独立成分分析を用いたテキストデータからのトピック検出

濱本　雅史† 　北川　博之†† 　Jia-Yu Pan††† 　Christos Faloutsos†††

† 筑波大学 システム情報工学研究科　〒305–8573 茨城県つくば市天王台 1–1–1
†† 筑波大学 電子情報工学系　〒305–8573 茨城県つくば市天王台 1–1–1
††† School of Computer Science, Carnegie Mellon Univeristy
E-mail: †hamamoto@kde.is.tsukuba.ac.jp, ††kitagawa@is.tsukuba.ac.jp, †††{jypan,christos}@cs.cmu.edu

あらまし　近年大量の文書の配信や交換がネットワークを介して盛んに行なわれるようになった。今後文書として配信される情報量はますます増加し、文書データから必要な情報の発見が困難になると考えられる。この状況において、文書データの内容を分析しどのような話題が含まれているかを検出することが有用である。このための既存の手法としてはクラスタリングを用いるものが一般的であるが、ここ数年信号処理の分野で発展した独立成分分析を用いる手法が一部で提案されている。しかし、未だ検討が不足しておりその性質の多くは明らかでない。そこで本論文では独立成分分析を用いたトピック検出手法について検討し、実験によりその性質を明らかにする。

キーワード　トピック検出, テキストマイニング, クラスタリング, 知識発見

Topic Detection from Text Data Using Independent Component Analysis

Masafumi HAMAMOTO†, Hiroyuki KITAGAWA††, Jia-Yu PAN†††, and Christos FALOUTSOS†††

† Graduate School of Systems and Information Engineering, University of Tsukuba, Tennohdai 1–1–1, Tsukuba, Ibaraki, 305–8573 Japan
†† Institute of Information Sciences and Electronics, University of Tsukuba, Tennohdai 1–1–1, Tsukuba, Ibaraki, 305–8573 Japan
††† School of Computer Science, Carnegie Mellon Univeristy
E-mail: †hamamoto@kde.is.tsukuba.ac.jp, ††kitagawa@is.tsukuba.ac.jp, †††{jypan,christos}@cs.cmu.edu

Abstract　A lot of electronic documents are distributed via network recently. Their number is increasing rapidly, and it will become difficult to discover important information from document data. In this situation, it is helpful to automatically analyze contents of document data and to detect what subject is included. In existing methods it is common to use clustering, but application of Independent Component Analysis, which was developed in signal processing area, to topic detection was proposed recently. However, its property has not yet been made clear. In this paper we examine the topic detection method using Independent Component Analysis, and show its properties by experimentation.

Key words　Topic Detection, Text Mining, Clustering, Knowledge Discovery

1. はじめに

近年大量の文書の配信や交換がネットワークを介して盛んに行なわれるようになった。特に電子ニュースやチャット、メールマガジンなど、継続的にテキストデータの配信を行なうサービスは増加している。従って今後テキストデータとして配信される情報量はますます増加し、必要な情報の発見が困難になると考えられる。

大量にテキストデータが存在する状況を考えた場合、その内容を分析しどのような話題が含まれているかを検出することが有用である。ここである特定の話題のことをトピックとし、トピック検出とは各トピックに対応する特徴的な単語群とトピックに合致するテキストデータのフラグメントの発見を行なうことと定義する。

本研究で対象とするテキストデータにおけるトピック検出には、論文や書籍などの一般的な文書からのトピック検出と異なる点がある。まず事前にトピックやトピックを特徴付ける単語群を学習しておくことは一般的に不可能である点である。なぜ

なら配信されるテキストデータは一般的な文書よりも新規性が高いことが多く、その内容も多岐に渡ることが多いからである。もう一点はフラグメント間に、タグなどの機械的に区切りとして認識できる情報が与えられるとは限らない点である。よって様々な形態を持つテキストデータを統一的に扱うためには、区切りの有無に依存しない方法を用いる必要がある。ここでは全テキストデータを連結した単語列に対し、ウィンドウと呼ばれる一定語数単位で区切られたフラグメントを1つの文書として扱い、ウィンドウの集合の中からトピックを検出する。

トピック検出は、ウィンドウが分布する空間から特徴的な軸を発見することと考えられる。全テキストデータ中に m 個の語彙が用いられている場合、各ウィンドウは m 次元のベクトルとして表現できる。このとき同じトピックのウィンドウならばウィンドウ中の単語の出現状況は似ており、これらのウィンドウを表すベクトルはあるひとつの特徴的な軸の周りに分布していると考えられる。したがってトピックを検出する問題は特徴軸を発見する問題として捉えることができる。

空間から特徴軸を発見するためのシンプルなアイディアとして、情報検索の分野で潜在的意味インデキシング [4] と呼ばれる、特異値分解を用いる手法が考えられる。しかし一般的にはクラスタリングを用いる手法が検討されており、これをベースにした研究 [11] [12] がなされている。一方近年信号処理の分野で発展した、独立成分分析 [7] を用いる手法 [2] [8] が一部で提案されているが、クラスタリングなどのトピック検出手法との比較検討が不足しており、その性質の多くはまだ明らかになっていない。そこで本研究では独立成分分析を用いたトピック検出手法を、他の手法と実データを用いて定量的に比較することで、その特徴を明らかにする。

本稿は以下のように構成される。2 章において関連研究について述べ、それらと比較した本研究の位置づけを行なう。3 章では対象となるテキストデータからウィンドウを表すベクトルをどう作成するか述べる。4 章では独立成分分析を用いた特徴軸の導出手法を示すとともに、比較対象となる特異値分解による手法およびクラスタリングを用いる手法を示す。5 章ではユーザに提示される、特徴軸に対応する単語群およびウィンドウの選択手法について述べる。6 章で比較実験を行ない各手法の性質を明らかにする。最後にまとめと今後の課題について述べる。

2. 関 連 研 究

トピック検出に関する既存の研究として米 NIST 主催の Topic Detection and Tracking がある(注1)。これはトピック検出とトピック追跡（ユーザにより与えられたトピックがどの記事中に現れるかを提示すること）に関するコンペティション形式のプロジェクトである。この成果として、インクリメンタルなクラスタリングを用いる手法 [12] や単連結の階層的クラスタリングを用いる手法 [11] などが提案された。この他にもこれまでに提案されたトピック検出手法は、多くはクラスタリングをベースにしたものとなっている。一方独立成分分析を用いる手法として、チャットログからのトピック検出の研究がある [2] [8]。これらはテキストデータの時間相関性に着目し、データに特化した独立成分分析アルゴリズムを用いることでトピック検出を行なっている。これに対し、本研究は特徴軸を見つける比較的汎用的なアルゴリズムを用いたトピック検出手法の相互比較および各手法の性質の分析を目的とする。

3. テキストデータの処理

ここでは本研究が対象とするテキストデータについて述べ、それらをどう処理するかについて述べる。本研究におけるトピック検出システムは図 1 にあげたように構成される。このシステムは以下の 3 つの部分からなる。

（1） テキストデータをウィンドウに分割
（2） 特徴軸の抽出
（3） 特徴軸に対応する単語群およびウィンドウの選択

このうち (2) については 4 章、(3) については 5 章で詳しく述べ、本章では (1) について説明する。

テキストデータをウィンドウに分割する理由は、様々な形態をもつテキストデータに対応するためである。現実におけるテキストデータは一様ではない。例としてニュース記事データを考えると、一つのデータ中に含まれる記事数が一つであるとは限らず、複数の場合もある。また各記事毎に送信時刻などのメタデータが与えられる場合もあるが、そうでない場合もある。一方チャットログのようなデータでは記事という単位で区切られていない。このような状況において単一のシステムであらゆる種類のテキストデータの処理を行なうには、すべてのデータを単一の表現形式にする必要がある。しかし一般的には各データがどのような表現形式を取っているのかわからない。そこで各データを最も単純な形式、すなわち記事間の区切りもメタデータも存在しない形式であるとみなす。この場合ファイルなどによるデータ間の区分も意味を持たないので、全データを単純に連結させた一つの単語列を作成する。

作成された単語列から実際にトピックを検出するためには、単語列から記事の区切りを推定することで疑似的に記事を作りだし、疑似的な記事の集合からトピック検出を行なう手法が考えられる [1]。しかしこの場合区切りの推定手法が大きな問題となり得る。そこである一定の語数単位に単語列を分割し（これをウィンドウと呼ぶ）この中からトピックを検出することを考える。こうすることで区切りの推定プロセスを省くことができる。この場合ウィンドウの大きさをどうするのかが問題となるが、この検証は 5 章において行なっている。各ウィンドウの前後関係を保つため、隣り合ったウィンドウは半分程度重ねられる。

ウィンドウへの分割後、ベクトル空間モデル [10] を用いて各ウィンドウをベクトルとして表現する。具体的にはテキストデータ中の語彙（語数 m）中の単語 $w_i(1 \leqq i \leqq m)$ に関し、j 番目のウィンドウのベクトルは、そのウィンドウにどの程度単語 w_i が現れるのかを i 番目の次元の値 x_{ij} として持つベクトルとなる。この x_{ij} は単純に単語の頻度 f_{ij} を取ることもあるが、一部分に極端に同じ単語が出現する場合の影響を抑えるために対数をとった値 $log(1 + f_{ij})$(注2)を取ることも考えられる。

(注1)：http://www.nist.gov/speech/tests/tdt/

(注2)：1 は $t_{ij} = 0$ のときにちょうど 0 となるように加えられている

5 章の実験では後者を用いた。

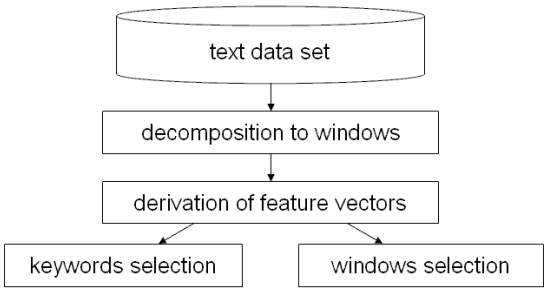

図 1　トピック検出システムの流れ

4.　特徴軸の導出手法

本節では特徴軸を求める 3 つの手法の説明を行なう。以下全テキストデータ中の語彙の単語数を m、全テキストデータから作られたウィンドウの数を n、先頭から i 番目のウィンドウを表すベクトルを $\mathbf{x_i} = (x_{i1}, ..., x_{im})^T$、全ウィンドウを並べた行列を $X = (\mathbf{x_1}, ..., \mathbf{x_n})$ とおく。またベクトルは列ベクトルとする。

4.1　独立成分分析

独立成分分析 [7] は信号処理の分野で発展した、混ざり合った信号から元の信号を復元する手法の一つである。これは源信号が独立に発生するという仮定のもと、k 個の混合信号 $X = (\mathbf{x_1}, ..., \mathbf{x_k})^T$ から源信号 $S = (\mathbf{s_1}, ..., \mathbf{s_k})^T$ と混合係数行列 $A = (\mathbf{a_1}, ..., \mathbf{a_k})$ を推定する方法である。これらの間には $X = AS$ という関係が成り立つ。推定には様々な手法があるが、最も一般的なのが尖度を最大化することで求める手法である。具体的な説明は文献 [6] [7] に示されている。

独立成分分析を用いたトピック検出手法の手順は図 2 に挙げた通りである。第 1 段階は次元削減より予め推定されたトピック数個の次元数にウィンドウを射影する。これは独立成分分析のアルゴリズム自体が信号数の推定を考慮していないからである。通常次元削減には次項で述べる特異値分解で求めた基底を用いる。第 2 段階で実際に独立成分分析を行ない源信号と混合係数行列に分解する。第 3 段階では次元削減で用いた基底および混合係数行列を用いて特徴軸を計算する。

（1） 特異値分解 $X = U\Sigma V$ を計算
（2） 左特異ベクトル U のうち対応する特異値が最大の k 個 U_k を用いて $X_k = U_k^T X$ と次元削減
（3） 独立成分分析により $X_k = AS$ を計算
（4） 特徴軸を $U_k A$ と計算

図 2　独立成分分析を用いた特徴軸の計算

4.2　特異値分解

特徴的な軸を計算する古典的な手法として、特異値分解 (SVD: Singular Value Decomposition) を用いる方法がある。ここで見つかる軸にはデータの分散を最大にするという性質を持っている。具体的には行列を特異値分解して得られる特異ベクトルが特徴軸にあたるというもので、考え方は非常にシンプルであるものの、特徴軸は文書中で共起する単語の関係を表すという性質を持つことが知られている [4]。ただしこれらの軸の間では直交するという制約を持つ。

行列計算として説明すると、特異値分解とは任意の行列 D を 2 つの直交行列 U, V と 1 つの対角行列 Σ を用いて $D = U\Sigma V^T$ という形式に分解することである。U あるいは V の列ベクトルは特異ベクトルと呼ばれ、D の列 (行) ベクトルで張られる空間の正規直交基底となる。ここでは各ウィンドウが列ベクトルであるので U のベクトルを用いる。特異ベクトルのデータを分散させる意味での重要度は、対応する対角行列 Σ の対角成分 (これを特異値と呼ぶ) の値により決定できる。よって k 個の特徴軸は、対応する特異値の最も大きな k 個の特異ベクトルとして求められる。

4.3　クラスタリング

クラスタリングは似ているデータオブジェクト同士を同じグループ (クラスタ) に分類することである。各クラスタの性質を調べることで、データ中にどのような性質を持つオブジェクトがあるのかがわかる。トピック検出として考えると、各ウィンドウをクラスタリングすることで単語の出現状況が似たウィンドウを同じクラスタに分類することができる。このときクラスタ内のウィンドウは同じトピックを表すと仮定すると、各クラスタの重心がトピックを表す特徴軸であるとみなせる。

クラスタリング手法は非常に様々な種類があるが、最も単純な手法と考えられるのが k-means 法 [9] である。また、ウィンドウを表すベクトルの類似度を測るためにはコサイン尺度 (2 つのベクトルの内積を各ベクトルのノルムで割ったもの) を用いる。

一方次元削減を行なった後クラスタリングを行なう手法も考えられる。これは独立成分分析の項で述べたのと同様の手法で次元削減を行ったデータをクラスタリングすることで求められる。具体的には図 3 のようにして計算される。あるクラスタ重心 C_i を次元削減すると $U_k^T C_i$ となる。これが次元削減されたデータをクラスタリングした結果のクラスタ重心となる。このとき削減された次元のクラスタ重心からもとの次元での表現を得るには、特異ベクトルが正規直交基底であるという性質より左側から U_k を掛ければよい。

入力：X=$\{x_1, ..., x_n\}$,　k(クラスタ数)
（1） 特異値分解 $X = U\Sigma V$ を計算
（2） 左特異ベクトル U のうち対応する特異値が最大の k 個 U_k を用いて $X_k = U_k^T X$ と次元削減
（3） X_k に対しクラスタリングを行ないクラスタ重心の集合 C=$\{C_1, ..., C_p\}$ を得る
（4） クラスタ重心 C_i に対応する特徴軸を $U_k C_i$ と計算

図 3　次元削減とクラスタリングを組み合わせた手法

5.　特徴軸に対応する単語・ウィンドウの選択

ここでは前章で述べた特徴軸からどのようにユーザに提示される単語を選び出すか、および各特徴軸に対応するウィンドウ

はどう選択されるかについて述べる。

各特徴軸はベクトルの形で表され、各次元はテキストデータ中の語彙に含まれる一単語と1対1で対応付けられている。この各次元の絶対値は対応する単語の、その特徴軸に対する影響力を表している。従ってユーザが、各特徴軸を表す単語を p 個出力してほしいとシステムに要求した場合、各特徴軸における各次元の絶対値を調べ、値が大きなほうから p 次元に対応する単語を返せばよい。

一方特徴軸に対応するウィンドウを選択するには、単純にコサインが大きなものを選ぶ手法が考えられる。つまり各特徴軸について全ウィンドウとのコサインを調べ、その値の大きなものから順にウィンドウを提示する方法である。特徴軸の導出に独立成分分析を用いた場合この手法の他に、推定された源信号を用いる手法も考えられる。これはある特徴軸について、それを導出する混合係数 $\mathbf{a_i}(1 \leqq i \leqq k, k$ は信号数) に対応する源信号 $\mathbf{s_i}$ を平均0にし、信号の振幅が大きなものほど特徴軸により適合していると考える手法である。

6. 比較実験

本章では、以上に挙げた手法に関して実データを用いた相互比較実験を行なうことで各手法の性質を明らかにする。実験は行列計算ソフト MATLAB を用いて行ない、SVD および k-means については MATLAB 付属のパッケージを用いた。独立成分分析については JADE [3] パッケージを用いた。

実データを用いた評価ではウィンドウの選択の質を測ることは難しい。これは各記事中において単語の出現状況が一様でないからである。従ってここでは特徴軸および特徴軸に対応する単語の質を測ることを目的とする。

6.1 実験データの概要

実験に使用したのは TDT2 [5] のデータである。これは CNN Headline News や New York Times News Service など 6 種類の配信源における 1998 年 1 月から 6 月までのニュース記事を収録したコーパスである。収録されたニュース記事の一部には、トピック付けの情報および記事とトピックとの適合具合 (完全に適合するか一部のみ適合するかの 2 種類) の情報が付加されている。以下に述べる実験 1 と 2 では表 1 に挙げた 20 個のトピックと完全に適合する CNN の記事を、実験 3 では表 1 のうち TP_1 から TP_{10} の 10 個のトピックと完全に適合する、CNN と New York Times の記事を使用した。この各記事に対し不要語の除去と語幹抽出を行い、ランダムに並べた記事を連結させたテキストデータを実験対象とした。ランダムに並べる理由は、この手法が対象とするデータがニュース記事に限るものではないため、各トピックに関する時間相関性の影響を少なくするためである。

またこの実験ではトピック数は何らかの手法により推定できていると仮定する。

6.2 評価手法

本実験において様々な評価手法が考えられるが、ここでは以下に述べる 2 種類の手法を用いて評価を行なった。クラスタリングを用いたトピック検出手法（次元削減した場合も含む）の場合はランダムな要素が含まれるため、手法を 10 回適用したときの評価値の平均および標準偏差を求めた。

Topic ID	トピック名
TP_1	アジア経済危機
TP_2	Monica Lewinsky
TP_3	長野オリンピック
TP_4	対イラク衝突
TP_5	スーパーボウル
TP_6	タバコ会社に対する健康被害訴訟
TP_7	インドの核疑惑
TP_8	イスラエルとパレスチナの対話
TP_9	インドネシアの反スハルト暴動
TP_{10}	爆弾犯 Theodore Kaczynski への判決
TP_{11}	ローマ法王のキューバ訪問
TP_{12}	アラバマ病院爆破事件
TP_{13}	イタリアのケーブルカー事故
TP_{14}	フロリダのトルネード被害
TP_{15}	Oprah Winfrey の狂牛病報道問題
TP_{16}	Gene Mckinney 軍曹の性的不品行事件
TP_{17}	バイアグラ
TP_{18}	Jonesboro での少年の銃乱射事件
TP_{19}	スペースシャトルでの生物実験
TP_{20}	General Motors のストライキ

表 1　使用した記事のトピック

また実行時間について、各手法を 10 回適用したときの平均時間および標準偏差を求めた。

Jaccard 尺度に基づく手法

Jaccard 尺度は 2 つの集合について、どの程度同じ要素を持つかを測る尺度である。具体的に書くと、2 つの集合を X, Y とし集合の要素数を返す関数を $count()$ としたときに次の式で表される。

$$Jaccard(X,Y) = \frac{count(X \cap Y)}{count(X \cup Y)}$$

本実験においては、集合の一方を各特徴軸に対応する２０単語、他方を各トピックを表す２０単語とした。各トピックを表す単語は次のようにして得られる。各トピック i の全記事を連結した単語列を D_i $(1 \leqq i \leqq k, k$ はトピック数) とする。一方全記事中の語彙 (語数 m) の語を $w_j (1 \leqq j \leqq m)$ とする。各 D_i について、D_i 中に含まれる w_j の頻度を tf_{ij}、w_j を含むトピックの数を df_j とする。このときトピック i について以下の m 次元ベクトルを与える。これはある特定のトピックのみに頻出する単語に対応する次元の値が大きくなるベクトルである。

$$t_i = (tf_{i1} log(\frac{k}{df_1}), \cdots, tf_{im} log(\frac{k}{df_m}))^T$$

このベクトルに対し絶対値が大きな方から p 次元に対応する単語が、そのトピックを表す p 単語となる。

実際の評価値であるが、特徴軸ごとに各トピックとの Jaccard 尺度を計算し、最も高い値をその特徴軸の評価値とする。またこの値を与えるトピックについては検出されたとみなす。各手法の評価値は全特徴軸の評価値の平均値とする。この評価値と検出したトピックの割合を各手法間で比較した。

この評価手法によりユーザへ提示される単語の質を測ることができるが、特徴軸全体の質を表しているわけではない。よって単語の選び方により結果が異なる可能性がある。

コサイン尺度に基づく手法

各手法により得られた k 個の特徴軸 $v_1, ..., v_k$ それぞれについて、上で述べた各トピックを表すベクトル $t_1, ..., t_k$ とのコサインを計算し、そのなかで最大となるコサインの平均を評価値した。式として表すと以下のようになる。

$$\frac{1}{k}\sum_{i=1}^{k} cos(v_i, t_\alpha), \text{ただし} \alpha = \underset{j}{\operatorname{argmax}}\, cos(v_i, t_j)$$

Jaccard 尺度と同様、特徴軸ごとに各トピックとのコサイン尺度を計算し、最も高い値をその特徴軸の評価値とし、それを与えるトピックは検出されたとみなす。また手法の評価値は全特徴軸の評価値の平均値とする。この評価値と検出したトピックの割合を各手法間で比較した。

この評価手法では Jaccard 尺度とは逆で特徴軸全体の質を測ることができるが、ユーザへ提示される単語の質とは必ずしも一致しない。

6.3 実験 1: 全トピックの記事が均一に現われる場合

この実験では最も単純な場合として、一種類のテキストデータについて、各トピックの記事が同数だけある場合を想定した。具体的には各トピックにつきランダムに 30 件の記事を選び、全体で 600 件の記事をトピック検出の対象とした。このとき全記事を連結したテキストデータの語数は 31787 語、語彙数は 4550 語である。実際の 1 記事あたりの平均語数は 53 語である。このデータに対し、ウィンドウ幅を 16、32、64、128、192、256（語）と設定し実験を行なった。

実験結果は図 4 から図 8 である。どの図も横軸はウィンドウ幅である。縦軸は図 4、5 は前項で述べた評価値、図 6、7 は検出されたと考えられるトピックの割合、図 8 は実行時間（秒単位）である。また最もウィンドウ幅が広い場合（256 語）にどのような単語が得られているのかを表 2 から表 5 に示した。各表において v_1 から v_5 の各列はあるひとつの特徴軸から得られた単語を表している。実際には 20 個の特徴軸が存在するが、紙面の都合でそのうち 5 個のみ示している。これらは v_1 から順に TP_5,TP_{11},TP_{15},TP_{16},TP_{20} に関する単語を含んでいる。

評価値を見ると、全体的な傾向としてどの手法も最適なウィンドウ幅を持っており、それよりも大きくても小さくても評価値が下がっていることがわかる。この理由として、大きな場合にはウィンドウ中に複数のトピックが混在し特徴的な単語を見つけることが困難になるため、小さな場合はトピックに対応した単語の共起関係を捉えることが難しくためと考えられる。各手法を比較すると、まず特異値分解は独立成分分析やクラスタリングを用いる手法より悪い結果となった。クラスタリングはウィンドウが小さな場合に他の手法より良い結果となったが、一方でウィンドウが大きくなると急激に値が下がった。一方独立成分分析を用いた手法ではウィンドウが大きな場合でも比較的評価値が下がらないが、逆にウィンドウが小さな場合はあまり良くない結果となった。その理由であるが、独立成分分析はクラスタリングよりも全体の分布を考慮するという性質があり、その結果全体の傾向がわかりづらい小さなウィンドウの場合は評価値が低く、逆に様々な記事が混ざってしまうが全体の傾向が掴みやすい大きなウィンドウの場合には評価値が高くなったと考えられる。また次元削減とクラスタリングを組み合わせた手法は非常に低い値となった。

実際に得られた単語を見ると、次元削減とクラスタリングを組み合わせた手法では、ある特定の語が集中して得られてことがわかる。これは次元削減の際にある特徴的な単語に対応する次元のみが残ってしまったことが原因と考えられる。ただし同じくデータを次元削減した独立成分分析による手法でこのような現象が起こらなかった。これは低次元空間の中でもより独立な軸を発見することで、他の軸と比較してより特徴的な語を見つけることができたと考えられる。またこの結果はウィンドウ幅が 256 語の場合であり、Jaccard 尺度では独立成分分析を用いた手法が他の 3 手法よりもかなり良い結果を与えている。実際に単語を見ても独立成分分析による手法では、v_3 で多少 TP_7 の語が混ざっているものの、他と比較するとテキストデータ中にどのようなトピックが含まれているのかがわかりやすくなっている。

トピックの割合に関して、独立成分分析を用いた手法は他の手法よりも多くのトピックを検出できていることがわかる。特にウィンドウ幅が 32, 64, 128 語のときにはすべてのトピックを検出している。これは独立成分分析による手法が他の手法よりも、独立である特徴軸を発見できていることを示している。逆に次元削減したデータをクラスタリングした場合はごくわずかのトピックしか検出できていない。これは評価値の部分で述べたように次元削減による影響であると考えられる。

実行時間に関して、次元削減をしない場合のクラスタリングのみ極端に時間がかかっていることがわかる。特にウィンドウ幅が小さく全ウィンドウ数が多い場合に顕著な差がある。これは次元数が非常に高い場合 k-means のアルゴリズムにおいてクラスタの重心がなかなか収束しないためと考えられる。つまりある次元の値は収束しているが、他の次元ではまだ収束していないといったことが多々発生するということである。

総合すると極端に狭いウィンドウ幅の場合を除き、独立成分分析が他の手法より有効であることがわかった。独立成分分析の場合、クラスタリングよりもウィンドウ幅に敏感ではないので、適当なサイズを設定することで一定の質を得られると考えられる。

6.4 実験 2: 一部のトピックの記事が頻繁に現われる場合

次に、現実のテキストデータでは一部トピックのみが頻出することがあることを想定した実験を行なった。実験では表 1 の TP_2(Monica Lewinsky) と TP_4(対イラク衝突) の記事が他のトピックよりも多数含まれている状況でのトピック検出を行なった。この 2 つのトピックはそれぞれランダムに 270 件選んだものを用い、他のトピックは 30 件のみ選んだ。従って全体では 1080 件の記事をトピック検出の対象とした。このとき全記事を連結したテキストデータの語数は 62165 語、語彙数は 5717 語である。実際の 1 記事あたりの平均語数は 58 語である。このデータに対しても、ウィンドウ幅を 16、32、64、128、192、256(語) と設定し実験を行なった。

実験結果は図 9 から図 13 である。評価値に関して、全体的に評価値は低下しているが、傾向は実験 1 とそれほど変わっていない。独立成分分析や特異値分解を用いた手法では実験 1 よりウィンドウ幅が広い場合によい評価値となった。これは全体の記事数が増加したことで語彙が増え、ウィンドウ幅が狭い場

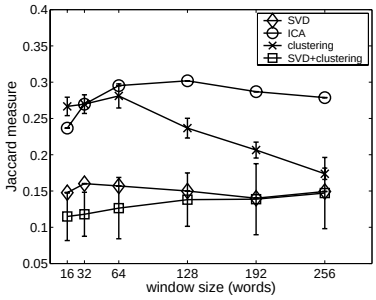

図 4　実験 1 の Jaccard 尺度による評価値

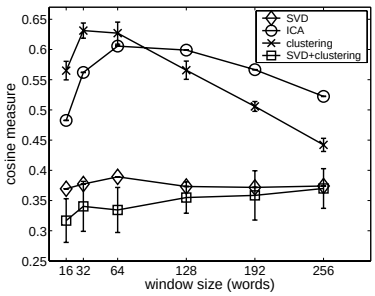

図 5　実験 1 のコサイン尺度による評価値

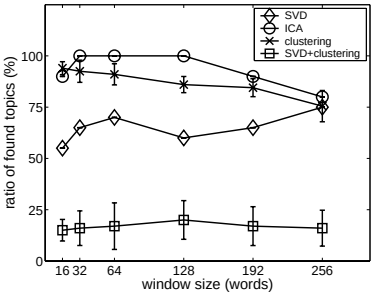

図 6　実験 1 の Jaccard 尺度によるトピック検出率

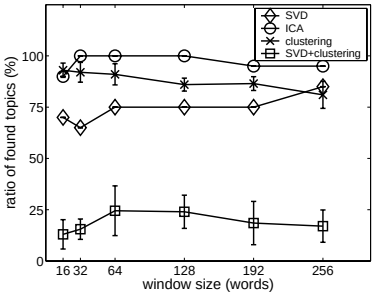

図 7　実験 1 のコサイン尺度によるトピック検出率

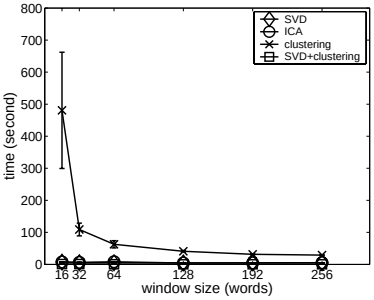

図 8　実験 1 の各手法の実行時間

v_1	super	bowl	bronco	game	denver	play	ve	team	win	plai
v_2	pope	cuba	cuban	castro	visit	paul	john	havana	ii	church
v_3	winfrei	beef	oprah	texa	talk	india	test	nuclear	cattl	cow
v_4	mckinnei	sexual	sergeant	accus	gene	major	armi	court	martial	misconduct
v_5	gm	worker	strike	plant	flint	compani	michigan	motor	car	try

表 2　実験 1 で独立成分分析を用いた手法により求められた単語群

v_1	bowl	super	tobacco	test	viagra	columbia	space	astronaut	asia	shuttl
v_2	cuba	pope	cuban	viagra	john	castro	paul	isra	talk	visit
v_3	bomb	winfrei	clinic	rudolph	isra	oprah	netanyahu	texa	beef	birmingham
v_4	mckinnei	sergeant	major	gene	militari	sexual	winfrei	accus	martial	court
v_5	gm	plant	strike	worker	flint	motor	michigan	viagra	winfrei	team

表 3　実験 1 で特異値分解を用いた手法により求められた単語群

v_1	game	super	play	presid	re	bowl	bronco	ve	don	cnn
v_2	visit	pope	church	peopl	cuba	cuban	dai	bomb	women	clinic
v_3	kaczynski	judg	trial	winfrei	dai	cnn	oprah	peopl	court	beef
v_4	mckinnei	accus	sexual	sergeant	major	gene	former	talk	week	presid
v_5	gm	strike	compani	worker	week	peopl	offici	unit	try	presid

表 4　実験 1 でクラスタリングを用いた手法により求められた単語群

v_1	pope	cuba	visit	super	castro	paul	bowl	havana	bronco	john
v_2	pope	cuba	visit	castro	paul	mckinnei	john	cuban	havana	bomb
v_3	pope	cuba	visit	winfrei	castro	mckinnei	paul	bomb	havana	john
v_4	pope	cuba	visit	paul	castro	mckinnei	john	havana	cuban	bomb
v_5	pope	mckinnei	cuba	bomb	gm	visit	compani	strike	car	super

表 5　実験 1 で次元削減とクラスタリングを組み合わせた手法で求められた単語群

合実験 1 よりも共起関係を捉えることが難しくなったが、ウィンドウ幅が広い場合ウィンドウ中の共起関係はそれほど変化しないため語彙が増加した影響が少なく、結果としてウィンドウ幅が広い場合がよい結果となったと考えられる。またトピックの割合に関しては実験 1 と異なり、広いウィンドウ幅を除き、独立成分分析を用いた手法の方がクラスタリングよりも少ない結果となった。また全体的に検出したトピックの割合が低下している。この原因は、次元削減によって記事が多数含まれるト

ピックの語が多く残ってしまい、その他のトピックの語は特に頻出している語を除いてあまり残らなかったことが考えられる。

そこで現在 20 次元に削減しているものを 40 次元に増やして実験を行なった。この場合 40 個の特徴軸が得られる。これにより次元削減が検出されるトピックの割合に影響を与えているのかがわかる。ここでは Jaccard 尺度による評価値と、検出されたトピックの割合を示す。

その結果は図 14、図 15 である。これを見るとトピックの割合に関しては 20 次元の場合よりもかなり増加しており、次元削減がトピックの割合に影響を与えていることがわかった。評価値は全体的に下がっているが、次元削減とクラスタリングを組み合わせた手法では最も広いウィンドウ幅の場合に極端に良い結果を与えている。ただし検出されたトピックの割合は非常に少ないことから、一部のトピックの語のみ集中的に選ばれていると考えられる。

6.5 実験 3: 記事の語数が大きく異なる 2 種類の配信源を扱う場合

本研究では、対象となるテキストデータは任意であることを想定している。従って同じトピックを含んでいてもテキストデータによって大きく記事の語数が異なることも考えられる。この実験では CNN Headline News と New York Times News Services の両方の記事を混在させたときのトピック検出を行なった。この 2 つで大きく異なるのがその記事の長さであり、CNN は 1 記事あたり 51 語なのに対し New York Times は 1 記事あたり 411 語となる。このどちらについても 1 トピックあたり 20 件、全体では 400 件の記事をトピック検出の対象とした。このとき全記事を連結したテキストデータの語数は 92204 語、語彙数は 8934 語である。実際の 1 記事あたりの平均語数は 231 語である。このデータに対して、ウィンドウ幅を 32、64、128、256、384、512(語) と設定し実験を行なった。

実験結果は図 16 から図 20 である。この結果では、実験 1、実験 2 よりも大きく評価値が下がっていることがわかる。考えられる理由として、同じことを表現する場合でも CNN と New York Times で異なり、共起関係を捉えることが非常に難しくなってしまったことがあげられる。トピックの割合に関しては Jaccard 尺度とコサイン尺度で異なる結果となった。Jaccard 尺度ではクラスタリングが最も多数のトピックを検出しているが、コサイン尺度では一部のウィンドウ幅を除き独立成分分析の方が多くのトピックを検出している。ただし Jaccard 尺度の値から、各トピックから得られた語と特徴軸から得られた語の一致は平均 4 語程度なので、単純にクラスタリングの方が多くのトピックを検出しているとは言えない。

この結果から、記事の語数が大きく異なる、複数の種類のテキストデータからトピックを検出することは比較的困難であり、よりよい手法の検討が望まれる。

7. まとめと今後の課題

本稿では独立成分分析を用いたトピック検出手法を、特異値分解あるいはクラスタリングを用いた手法と比較を行なうことで各手法の性質を明らかにした。独立成分分析を用いた手法は、ウィンドウ幅が小さな場合クラスタリングと比較すると良くないが、ウィンドウ幅が広く様々な記事が含まれている状況において有効であることがわかった。また次元削減を行なったデータに対しより独立な軸を見つけることで、クラスタリングよりも次元削減の影響を軽減していることがわかった。またどの手法も記事の語数が大きく異なる複数の種類のテキストデータからトピックを検出することは難しいことがわかった。

今後の課題としてより詳細な各手法の特徴の分析、上に述べた困難な状況におけるトピック手法の検討、トピック数の推定法や最適なウィンドウ幅の決定手法の検討がある。その他、今回の結果を踏まえ狭いウィンドウ幅ではクラスタリング、広いウィンドウ幅では独立成分分析を用いるといった、両手法の特徴を組み合わせた手法の検討があげられる。

謝辞

本研究の一部は、日本学術振興会日米科学協力事業・共同研究、科学研究費補助金基盤研究 (B)(#15300027)、特定領域研究 (2)(#15017207) による。

文　献

[1] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang. Topic Detection and Tracking Pilot Study Final Report. *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, Virginia, pp.194-218, 1998.

[2] E. Bingham. Topic Identification in Dynamical Text by Extracting Minimum Complexity Time Components. *3rd International Conference on Independent Component Analysys and Blind Signal Separation*, San Diego, California, pp. 546-551, 2001.

[3] J.-F. Cardoso, and A. Souloumiac. Jacobi Angles for Simultaneous Diagonalization. *SIAM Journal on Matrix Analysis and Applications*, vol. 17, no. 1, pp. 161-164, 1996.

[4] S. Deerwester, S.T. Dumais, G.W. Furnas, and T.K. Landauer. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391-407, 1990.

[5] J. Fiscus, G. Doddington, J. Garofolo, and A. Martin. NIST's 1998 Topic Detection and Tracking Evaluation (TDT2). *Proc. of the DARPA Broadcast News Workshop*, Hemdon, Virginia, pp. 19-24, 1999.

[6] A. Hyvärinen. Fast and Robust Fixed-Point Algorithms for Independent Component Analysis. *IEEE Transactions on Neural Networks*, vol. 10, no. 3, pp.626-634, 1999.

[7] A. Hyvärinen, J. Karhunen, and E. Oja. Independent Component Analysis. John Wiley, New York, 2001.

[8] T. Kolenda, L.K. Hansen, and J. Larsen. Signal Detection Using ICA: Application to Chat Room Topic Spotting. *3rd International Conference on Independent Component Analysys and Blind Signal Separation*, San Diego, California, pp. 540-545, 2001

[9] J. MacQueen. Some Methods for Classification and Analysis of Multivariate Observations. *Proc. 5th Berkeley Symposium in Mathematics, Statistics and Probability*, 1:281-297, 1967.

[10] G. Salton, A. Wong, and C.S. Yang. A Vector Space Model for Automatic Indexing. *Communications of the ACM*, vol. 18, no. 11, pp.613-620, 1975.

[11] J. M. Schultz, and M. Liberman. Topic Detection and Tracking using idf-Weighted Cosine Coefficient. *Proc. DARPA Broadcast News Workshop*, Hemdon, Virginia, pp. 189-192, 1999.

[12] F. Walls, H. Jin, S. Sista, and R. Schwartz. Topic Detection in Broadcast News. *Proc. DARPA Broadcast News Workshop*, Hemdon, Virginia, pp. 193-198, 1999.

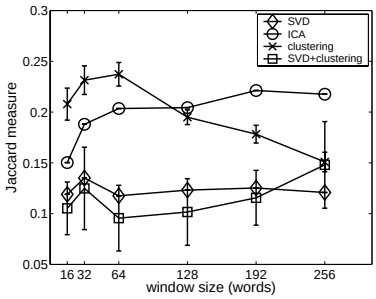

図 9　実験 2 の Jaccard 尺度による評価値

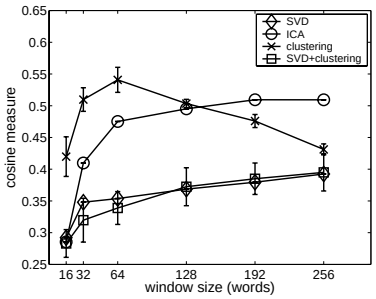

図 10　実験 2 のコサイン尺度による評価値

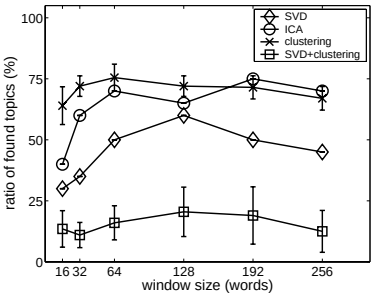

図 11　実験 2 の Jaccard 尺度によるトピック検出率

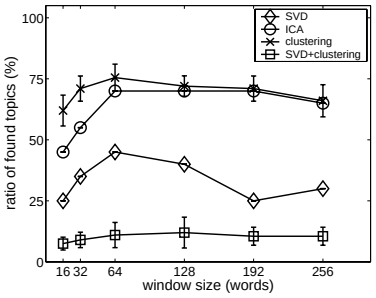

図 12　実験 2 のコサイン尺度によるトピック検出率

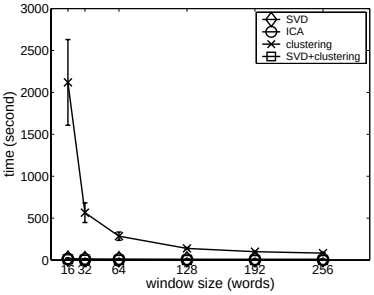

図 13　実験 2 の各手法の実行時間

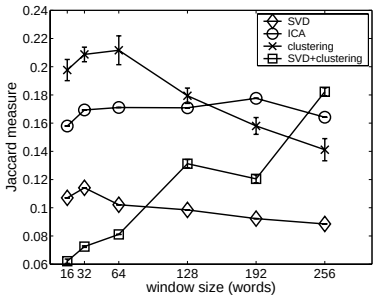

図 14　実験 2 の Jaccard 尺度による評価値 (40 特徴軸)

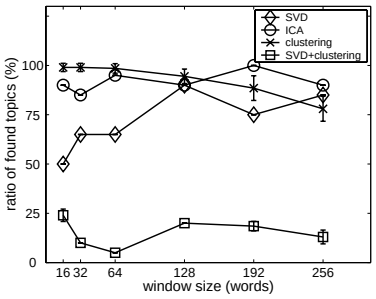

図 15　実験 2 の Jaccard 尺度によるトピック検出率 (40 特徴軸)

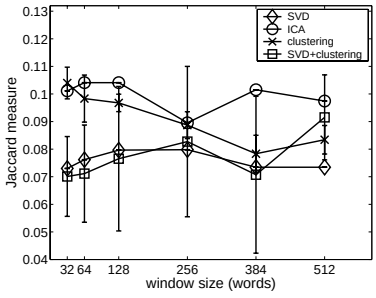

図 16　実験 3 の Jaccard 尺度による評価値

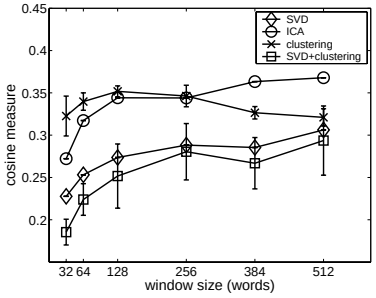

図 17　実験 3 のコサイン尺度による評価値

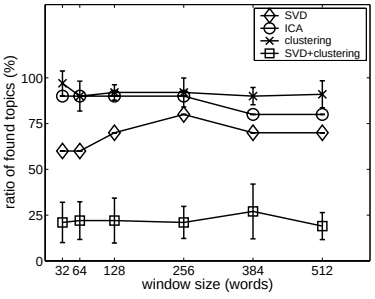

図 18　実験 3 の Jaccard 尺度によるトピック検出率

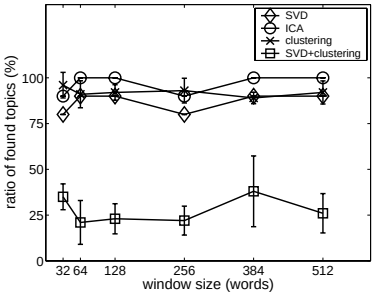

図 19　実験 3 のコサイン尺度によるトピック検出率

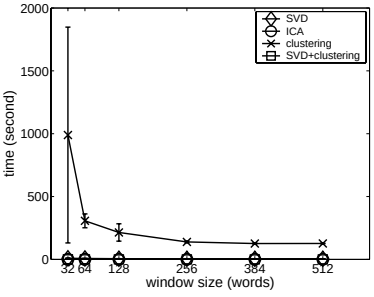

図 20　実験 3 の各手法の実行時間