなす。

最後に，文書やクラスタを示すアイコンを，3 次元空間に布置する。それらの位置は，クラスタ中の文書間の距離やクラスタ間の距離が，概念空間中のそれらに比例するように決められる。ただし，これは多次元空間中の分布をより少ない次元で表現しようとする試みであり，当然，概念空間中の距離がすべて 3 次元空間中に正確に反映される保証はない。

5. その他の関連研究

本節の最後として，文書クラスタリングを直接的には含まないものの，何らかのかたちで文書やその構造の視覚化を試みている研究を 2 件ほど挙げておく。

TOPIC ISLANDS（Miller ほか (1998)[59]）では，1 件の文書内の内容的な構造を解析するために Wavelet 変換を利用する。その結果として，1 件の文書に含まれるいくつかの主題的な構成要素（主題的なまとまり）が識別され，それらが島の形状を模倣して表現される。その 2 次元空間における各構成要素の位置は，多次元尺度構成法によって計算される。

一方，ThemeRiver (Havre ほか (2002)[33]) は，ある文書集合をその出版年の時間軸に沿って整理・視覚化するシステムであり，各主題が各時点でどの程度研究されているかを川の流れを模倣した表現によって利用者に提示する。具体的には，各主題を示すいくつかの語を手がかりに，各時点での文書数を集計し，そのデータに基づいて，スプライン関数を使って川の流れに似た曲線を描く。

G. 文書クラスタリング技法の総括

以上，本章で議論してきた文書クラスタリングのための技法あるいはアルゴリズムを第 1 表に総括する。第 1 表は，F 節の「GUI への応用」を除いた，A 節から E 節までに議論した技法のうち主なものについて，その特徴と入力パラメータをまとめたものである。

まず，単一パス・アルゴリズムは，おおよそ，

1. k-means 法の拡張
2. leader-follower 法の拡張
3. Kohonen の SOM の応用 (WEBSOM)
4. その他

に分類できる（第 1 表参照)。

k-means 法は，クラスタの個数を先験的に与えなければならないが，この個数がそれほど大きくなければ，かなり大きな文書集合に対して適用可能である。第1表に示したとおり，この k-means 法についてさまざまな点での改良・拡張が試みられている。例えば，各クラスタのよりよい重心を

第1表 文書クラスタリング技法の総括（主なもの）

A. 単一パス・アルゴリズム

名 称	特 徴	入力パラメータ	主な典拠
単純な k-means 法	$O(N)$ 程度の計算量でクラスタリングが可能	クラスタ個数	—
Willett 法	k-means 法の変種で，クラスタ個数を自動決定	反復計算の回数	Willett (1980)[91]
平均クラスタリング法	分割を自動修正する incremental k-means 法	クラスタ個数，基準関数に対する閾値	Dhillon ら (1994)[21]
SKWIC 法	語の最適重みも同時に推計する k-means 法	クラスタ個数	Frigui と Nasraoui (2004)[26]
Scatter/Gather	k-means 法の変種で，階層的クラスタリングによって中心を設定	クラスタ個数	Cutting ら (1992)[19]
C^3M	k-medoid 法の変種で，種子点の数を自動的に算出	なし	Can と Ozkarahan (1984)[11]
単純な leader-follower 法	クラスタ個数が未知の場合に適用可能	文書・クラスタ間の類似度の閾値	岸田 (2003)[46]
Crouch 法	1回目のパスでクラスタを設定し，2回目で文書を割り当てる leader-follower 法	文書・クラスタ間の類似度の閾値	Crouch (1975)[17]
WEBSOM	Kohonen の SOM の応用	出力ベクトルの個数（出力領域）等	Kohonen ら (2000)[50]

B. 階層的クラスタリング

名 称	特 徴	入力パラメータ	主な典拠
Willett 法	転置ファイルを利用して計算量を軽減	なし	Willett (1981)[92]
Voorhees 法	樹形図に取り込まれていない文書のうち，すでに取り込まれたものと最大の類似度を持つものを探索	なし	Voorhees (1986)[88]
bisecting k-means 法	k-means 法を使って文書集合を反復的に分割	なし	Steinbach ら (2001)[81]
制限された凝集型法	最初に k-means 法で文書集合を分割し，それぞれに対して凝集型アルゴリズムを適用	初期的なクラスタ個数	Zhao と Karypis (2002)[100]
PDDP 法	主成分分析を使って文書集合を反復的に分割	なし	Boley ら (1999)[6]

C. 次元縮約法に基づくクラスタリング

名 称	特 徴	入力パラメータ	主な典拠
IRR 法	LSI において影響力の小さなクラスタを識別する	スケール係数等	Ando (2000)[2]
COV-rescale 法	IRR 法を改良し，主成分分析によってクラスタを識別	スケール係数等	Kobayashi と Aono (2004)[47]
NMF 法	語×文書の重み行列を直交しない空間に分解	クラスタ個数	Xu ら (2003)[96]

D. 確率モデルに基づく方法

名　称	特　徴	入力パラメータ	主な典拠
GMM＋EM 法	ガウス型混合モデルと EM アルゴリズムを利用	クラスタ個数等	Liu ら (2002)[58]

E. データマイニングの手法の応用

名　称	特　徴	入力パラメータ	主な典拠
Boley らの方法	関連ルールに基づく一種のグラフ理論によるクラスタリング	なし	Boley ら (1999)[6]
FTC 法	頻出語の出現パターンに基づくクラスタリング	頻出語を決めるための閾値	Beil ら (2002)[5]
FIHC 法	頻出語の出現パターンに基づくクラスタリング	頻出語を決めるための閾値等	Fung ら (2003)[27]

求めようとするもの（平均クラスタリング法，Scatter/Gather など），キーワードの重みも同時に最適化するもの（SKWIC 法），クラスタの個数を自動的に決定するもの（C^3M）などが提案されている。

一方，leader-follower 法ではクラスタ個数を与える必要はないが，文書とクラスタとの類似度の閾値を前もって設定しておかなければならない。この場合，その閾値の与え方によってはクラスタの個数が膨大なものとなり，多くの計算量が必要になる可能性がある（文書とクラスタとの類似度の比較回数やクラスタベクトルの保存領域が増大する）。

WEBSOM は，これらとは異なる原理に基づく文書クラスタリングの実現を可能にするもので，2 次元の出力領域に矩形によってクラスタが表示される点に特徴がある。ただし，この図を計算するために，一定回数の反復計算が必要となり，非常に大規模な文書集合に対して応用が試みられているものの，その計算量はかなり多い。

次に，階層的クラスタ分析法については，凝集型の場合，やはり文書集合が大きいと，その実行は難しい。その中で，第 1 段階で k-means 法を応用し，それによって得られたクラスタごとに凝集型アルゴリズムを適用する方法（「制限された凝集型アルゴリズム」）は，実行可能性という点で，有望であると思われる。bisecting k-means 法などの分割型も計算量の点では優れており（後述），今後，どちらの方法がより妥当であるかを多角的に比較研究していくことが重要である。

次元縮約に基づくクラスリングについては，SVD の場合には，語数 M を減らすことができたとしても，N は不変であるから，計算量の点で問題が生じる。一方，主成分分析の場合には，$M \times M$ の行列が入力データとなるので，特徴抽出で語数を減らすことができれば，実行可能性は高くなる。さらに，主成分分析は一般的によく利用されている多変量解析法であり，この意味で，「文書集合を主題的に均質なグループに分ける」という性能自体も，SVD を上回ることが予想される（この点での SVD と主成分分析の特徴の相違については，Kobayashi と Aono (2004)[47] が詳しい）。

最後に，確率モデルに基づく方法とデータマイニングの手法の応用については，さらなる研究結果の蓄積が必要であると考えられる。直観的には，確率モデルは，計算の簡便性の観点からは正規（ガウス）分布に依拠しなければならないことと，EM アルゴリズムのような近似的な方法でパラメータを推定しなければならないという点から，文書クラスタリングにはそれほど向かないように思われる。一方，データマイニングの代表的な手法である関連ルールを応用する方法は，伝統的な文書クラスタリング研究にはなかった発想であり，その長所・短所を今後探究していく必要があろう。

IV. 文書クラスタリングの研究課題

A. 計算量の問題

計算量の問題については本稿ですでに何度も述べてきた。一般的には，計算機の資源が最も少なくて済むのがk-means法であり，その対極には，凝集型の階層的アルゴリズムがある。そこで，計算量の問題に対するアプローチとしては，大まかにいって，

- k-means法を基本に据えて，その改良を目指すもの
- 階層的アルゴリズムの効率化を目指すもの

の2つの異なる方向性を考えることができる。

k-means法は，基本的には，単一パス・アルゴリズムなので，計算量は$O(NLMr)$程度で済むが(第III章A節参照)，クラスタの個数Lを先験的に与えなければならず，そのため，当該文書集合にとって「自然な」分割とならない可能性がある。この点を補う方法として，Willettの方法やleader-follower法の適用が試みられているが，ともに，$L \approx N$が成立してしまう可能性があり(Willettの方法では最初に$L=M$と設定して，次第にクラスタを減らしていくが，一般に，$N<M$であり，減少の程度にもよるが，Lはかなり大きいと予想される)，計算量は$O(N^2Mr)$に近づいてしまう。

分割型の階層的アルゴリズムについては，その妥当性に関する研究結果の蓄積はまだ十分ではないが，計算量の点だけからいえば，大きな望みがあるように思われる。もし完全に平衡な2分木を構成した場合，走査する文書の延べ総数は，$N \log N$である(第III章B節参照)。例えば，その2分割に計算量の少ないk-means法を使うならば(すなわち，bisecting k-means法)，各文書ごとに，2個のM次元クラスタベクトルとの比較が必要なので，計算量はおおよそ$O(2N \log NMr)$となる。したがって，$2 \log N \approx L$ならば，十分に，一般のk-means法の計算量に比肩しうる。しかも，bisecting k-means法ならば，情報検索という応用目的にとって有用な，クラスタ間の階層構造をも得ることができる。

もし，一般的によく使用され，実績のある群平均法などの凝集型アルゴリズムのほうが，妥当性の観点からは望ましいとするならば，現時点では，「制限された凝集型アルゴリズム」を使うほかはないだろう。簡単のため，文書ベクトルの次元Mと反復回数rとを無視することとし，さらに，n件の文書に対する凝集型アルゴリズムの計算量を$O(n^2)$と仮定すれば，制限された凝集型のアルゴリズムの計算量は，

$$O(NL)+L \times O(n^2) \tag{39}$$

で近似される。ここで，nは，制限された凝集型アルゴリズムにおける第1段階での各分割に含まれる文書数である(すなわち，$n=N/L$)。ただし，実際には，$n=N/L$が成立するとは限らない。つまり，第1段階のk-means法によって，N件の文書が均等な大きさの部分集合に分割できるという保証はない。したがって，式(39)および以下の議論は，「第1段階のk-means法で均等な大きさの部分集合が得られる」という仮定の下でのみ妥当である。また，凝集型アルゴリズムの計算量は一般に$O(n^2)$よりも大きくなるから，式(39)は過小評価となっている点にも注意する必要がある。実際には，単連結法や完全連結法の場合で，$O(n^2 \log n)$程度の計算量が必要である(Jainほか(1999)[39])。

以上の仮定の下に，式(39)からは，第2段階における計算量を減らすためにNに対するnを小さくしても，第1段階のk-means法に多くの計算量が必要となり，必ずしもうまくいかないことがわかる。例えば，$N=1{,}000{,}000$の場合に，$L=100$とすれば，第1段階での計算量は$L \times N=10^8$に比例し，第2段階の$L \times n^2=10^{10}$に比べて，かなり小さい。それに対して，$L=10{,}000$とした場合には，第2段階は$L \times n^2=10^8$のように小さくなるが，その分，第1段階は，$L \times N=10^{10}$となってしまい，結果的に全体の計算量は変わらない。

最適な分割数$\tilde{L}$を求めるには，$n=N/L$の関係を使って$y=NL+L(N/L)^2=NL+N^2/L$とし，こ

れをLで微分して0とおけばよい（LとNを近似的に連続変量とみなす）。すなわち，

$$\frac{dy}{dL}=N-N^2/L^2=0$$

より，$\tilde{L}=\sqrt{N}$を得る。したがって，制限された凝集型アルゴリズムの最小の計算量は，ここでの仮定の下では，$N\sqrt{N}+\sqrt{N}\times N^2/\sqrt{N^2}=2N\sqrt{N}$となる。一方，$M$と$r$を無視した場合のbisecting k-means法の計算量は，$O(2N\log N)$であるから，bisecting k-means法の計算量に対する「制限された凝集型アルゴリズム」の計算量の比$R_c(N)$は，ここでの仮定の下に，

$$R_c(N)=\frac{2N\sqrt{N}}{2N\log N}=\frac{\sqrt{N}}{\log N}$$

によって近似される。この式の値を計算してみると，$R_c(10^3)=3.2$，$R_c(10^4)=7.5$，$R_c(10^5)=19.0$，$R_c(10^6)=50.2$，$R_c(10^7)=136.0$などとなり，文書集合が大きくなるにつれて，制限された凝集型アルゴリズムの計算量は，分割型のbisecting k-means法に比べて，次第に大きくなってしまう（例えば，100万件の文書で約50倍程度）。このことから，凝集型アルゴリズムを大規模文書集合に適用するのがいかに難しいかがわかる。なお，上で注意したように，この計算は，凝集型アルゴリズムの計算量を過小評価しており，実際には，その実行にはより多くの計算が必要になる。

B. 特徴抽出

計算量の問題を解決する1つの有力な方法は，文書ベクトルおよびクラスタベクトルの次元Mを減らす（すなわち語を減らす）ことである。これによって，類似度の計算が速くなるだけでなく，データを保存するための主記憶装置の容量も節約できる。さらに，階層的クラスタ分析法における類似度行列の計算に転置索引ファイルを用いる手法においては（第III章B節参照），語を減らせばそれだけ語を共有する文書の組数が少なくなるので，抜本的な計算量の改善につながる可能性がある（岸田(2003)[46]）。

第III章で概観したように，有用な語（または概念）を選択するための特徴抽出の方法には以下のようなものがある。

1. 何らかの語の重みに従って，語を選択する方法
2. LSIを応用する方法
3. WEBSOMで使用されている次元縮約法(23)式

実際には，上記1.を使用する研究例が多く，前章で見たように，さまざまな方法が考案されている。一方，LSIを応用する方法は，文書数Nが大きい場合には，SVDの計算が難しいという欠点がある。WEBSOMで提案されている式(23)は，LSIに比べれば実装が容易であり，実用的といえるが，研究例が少なく，現時点ではその有効性の評価は難しい。

情報検索の理論に照らせば，ストップワードや非専門的な用語を文書ベクトルから除去するのは当然であろう。特に，情報検索のベクトル空間モデルや確率型モデルなどに基づく照合関数を使用して文書やクラスタの間の類似度を測定する場合には，idfの要因が小さな（つまり非専門的な）語を除去しても，クラスタリングの結果に大きな影響は与えないと見るのは妥当であると思われる。

ただし，理論的には，どの程度まで語を削除してよいのかを決めるのは難しいようである。諸研究の結果を概観した限りでは，かなりの数を減らせるという印象はあるものの，さらなる経験的な研究結果の蓄積が必要であろう。

C. 重みと類似度の計算方法

抽出された語の重みをいかに設定し，文書ベクトルを構成するか，さらには，2つの文書ベクトル間あるいは文書ベクトルとクラスタベクトル間の類似度をいかに計算するかは，文書クラスタリングの結果を大きく左右する重要な問題である。

前章で見たように，文書ベクトル中の重みw_{ij}あるいはクラスタベクトル中の重み$\tilde{w}_{kj}$の設定には，さまざまな方法が使用されている。しかし，全体的な傾向として，以前の単純な計算方法に比べて，最近では，情報検索理論（ベクトル空間，

確率型など）に基づいて，複雑な式により重みを計算する場合が多くなっているようである。一方，類似度の計算については，重複係数やDice係数が使われることもあるが，余弦係数の適用例が多く，また語の重みの計算に情報検索の理論を適用した場合には，その理論に沿ったかたちで類似度の計算がなされることもある。

さまざまな重みの設定方法あるいは類似度の計算方法のうちどれが優れているかという点についての比較研究は少なく，今後，これについて探究を進めていく必要がある。

D. クラスタリングの結果の評価

クラスタリングの結果の妥当性の評価には，次のような類型が考えられる。

1. 直接的な評価
   (a) 外部的な基準との評価（「正解」を使った評価）
   (b) 内部的な評価
2. 間接的な評価

各文書に分類記号が付与されているような「正解付き」の文書集合が利用できれば，クラスタリングの結果を直接的に，その「正解」という外部基準に照らして，評価することが可能である（上記1(a)）。機械学習型のテキスト分類の実験では，通常，この種の文書集合が使用されている。「教師なし」の文書クラスタリングの場合には，分類作業において「正解」をまったく使用せず，最後の評価の段階でのみ，正解情報を利用することになる。「正解」のような外部的な基準が利用可能なときには，評価指標として，

1. F尺度 (F measure)
2. エントロピーまたは相互情報量
3. 正確性 (accuracy)

などを使うことができる。

F尺度は情報検索における伝統的な評価指標であり，再現率と精度との調和平均として計算される。ここでは，「正解」にはH個の分類記号が含まれるとし，それぞれの分類記号が付与された文書集合を$K_1, \ldots, K_H$と書く。また，正解集合K_hに含まれる文書数およびクラスタC_kに含まれる文書数をそれぞれ，$\tilde{n}_h$, $\tilde{n}_k$で表記し，$\tilde{n}_{hk}$は，K_hとC_kとに共通の文書の数を表すものとする。このとき，ある正解集合K_hとあるクラスタC_kが与えられた場合の再現率R_eおよび精度P_rは，

$$R_e(K_h, C_k) = \frac{\tilde{n}_{hk}}{\tilde{n}_h}$$

$$P_r(K_h, C_k) = \frac{\tilde{n}_{hk}}{\tilde{n}_k}$$

となる。F尺度はこれらの調和平均，

$$F_m(K_h, C_k) = \frac{2 \times R_e(K_h, C_k) \times P_r(K_h, C_k)}{R_e(K_h, C_k) + P_r(K_h, C_k)}$$

として求められる。ただし，この値は，ある特定の正解集合と特定クラスタの組ごとに計算されるので，クラスタリングの結果全体を評価するには，

$$F_s = \sum_{h=1}^{H} \frac{\tilde{n}_h}{N} \max_k F_m(K_h, C_k)$$

という指標（Larsen と Aone (1999)[55]）が使用される（FScore, overall F-measure などと呼ばれている）。この指標は，分類記号ごとにF尺度が最大になるクラスタだけを取り出して，その値を，文書数で重み付け平均したものである。

一方，正解集合K_hにクラスタC_kの文書が属する確率$P(K_h|C_k)$を考えれば，各クラスタのエントロピーを，

$$E_k = -\sum_{h=1}^{H} P(K_h|C_k) \log P(K_h|C_k)$$

のように定義できる。確率$P(K_h|C_k)$については，$P(K_h|C_k) = P_r(K_h, C_k)$で推定すればよい。もし，クラスタ$C_k$中の文書がさまざまな正解集合に属するならば，エントロピーE_kの値は増加する。したがって，E_kの値が小さいほど，望ましいクラスタリングであると解釈できる。クラスタリングの結果全体に対しては，クラスタに関する全エントロピー$E(C|K)$を，文書数による重み付け平均として，

$$E(C|K)=\sum_{k=1}^{L}\frac{\tilde{n}_k}{N}E_k$$

で定義すればよい（Steinbach ほか (2000)[81]）。

同様な考え方に基づいて，相互情報量によって，クラスタリングの結果を評価することもできる。つまり，文書集合全体から1件の文書を選んだときに，それが正解集合 K_h とクラスタ C_k の両方に属する確率 $P(K_h, C_k)$ を考えて（この確率は $\tilde{n}_{hk}/N$ で推計できる），正解集合とクラスタとの相互情報量 $M_I(C, K)$ を

$$M_I(C,K)=\sum_{h=1}^{H}\sum_{k=1}^{L}P(K_h,C_k)\log\frac{P(K_h,C_k)}{P(K_h)P(C_k)}$$

で定義すればよい（Liu ほか (2002)[58]）。ここで，$P(K_h)$ と $P(C_k)$ はそれぞれ $\tilde{n}_h/N, \tilde{n}_k/N$ で推定できる。この $M_I(C,K)$ の値は，正解集合とクラスタとが完全に独立ならば0，完全に一致すれば $\max(E(C), E(K))$ となる。ここで $E(C)$ はエントロピーで，$E(C)=-\sum_k P(C_k)\log P(C_k)$ である（$E(K)$ も同様）。したがって，最終的に，

$$\hat{M}_I(C,K)=\frac{M_I(C,K)}{\max(E(C),E(K))}$$

とすれば，0から1の値をとる評価指標となる。

もし，正解集合 K_h とクラスタ C_k との1対1の対応づけが容易ならば，より単純な指標である「正確性 (accuracy)」を使うことができる（ただし，正解集合 $K_1,\ldots,K_H$ も，クラスタ $C_1,\ldots,C_L$ もともに排他的で，$H=L$ でなければならない）。1つの文書 d_i に対して1つのクラスタが決定されたとき，もし，そのクラスタ C_k が，その文書の属する正解集合 K_h に対応したものであるならば，「正しい」分類が行われたと判断できる。したがって，N 件の文書中，何件が「正しく」分類されたかの割合を計算すればクラスタリングを評価できるわけで，この割合を「正確性 (accuracy)」と呼ぶ（Liu ほか (2002)[58]）。実際には，TDT における Tracking のような場合を除けば，C_k と K_h との1対1の対応関係を特定するのは難しいと思われる。

以上の種々の評価指標を使うには，正解集合 $K_1,\ldots,K_H$ が必要である。この集合が利用できない場合には，クラスタ中の文書間に適当な類似度（または非類似度）を設定し，それに基づいて，クラスタがどれだけうまくまとまっているかを評価すればよい。つまり，1つのクラスタ中の文書相互間の類似度が高く，逆に，他のクラスタに含まれる文書との類似度が低いほど，クラスタリングは成功していることになる。これが，上記 1(b) の「内部的評価」に相当する。

この程度を測る指標としては，非類似度（または距離）の場合には，平方誤差の総和 (6) 式などがある（Duda ほか (2001)[22]）。また，この平方誤差の総和は，類似度の場合には，式 (26) にほぼ等しい[22),46]。ただし，これらの指標は，場合によっては，必ずしも適切でないこともありうる（Duda ほか (2001)[22] によい例が掲載されている）。

以上の各種の評価指標は非階層的なクラスタリングの場合にはそのまま使用することができる。一方，階層的な方法の場合には，樹形図の適当な結合レベルで階層を「輪切り」にし，クラスタの集合を設定する必要がある。ただし，F 尺度 F_m の場合には，階層構造のあらゆる結合レベルでのクラスタでの最大値を使えばよいので，特に，特定のレベルで「輪切り」にする必要はない。

最後に，上記 2. の「間接的な評価」とは，例えば，文書クラスタリングの目的が情報検索であるときに，そのクラスタリングの結果を応用した情報検索の実行が，どの程度の検索性能の改善をもたらすかという点から評価するような場合である。この際には，検索実験用のテストコレクションをデータとして使用することができる。実際には，情報検索の性能にはさまざまな要因が働くので，この方法によって，クラスタリングの結果を独立的に評価することは難しい。しかし，クラスタリングの最終的な目的に照らして，総合的に評価できるという点では，優れた方法であるとも考えられる。

情報検索分野における評価研究に比べて，文書クラスタリングの評価の歴史は浅く，標準化はまだ十分でない（すなわち研究者によって使用する指標が異なることが多い）。各指標の特徴の把握，信頼性 (reliability)・頑健性 (robustness) の分析などが，今後必要であると考えられる。

E. 実験による性能比較

文書クラスタリングの研究課題として，上で述べたような評価指標を使って，文書クラスタリングの各技法の妥当性を明らかにしていくことは当然，必要である。しかし，文書集合の規模・性質に関する多様な条件の中で，クラスタリングの妥当性に関する客観的な知見を体系的に積み上げていくことは難しい。ひとつの理想的な形は，TREC[82]やNTCIR[65]などで試みられている，数多くの研究グループが参加する評価型ワークショップかもしれない（TDTはその試みの一例であろう）。前章で概観した研究の中には，複数の手法の比較評価を試みているものもいくつか存在するが，技法の単なる提案に留まっているものも少なくない。研究領域の成熟とともに，次第に，妥当性に関するベンチマークを設定した研究が増えていくことが予想されるが，今後，実験による性能比較の結果を蓄積し，体系的な知見としていくことが必要であろう。

V. おわりに

本稿では，情報検索を応用目的とした文書クラスタリングの技法・アルゴリズムについてレビューし，それらの整理を試みた。また，今後，文書クラスタリングの方法を発展させるための研究課題について，いくつか議論した。

WWWの発展や電子文書の増加に伴って，それらを効果的・効率的に組織化する必要性が高まっている。それに対して，本稿で見たように，現在の計算機環境でも，100万件を超える規模の文書集合を合理的に分割することはまだまだ難しい。デジタル時代の社会的要請に応えていくために，さらに研究を発展させる努力が必要である。

引用文献

1) Anderberg, Michael R. Cluster Analysis for Applications. Academic Press, 1973. (翻訳: 西田英郎ほか. クラスター分析とその応用. 内田老鶴圃, 1988, 442 p.)
2) Ando, Rie Kubota "Latent semantic space: iterative scaling improves precision of inter-document similarity measurement". Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2000, p. 216–223.
3) Ando, Rie Kubota; Lee, Lillian. "Iterative residual rescaling: an analysis and generalization of LSI". Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2001, p. 154–162.
4) Azcarraga, Arnulfo P.; Yap, Teddy N., Jr. "Extracting meaningful labels for WEBSOM text archives". Proceedings of the 2001 ACM CIKM, 2001, p. 41–48.
5) Beil, Florian; Ester, Martin; Xu, Xiaowei. "Frequent term-based text clustering". Proceeding of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002, p. 436–442.
6) Boley, Daniel; Gini, Maria; Gross, Robert; Han, Eui-Hong; Hastings, Kyle; Karypis, George; Kumar, Vipin; Mobasher, Bamshad; Moore, Jerome. Document categorization and query generation on the World Wide Web using WebACE. Artificial Intelligence Review. vol. 13, no. 5/6, 1999, p. 365–391.
7) Bote, Vicente P. Guerrero; Anegón, Felix de Moya; Solana, Victor Herrero. Document organization using Kohonen's algorithm. Information Processing and Management. vol. 38, 2002, p. 79–89.
8) Buckley, C.; Allan, J.; Salton, G. "Automatic routing and ad-hoc retrieval using SMART: TREC2". Proceedings of the Second Text Retrieval Conference (TREC2). D. Harman ed. National Institute of Standards and Technology, 1994, p. 45–55.
9) Can, Fazli. Incremental clustering for dynamic information processing. ACM Transactions on Information Systems. vol. 10, no. 2, 1993, p. 143–164.
10) Can, Fazli; Fox, Edward A.; Snavely, Cory D.; France, Robert K. Incremental clustering for very large document databases: initial MARIAN experience. Information Sciences. vol. 84, 1995, p. 101–114.
11) Can, Fazli; Ozkarahan, Esen A. Two partitioning type clustering algorithms. Journal of the American Society for Information Science. vol. 35, no. 5, 1984, p. 268–276.
12) Can, Fazli; Ozkarahan, Esen A. Similarity and stability analysis of the two partitioning type clustering algorithms. Journal of the American Society for Information Science. vol. 36, no. 1, 1985, p. 3–14.

13) Can, Fazli; Ozkarahan, Esen A. Computation of term/document discrimination values by use of the cover coefficient concept. Journal of the American Society for Information Science. vol. 38, no. 3, 1987, p. 171–183.
14) Can, Fazli; Ozkarahan, Esen A. Concepts and effectiveness of the cover-coefficient-based clustering methodology for text databases. ACM Transactions on Database Systems. vol. 15, no. 4, 1990, p. 483–517.
15) Chen, H.; Hsu, P.; Orwig, R.; Hoopes, L.; Nunamaker, F., Jr. Automatic concept classification of text from electronic meetings. Communications of the ACM. vol. 37, no. 10, 1994, p. 56–73.
16) Croft, W. Bruce. Clustering large files of documents using the single-link method. Journal of the American Society for Information Science. vol. 28, no. 6, 1977, p. 341–344.
17) Crouch, Donald B. A file organization and maintenance procedure for dynamic document collections. Information Processing and Management. vol. 11, no. 1/2, 1975, p. 11–21.
18) Cugini, John; Laskowski, Sharon; Piatko, Christine. "Document clustering in concept space: The NIST Information Retrieval Visualization Engine (NIRVE)". CODATA Euro-American Workshop, 1997.
19) Cutting, Douglass R.; Karger, David R.; Pedersen, Jan O.; Tukey, Joh W. "Scatter/Gather: A cluster-based approach to browsing large document collections". Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1992, p. 318–329.
20) Deerwester, S.; Dumais, S.; Furnas, G.; Landauer, T.; Harshman, R. Indexing by latent semantic analysis. Journal of the American Society for Information Science. vol. 41, no. 6, 1990, p. 391–407.
21) Dhillon, Inderjit; Kogan, Jacob; Nicholas, Charles. "Feature selection and document clustering". Survey of Text Mining. M. W. Berry, ed., Springer, 2004, p. 73–100.
22) Duda, R. O.; Hart, P. E.; Stork, D. G. Pattern Classification. 2nd ed. Wiley, 2001.
23) 江口浩二; 伊藤秀隆; 隈元　昭; 金田彌吉. 漸次的に拡張されたクエリを用いた適応型文書クラスタリング法. 電気情報通信学会論文誌 D-1. vol. J82-D-1, no. 1, 1999, p. 140–149.
24) El-Hamdouchi, Abdelmoula; Willett, Peter. Techniques for the measurement of clustering tendency in document retrieval systems. Journal of Information Science. vol. 13, 1987, p. 361–365.
25) Franz, Martin; McCarley, J. Scott; Ward, Todd; Zhu, Wei-Jing. "Unsupervised and supervised clustering for topic tracking". Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2001, p. 310–317.
26) Frigui, Hichem; Nasraoui, Olfa. "Simultaneous clustering and dynamic keyword weighting for text documents". Survey of Text Mining. M. W. Berry, ed. Springer, 2004, p. 45–72.
27) Fung, Benjamin C. M.; Wang, Ke; Ester, Martin. "Hierarchical document clustering using frequent itemsets". SIAM International Conference on Data Mining, 2003.
28) Golub, G. H.; van Loan, C. F. Matrix Computations, 3rd ed. Johns Hopkins Univ. Press, 1996.
29) Griffith, Alan; Robinson, Lesley A.; Willett, Peter. Hierarchic agglomerative clustering methods for automatic classification. Journal of Documentation. vol. 40, no. 3, 1984, p. 175–205.
30) Han, Jiawei; Kamber, Micheline. Data Mining: Concepts and Techniques. Morgan Kaufmann, 2001.
31) Harding, Alan F.; Willett, Peter. Indexing exhaustivity and the computation of similarity matrices. Journal of the American Society for Information Science. vol. 31, no. 4, 1980, p. 298–300.
32) Hatzivassiloglou, Vasileios; Gravano, Luis; Maganti, Ankineedu. "An investigation of linguistic features and clustering algorithms for topic document clustering". Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2000, p. 224–231.
33) Havre, Susan; Hetzler, Elizabeth; Whitney, Paul; Nowell, Lucy. ThemeRiver: visualizing thematic changes in large document collection. IEEE Transactions on Visualization and Computer Graphics. vol. 8, no. 1, 2002, p. 9–20.
34) Hearst, Marti A.; Pedersen, Jan O. "Reexamining the cluster hypothesis: Scatter/Gather on retrieval results". Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1996, p. 76–84.
35) Hofmann, Thomas. "The cluster-abstraction model: unsupervised learning of topic hierarchies from text data". Proceedings of IJCAI-99, 1999.
36) Honkela, Timo; Kaski, Samuel; Lagus, Krista; Kohonen, Teuvo. "Exploration of full-text

databases with self-organizing maps". Proceedings of ICNN'96, IEEE International Conference on Neural Networks, 1996, p. 56-61.

37) Ishikawa, Yoshiharu; Chen, Yibing.; Kitagawa, Hiroyuki. "An on-line document clustering method based on forgetting factors". Research and Advanced Technology for Digital Libraries: 5th European Conference, ECDL 2001. Panos Constantopoulos and Ingeborg T. Solvberg eds. Springer, LNCS 2163, 2001, p. 325-339.

38) 石岡恒憲. クラスター数を自動的に決定するk-meansアルゴリズムの拡張について. 応用統計学. vol. 29, no. 3, 2000, p. 141-149.

39) Jain, A. K.; Murty, M. N.; Flynn, P. J. Data clustering: a review. ACM Computing Surveys. vol. 31, no. 3, 1999, p. 264-323.

40) Jardin, N.; van Rijsbergen, C. J. The use of hierarchic clustering in information retrieval. Information Storage and Retrieval. vol. 7, no. 5, 1971, p. 217-240.

41) Jones, Gareth; Robertson, Alexander M.; Santimetvivul, Chawchat; Willett, Peter. Non-hierarchic document clustering using a genetic algorithm. Information Research. vol. 1, no. 1, 1995. http://informationr.net/ir/1-1/paper1.html

42) 神嶌敏弘. データマイニング分野のクラスタリング手法 (1). 人工知能学会誌. vol. 18, no. 1, 2003, p. 59-65. およびデータマイニング分野のクラスタリング手法 (2). 人工知能学会誌. vol. 18, no. 2, 2003, p. 170-176.

43) Kaski, Samuel. "Dimensionality reduction by random mapping: fast similarity computation for clustering". Proceedings of IJCNN'98, IEEE International Joint Conference on Neural Networks. vol. 1, 1998, p. 413-418.

44) Kaski, Samuel. "Fast winner search for SOM-based monitoring and retrieval of high-dimensional data". Proceedings of ICANN 99, the 9th International Conference on Neural Networks. vol. 2, 1999, p. 940-945.

45) 岸田和明. 図書館情報学における自動分類と自動索引作成のための統計的手法: 文献レビュー. 日本図書館情報学会誌. vol. 47, no. 1, 2001, p. 17-28.

46) 岸田和明. 大規模文献集合に対して階層的クラスタ分析法を適用するための単連結法アルゴリズム. Library and Information Science. no. 47, 2003, p. 27-38.

47) Kobayashi, Mei; Aono, Masaki. "Vector space models for search and cluster mining". Survey of Text Mining. M.W. Berry ed. Springer, 2004. p. 103-122.

48) Kogan, Jacob. "Means clustering for text data". Proceedings of the Workshop on Text Mining at the First SIAM International Conference on Data Mining. 2001. p. 47-57.

49) Kohonen, T. Self-Organizing Maps. Springer-Verlag, 1995. (翻訳: 徳高平蔵, 岸田悟, 藤村喜久郎. 自己組織化マップ. シュプリンガー・フェアラーク東京, 1996)

50) Kohonen, Teuvo; Kaski, Samuel; Lagus, Krista; Salojarvi, Jarkko; Honkela, Jukka; Paatero, Vesa; Saarela, Antti. Self organization of a massive document collection. IEEE Transactions on Neural Networks. vol. 11, no. 3, 2000, p. 574-585.

51) Kolatch, Erica. "Clustering algorithms for spatial databases: a survey". 2001. http://citeseer.nj.nec.com/436843.html

52) Korfhage, Robert R. Information Storage and Retrieval. John Wiley and Sons, 1997.

53) Lagus, Krista. Text Retrieval Using Self-organized Document Maps. Technical Report A61. Helsinki University of Technology, Laboratory of Computer and Information Science, 2000.

54) Lagus, Krista; Kaski, Samuel. "Keyword selection method for characterizing text document maps". Proceedings of ICANN 99, the 9th International Conference on Neural Networks, 1999, p. 371-376.

55) Larsen, Bjornar; Aone, Chinatsu. "Fast and effective text mining using linear-time document clustering". Proceedings. of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1999, p. 16-22.

56) Lin, Xia. Map displays for information retrieval. Journal of the American Society for Information Science. vol. 48, no. 1, 1997, p. 40-54.

57) Lin, X.; Soergel, D.; Marchionini, G. "A self-organizing sematic map for information retrieval". Proceedings of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 1991, p. 262-269.

58) Liu, Xin; Gong, Yihong; Xu, Wei; Zhu, Shenghuo. "Document clustering with cluster refinement and model selection capabilities". Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 2002, p. 191-198.

59) Miller, Nancy; Wong, Pak Chung; Brewster, Mary; Foote, Harlan. "TOPIC ISLANDS: a wavelet-based text visualization system".

IEEE Visualization '98, 1998, p. 189–196.
60) 宮本定明. クラスター分析入門：ファジィクラスタリングの理論と応用. 森北出版, 1999.
61) Muresan, Gheorghe; Harper, David J. "Document clustering and language models for system-mediated information access". Research and Advanced Technology for Digital Libraries: 5th European Conference, ECDL 2001. Panos Constantopoulos and Ingeborg T. Solvberg eds. Springer, LNCS 2163, 2001, p. 438–449.
62) Murtagh, F. Structure of hierarchic clustering: implications for information retrieval and for multivariate data analysis. Information Processing and Management. vol. 20, no. 5/6, 1984, p. 611–617.
63) Nilsson, Martin. Hierarchical clustering using non-greedy principal direction divisive partitioning. Information Retrieval. vol. 5, 2002, p. 311–321.
64) NIST. Topic Detection and Tracking (TDT). http://www.nist.gov/speech/tests/tdt/ (参照 2004-06-07)
65) NTCIR. NTCIR (NII-NACSIS Test Collection for IR Systems) Project. http://research.nii.ac.jp/ntcir/ (参照 2004-06-07)
66) Olsen, Kai A.; Korfhage, Robert R.; Sochats, Kenneth M.; Spring, Michael B.; Williams James G. Visualization of a document collection: the VIBE system. Information Processing and Management.vol. 29, no. 1, 1993, p. 66–81.
67) Orwig, Richard E.; Chen, Hsinchun; Nunamaker, Jay F., Jr. A graphical, self-organizing approach to classifying electronic meeting output. Journal of the American Society for Information Science. vol. 48, no. 2, 1997, p. 157–170.
68) Pantel, Patrick; Lin, Dekang. "Document clustering with committees". Proceedings. of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 2002, p. 199–206.
69) Papka, Ron; Allan, James. "Topic detection and tracking: event clustering as a basis for first story detection". Advances in Information Retrieval. W. Bruce Croft ed. Kluwer Academic Publishers, 2000. p. 97–126.
70) Rasmussen, Edie. "Clustering algorithms". Information Retrieval: Data Structure and Algorithms. William B. Frakes and Ricardo Baeza-Yates eds. PTR Prentice Hall, 1992. p. 419–442.
71) Roussinov, Dmitri G.; Chen, Hsinchun. Information navigation on the web by clustering and summarizing query results. Information Processing and Management. vol. 37, 2001, p. 789–816.
72) Sahami, Mehran; Yusufali, Salim; Baldonado, Michelle Q. W. "SONIA: A service for organizing networked information autonomously". Proceeding of the third ACM International Conference on Digital Libraries. 1998, p. 200–209.
73) Salton, G.; McGill, M. J. Introduction to Modern Information Retrieval. McGraw-Hill. 1983.
74) Schutze, Hinrich; Silverstein, Craig. "Projections for efficient document clustering". Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 1997, p. 74–81.
75) Sibson, R. SLINK: an optimally efficient algorithm for the single-link cluster method. The Computer Journal. vol. 16, no. 1, 1973, p. 30–34.
76) Silverstein, Craig; Pedersen, Jan O. "Almost-constant-time clustering of arbitrary corpus subsets". Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 1997, p. 60–66.
77) Slonim, Noam; Tishby, Naftali. "Document clustering using word clusters via the information bottleneck method". Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 2000, p. 267–273.
78) Small, Henry. Update on science mapping: creating large documentation space. Scientometrics. vol. 38, no. 2, 1997, p. 275–293.
79) Small, Henry. Visualizing science by citation mapping. Journal of the American Society for Information Science. vol. 59, no. 9, 1999, p. 799–813.
80) Smeaton, Alan F.; Burnett, Mark; Crimmins, Francis; Quinn, Gerard. "An architecture for efficient document clustering and retrieval on a dynamic collection of newspaper texts". 20th BCS-IRGS Colloquium on Information Retrieval. 1998.
81) Steinbach, Michael; Karypis, George; Kumar, Vipin. "A comparison of document clustering techniques". KDD Workshop on Text Mining. 2000.
82) TREC. Text REtrieval Conference (TREC) http://trec.nist.gov/ (参照 2004-06-07)
83) van Hulle, Marc M. Faithful Representations

and Topographic Maps. John Wiley and Sons, 2000.（翻訳：徳高平蔵，藤村喜久郎．自己組織化マップ：理論・設計・応用．海文堂，2001）

84) van Rijsbergen, C. J. Further experiments with hierarchic document clustering in document retrieval. Information Storage and Retrieval. vol. 10, no. 1, 1974, p. 1–14.

85) van Rijsbergen, C. J. Information Retrieval. 2nd ed. Butterworths, 1979.

86) van Rijsbergen, C. J; Croft, W. B. Document clustering: an evaluation of some experiments with the Cranfield 1400 collection. Information Processing and Management. vol. 11, no. 5/7, 1975, p. 171–182.

87) van Rijsbergen, C. J.; Sparck Jones, K. A test for the separation of relevant and non-relevant documents in experimental retrieval collections. Journal of Documentation. vol. 29, no. 3, 1973, p. 251–257.

88) Voorhees, Ellen M. Implementing agglomerative hierarchic clustering algorithms for use in document retrieval. Information Processing and Management. vol. 22, no. 6, 1986, p. 465–476.

89) Wang, Yitong; Kitsuregawa, Masaru. "Evaluating contents-link coupled web page clustering for web search results". Proceedings of the 2002 ACM CIKM. 2002, p. 499–506.

90) Weiss, Sholom; White, Brian F.; Apte, Chidanand V. "Lightweight document clustering". IBM Research Report. No. RC–21684, 2000. http: //www.research.ibm.com/dar/papers/pdf/weiss_ldc_with_cover.pdf

91) Willett, Peter. Document clustering using an inverted file approach. Journal of Information Science. vol. 2, no. 5, 1980, p. 223–231.

92) Willett, Peter. A fast procedure for the calculation of similarity coefficients in automatic classification. Information Processing and Management. vol. 17, 1981, p. 53–60.

93) Willett, Peter. Recent trends in hierarchic document clustering: a critical review. Information Processing and Management. vol. 24, no. 5, 1988, p. 577–597.

94) Wise, James A. The ecological approach to text visualization. Journal of the American Society for Information Science. vol. 50, no. 13, 1999, p. 1224–1233.

95) Xu, J.; Croft, W. B. "Topic-based language models for distributed retrieval". Advances in Information Retrieval: Recent Research from the Center for Intelligence Information Retrieval. W. B. Croft ed. Kluwer, 2000, p. 151–172.

96) Xu, Wei; Liu, Xin; Gong,Yihong. "Document clustering based on non-negative matrix factorization". Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 2003, p. 267–273.

97) Yu, Clement T. A clustering algorithm based on user queries. Journal of the American Society for Information Science. vol. 25, no. 4, 1974, p. 218–226.

98) Zamir, Oren; Etzioni, Oren. "Web document clustering: a feasibility demonstration". Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 1998, p. 45–54.

99) Zamir, Oren; Etzioni, Oren. "Grouper: a dynamic clustering interface to Web search results". The Eighth International World Wide Web Conference. 1999. http://www8.org/w8-papers/

100) Zhao, Ying; Karypis, George. "Evaluation of hierarchical clustering algorithms for document databases". Proceeding of the 2002 ACM CIKM. 2002, p. 515–524.