ランダム・プロジェクションによる ベクトル空間情報検索モデルの次元削減

佐々木 稔[†] 北 研 二^{††}

ベクトル空間モデルは情報検索における代表的な検索モデルである。ベクトル空間モデルでは文書を索引語の重みベクトルで表現するが、文書ベクトルは一般に要素数が非常に多く、スパースなベクトルになるため、検索時間の長さや必要なメモリの量が大きな問題となる。本論文では、この問題を解決するため、ベクトル空間モデルにおけるベクトルの次元圧縮を行う手法としてランダム・プロジェクションを用いた検索モデルを提案する。その有効性を評価するために、評価用テストコレクションであるMEDLINEを利用して、検索実験を行った。その結果、ランダム・プロジェクションはLSI(Latent Semantic Indexing)に比べ高速で、かつ同等な検索性能を持つ次元圧縮手法であることが確認された。また、ランダム・プロジェクションで次元圧縮に必要な行列を得るために、球面 k 平均アルゴリズムで得られる概念ベクトルの利用を提案する。同様に検索実験を行った結果、任意のベクトルを用いた検索性能に比べ改善され、概念ベクトルが検索性能の向上に有効であることが確認された。

キーワード: 情報検索, ベクトル空間モデル, ランダム・プロジェクション, LSI, 概念ベクトル

Dimensionality Reduction of Vector Space Information Retrieval Model Based on Random Projection

MINORU SASAKI[†] and KENJI KITA^{††}

Vector space model is a conventional information retrieval model, in which text documents are represented as high-dimensional and sparse vectors using words as features in a multidimensional space. These vectors require a large number of computer resources and it is difficult to capture underlying concepts referred to by the terms. In this paper, we present a technique of an information retrieval model using a random projection to project document vectors to a low-dimensional space as a way of solving these problems. To evaluate its efficiency, we show results of retrieval experiments on the MEDLINE test collection. Experiments show that the proposed method is faster than LSI(Latent Semantic Indexing) and efficient close to the LSI. In addition, we propose to apply a concept vector, which random projection needs for dimensionality reduction, produced by a spherical k -means algorithm. A result of this evaluation shows that the concept vector captures the underlying concepts of the corpus effectively.

KeyWords: *Information retrieval, Vector space model, Random projection, Latent semantic indexing, Concept vector*

[†] 徳島大学大学院工学研究科, Graduate School of Engineering, University of Tokushima

^{††} 徳島大学工学部, Faculty of Engineering, University of Tokushima

1 はじめに

近年、インターネットの普及とともに、個人で WWW (World Wide Web) を代表とするネットワーク上の大量の電子データやデータベースが取り扱えるようになり、膨大なテキストデータの中から必要な情報を取り出す機会が増加している。しかし、このようなデータの増加は必要な情報の抽出を困難とする原因となる。この状況を反映し、情報検索、情報フィルタリングや文書クラスタリング等の技術に関する研究開発が盛んに進められている。

情報検索システムの中でよく使われている検索モデルに、ベクトル空間モデル (Salton and McGill 1983) がある。ベクトル空間モデルは、文書と検索要求を多次元空間ベクトルとして表現する方法である。基本的には、文書集合から索引語とするタームを取り出し、タームの頻度などの統計的な情報により、文書ベクトルを表現する。この際、タームに重みを加えることにより、文書全体に対するタームの特徴を目立たせることが可能である。この重みを計算するためには、IDF (Inverse Document Frequency) (Chicholm and Kolda 1998) などの重みづけ方法が数多く提案されている。また、文書と検索要求を比較する類似度の尺度として、内積や余弦 (cosine) がよく用いられている。この類似度計算により、類似度の高いものからランクづけを行い、ユーザに表示することができることもベクトル空間モデルの特徴のひとつである。

ベクトル空間モデルを用いた検索システムを新聞記事などの大量の文書データに対して適用した場合、文書データ全体に存在するタームの数が非常に多くなるため、文書ベクトルは高い次元を持つようになる。しかし、ひとつの文書データに存在するタームの数は文書データ全体のターム数に比べると非常に少なく、文書ベクトルは要素に 0 の多い、スパースなベクトルになる。このような文書ベクトルを用いて類似度を計算する際には、検索時間の増加や文書ベクトルを保存するために必要なメモリの量が大きな問題となる。このため、単語の意味や共起関係などの情報を用いたり、ベクトル空間の構造を利用してベクトルの次元を圧縮する研究が盛んに行われている。このようなベクトルの次元圧縮技術には、統計的なパターン認識技術や線形代数を用いた手法などが用いられている (Kolda and O'Leary 1998) (Faloutsos and Lin 1995)。この中で、最も代表的な手法として、LSI (Latent Semantic Indexing) がある (Deerwester, Dumais, Landauer, Furnas, and Harshman 1990) (Dumais 1995)。この手法は、文書・単語行列を特異値分解を用いて、低いランクの近似的な行列を求めるものであり、これを用いた検索システムは、次元圧縮を行わない検索モデルと比較して一般的に良い性能を示す。しかし、特異値分解に必要な計算量が大いいために、検索モデルを構築する時間が非常に長いことが問題となっている。

上記の問題を解決するベクトル空間モデルの次元圧縮手法に、ランダム・プロジェクション (Arriaga and Vempala 1999) が存在する。ランダム・プロジェクションは、あらかじめ指定した数のベクトルとの内積を計算することで次元圧縮を行う手法である。これまでに報告されているランダム・プロジェクションを用いた研究には、VLSI (Very Large-Scale Integrated circuit) の設計問題への利用 (Vempala 1998) や次元圧縮後の行列の特性を理論的に述べたもの

がある (Papadimitriou, Raghavan, Tamaki, and Vempala 1998)(Arriaga and Vempala 1999). しかし、これらの文献では、ランダム・プロジェクションの理論的な特性は示されているものの、情報検索における具体的な実験結果は報告されていない。そのため、情報検索に対するランダム・プロジェクションの有効性に疑問が残る。

我々は、ランダム・プロジェクションを用いた情報検索モデルを構築し、評価用テストコレクションである MEDLINE を利用した検索実験を行った。この検索実験より、情報検索における次元圧縮手法として、ランダム・プロジェクションが有効であることを示す。また、ランダム・プロジェクションを行う際にあらかじめ指定するベクトルとして、文書の内容を表す概念ベクトル (Dhillon and Modha 1999) の利用を提案する。概念ベクトルは文書の内容が似ているベクトル集合の重心で、この概念ベクトルを得る際、高次元でスパースな文書データ集合を高速にクラスタリングすることができる球面 k 平均アルゴリズム (Dhillon and Modha 1999) を用いる。これにより、文書集合を自動的にクラスタリングできるだけでなく、ランダム・プロジェクションに必要な概念ベクトルも同時に得ることができる。この概念ベクトルをランダム・プロジェクションで用いることにより、任意のベクトルを用いた検索性能と比較して、検索性能が改善されていることを示し、概念ベクトルを利用した次元圧縮の有効性を示す。

2 ランダム・プロジェクションによるベクトルの次元圧縮

本節では、ランダム・プロジェクションを用いたベクトル空間モデル (Papadimitriou et al. 1998)(Arriaga and Vempala 1999) についての概観を述べる。ランダム・プロジェクションは、ひとつの文書データを n 次元空間上のベクトル $\mathbf{u}$ として表現するとき、このベクトルを k ($k < n$) 次元空間に射影する手法である。その際、 k 個の任意の n 次元ベクトル $\mathbf{r}_1, \dots, \mathbf{r}_k$ を用意する。用意したこれらのベクトルと n 次元ベクトル $\mathbf{u}$ の内積、

$$\mathbf{u}'_1 = \mathbf{r}_1 \cdot \mathbf{u}, \dots, \mathbf{u}'_k = \mathbf{r}_k \cdot \mathbf{u} \quad (1)$$

をそれぞれ計算する。その結果、 k 次元に圧縮した $\mathbf{u}'_1, \dots, \mathbf{u}'_k$ を要素とするベクトルが得られる。

次元圧縮に必要なベクトル $\mathbf{r}_1, \dots, \mathbf{r}_k$ を列ベクトルとする $n \times k$ の行列 $\mathbf{R}$ を用いると、求める k 次元ベクトルは

$$\mathbf{u}' = \mathbf{R}^T \mathbf{u} \quad (2)$$

となり、ランダム・プロジェクションは行列計算のみの簡単な形で表現することができる。この行列 $\mathbf{R}$ が任意の正規直交行列のとき、すなわち、行列 $\mathbf{R}$ の列ベクトルがすべて単位ベクトルで、かつ、相異なる列ベクトルが互いに直交していれば、ランダム・プロジェクションは射影前後におけるベクトル間距離を近似的に保存する特性を持っている。

3 概念ベクトルを用いたランダム・プロジェクション

ランダム・プロジェクションに必要な行列 $\mathbf{R}$ は、これまでの研究では正規分布などの確率分布をなす任意の行列が用いられている (Papadimitriou et al. 1998)(Vempala 1998)(Arriaga and Vempala 1999) (Kleinberg, Papadimitriou, and Raghavan 1998)(Blum, Konjevod, Ravi, and Vempala 1998)(Feige 1998). このような行列を用いて任意の部分空間に射影する場合、次元圧縮を行う前後の任意のベクトル間距離は近似的に保存されることが示されている (Frankl and Maehara 1988)(Johnson and Lindenstrauss 1984). しかし、任意の正規直交行列を用いる場合、次元圧縮を行う前後のベクトル間距離を保存する効果は得られたとしても、LSI のように、ベクトルの要素が抽象的な意味を持つ索引語の生成や内容的に関連のある文書をまとめる効果があるとは考えられない. このことから、LSI のような、情報検索に有効な索引語を生成するために、ランダム・プロジェクションの改良が課題となる.

このような課題を解決するものとして、ランダム・プロジェクションでベクトルを次元圧縮をした後、さらに特異値分解を行うことにより、LSI の効果を得る手法が提案されている (Papadimitriou et al. 1998). この手法は、関連文書をまとめる効果を得ると同時に、特異値分解のみを用いた場合に比べ、モデル作成に必要な時間を短縮したものである. しかし、ランダム・プロジェクションと特異値分解は、共にベクトル間距離を保存する効果を持つ手法であるため、特異値分解が内容的に関連のある文書、あるいはタームをまとめるために適用されているとしても、これらの手法を同時に利用することは、検索モデルを構築する時間に関して、効率の良い手法であるとはいえない. さらに、非常に大きい次元数をもつ行列について考えた場合、特異値分解に多くの計算量が必要であることも問題となる. したがって、特異値分解により誤差を最小とする近似行列を得る代わりに、誤差は最小ではないものの、ランダム・プロジェクションのみを用いて LSI の効果を得ることで、より高速に検索モデルが構築できるのではないかと考えられる.

これを実現するために、我々は、ランダム・プロジェクションにおける行列 $\mathbf{R}$ に、文書の内容を表現した概念ベクトルを利用することを提案する. 概念ベクトルは、文書ベクトル集合をクラスタリングしてできたクラスタの、各クラスタに属する文書ベクトルの重心を正規化したベクトルとして表される. この概念ベクトルによる次元圧縮は、単にベクトルを近似するだけではなく、クラスタに属するベクトル集合の重心を求めることにより、ターム間で特徴づけられる隠れた関連性やタームの同義性と多義性を捉えることができる. クラスタリングにより得られた各クラスタは互いに異なる概念を持ち、これより得られる概念ベクトルが圧縮した空間の軸となるように用いられる. これにより、次元圧縮された行列は文書と概念ベクトルの類似度を表し、元の空間において内容の近い文書は、圧縮した空間においても近くなる可能性がある. また、類似しているが、異なるタームを使った文書の場合、元の空間では近くないが、圧縮した空間では近くなる可能性があり、検索性能が改善されると考えられる. さらに、多義語

により元の空間において近いとされる文書どうしが圧縮した空間では遠くに離れ、誤った検索が取り除かれる可能性も期待できる。このように、これまで単語などが要素であったベクトルが、文書の内容を要素とするようなベクトルに変換され、文書を低い次元で、より検索性能が向上するベクトル表現ができると考えられる。

概念ベクトルからなる行列 $\mathbf{R}$ を求めるために、球面 k 平均アルゴリズム (Dhillon and Modha 1999) と呼ばれるクラスタリング手法を用いる。球面 k 平均アルゴリズムは、目的関数が局所的に最大となるまで、高い次元でスパースな文書データ集合をクラスタリングする手法である。球面 k 平均アルゴリズムでは、ユークリッド空間内でベクトル間のなす角の余弦を類似度とし、多次元空間の単位円を分割することによりクラスタリングを行う。これにより、文書ベクトルの集合は指定した数の部分集合に分割され、各クラスタの中心を計算することで、容易に概念ベクトルを作ることができる。さらに、このアルゴリズムは文書ベクトルのスパースさを逆に利用して高速に収束する利点を持ち、得られる概念ベクトルは特異値分解を用いたものに非常に近いことが示されている (Dhillon and Modha 1999)。

しかし、球面 k 平均アルゴリズムにより得られる概念ベクトルは一般的に直交性を満たしているとは限らないため、概念ベクトルをランダム・プロジェクションに適用するには疑問が生じる。先に述べたように、距離を保存するには正規直交性を満たすベクトルを利用する必要があるが、この概念ベクトルをランダム・プロジェクションに適用する場合、直交性を満たしていないとしても独立であれば、任意の行列においても十分に距離を保存する可能性のあることが示されている (Arriaga and Vempala 1999)。球面 k 平均アルゴリズムでは、内容的に似通ったベクトルをクラスタとしてまとめるため、原理的には独立した概念ベクトルを生成すると考えられる。このため、直交性に関して、概念ベクトルをランダム・プロジェクションに適用するのは問題ないと考えられる。

本節では、まず、球面 k 平均アルゴリズムの概要を述べる前に、クラスタリングにより得られる概念ベクトルについて述べる。

3.1 概念ベクトル

ベクトルの集合をベクトル空間にプロットしたとき、同質のベクトルが多く存在する場合を除いて、いくつかのグループに分かれる。このようなグループはクラスタと呼ばれ、類似した内容をもつベクトルの集合が形成される。概念ベクトルはクラスタに属するベクトルの重心を求めることにより得られ、そのクラスタの内容を表す代表ベクトルである。

概念ベクトルを求める例として、正規化された N 個のベクトル $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ を、異なる s ($s < N$) 個のクラスタ $\pi_1, \pi_2, \dots, \pi_s$ にクラスタリングすることを考える。このとき、ひとつ

のクラスタ π_j に含まれるベクトル x_i の平均である重心 $\mathbf{m}_j$ は以下のように表される.

$$\mathbf{m}_j = \frac{1}{n_j} \sum_{\mathbf{x}_i \in \pi_j} \mathbf{x}_i \quad (3)$$

ここで n_j はクラスタ π_j に含まれるベクトルの数を表す. ベクトルの重心は単位長にはなっていないので, そのベクトルの長さで割ることにより概念ベクトル $\mathbf{c}_j$ を得る.

$$\mathbf{c}_j = \frac{\mathbf{m}_j}{\|\mathbf{m}_j\|} \quad (4)$$

3.2 目的関数

k 平均アルゴリズムでは, 目的関数は一般的に概念ベクトルとクラスタに属するベクトルとの距離の和

$$\sum_{\mathbf{x}_i \in \pi_j} \|\mathbf{m}_j - \mathbf{x}_i\| \quad (5)$$

を最小にするような概念ベクトルを求める, 最小二乗法が用いられる. 球面 k 平均アルゴリズムでは, このような最小化問題ではなく, ミクロ経済学の分野における, 生産計画の最適化問題で扱われている目的関数を用いている (Kleinberg et al. 1998). これは, 各クラスタ $\pi_j (1 \leq j \leq s)$ の密度を

$$\sum_{\mathbf{x}_i \in \pi_j} \mathbf{x}_i^T \mathbf{c}_j \quad (6)$$

とし, クラスタの結合密度の和を目的関数としている.

$$D = \sum_{j=1}^s \sum_{\mathbf{x}_i \in \pi_j} \mathbf{x}_i^T \mathbf{c}_j \quad (7)$$

クラスタの密度は, 以下のコーシー・シュワルツの不等式より, 任意の単位ベクトル $\mathbf{z}$ に対して, クラスタ π_j に含まれるベクトル x_i と概念ベクトルとの内積の総和が最大となる.

$$\sum_{\mathbf{x}_i \in \pi_j} \mathbf{x}_i^T \mathbf{z} \leq \sum_{\mathbf{x}_i \in \pi_j} \mathbf{x}_i^T \mathbf{c}_j \quad (8)$$

また, クラスタの密度は, それに属するベクトル和の距離に等しくなるという特徴を持っている.

$$\sum_{\mathbf{x}_i \in \pi_j} \mathbf{x}_i^T \mathbf{c}_j = \left\| \sum_{\mathbf{x}_i \in \pi_j} \mathbf{x}_i \right\| \quad (9)$$

3.3 球面 k 平均アルゴリズム

3.2節で示した目的関数 D を最大にするように, ベクトルの集合を反復法によりクラスタリングする. 文書ベクトル $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ を s 個のクラスタ $\pi_1^*, \pi_2^*, \dots, \pi_s^*$ に分割するためのアルゴリズムを以下に示す.

- (1) すべての文書ベクトルを s 個のクラスタに任意に分割する．これらの部分集合を $\{\pi_j^{(0)}\}_{j=1}^s$ とし、これより求められた概念ベクトルの初期集合を $\{\mathbf{c}_j^{(0)}\}_{j=1}^s$ とする．また、 t を繰り返しの回数とし、初期値は $t=0$ である．
- (2) 各文書ベクトル $\mathbf{x}_i (1 \leq i \leq N)$ に対し、余弦が最も大きい、最も文書ベクトルに近い概念ベクトルを見つける．このとき、すべての概念ベクトルは正規化されているので、余弦は文書ベクトル $\mathbf{x}_i$ と概念ベクトル $\mathbf{c}_j^{(t)}$ の内積を求めることと同値である．これにより、前回の繰り返しの求めた概念ベクトル $\{\mathbf{c}_j^{(t)}\}_{j=1}^s$ から、文書ベクトルが新たな部分集合 $\{\pi_j^{(t+1)}\}_{j=1}^s$ に分割される．

$$\pi_j^{(t+1)} = \{\mathbf{x}_i : \mathbf{x}_i^T \mathbf{c}_j^{(t)} \geq \mathbf{x}_i^T \mathbf{c}_l^{(t)} \mid (1 \leq l \leq N, 1 \leq j \leq s)\} \quad (10)$$

ここで、 $\pi_j^{(t+1)}$ は概念ベクトル $\mathbf{c}_j^{(t)}$ に近いすべての文書ベクトルの集合とする．

- (3) 新たに導かれた概念ベクトルの長さを正規化する．

$$\mathbf{c}_j^{(t+1)} = \frac{\mathbf{m}_j^{(t+1)}}{\|\mathbf{m}_j^{(t+1)}\|}, \quad (1 \leq j \leq s) \quad (11)$$

ここで、 $\mathbf{m}_j^{(t+1)}$ はクラスタ $\pi_j^{(t+1)}$ の文書ベクトルの重心を表す．

- (4) 目的関数 $D^{(t+1)}$ の値を求め．前回の繰り返しの目的関数の値 $D^{(t)}$ との差を計算する．このとき、

$$\|D^{(t)} - D^{(t+1)}\| \leq 1 \quad (12)$$

を満たす場合、 $\pi_j^* = \pi_j^{(t+1)}$ 、 $\mathbf{c}_j^* = \mathbf{c}_j^{(t+1)}$ ($1 \leq j \leq s$) とし、アルゴリズムを終了する．停止基準を超えていない場合は、 t に 1 を加え、ステップ 2 に戻る．ここで、停止基準における目的関数の差は、文書数が約 4000 で、クラスタの数が 8 よりも大きい場合、収束した時の目的関数は 1000 を超えることがこれまでの研究で報告されている (Dhillon and Modha 1999)．このため、繰り返しの 1 以下の差は無視できるとし、便宜的に 1 という値を設定した．

4 実験

本節では、ランダム・プロジェクションを用いた検索モデルを構築し、その評価として、MEDLINE を用いた検索実験について述べる．

4.1 データ

実験で用いたデータは、情報検索システムの評価用テストコレクションである MEDLINE を利用した．MEDLINE は医学・生物学分野における英文の文献情報データベースで、検索

の対象となる文書の件数は 1033 件で、約 1Mbyte の容量を持つテキストデータである。また、MEDLINE には 30 個の評価用検索要求文と各要求文に対する正解文書が用意されている。

MEDLINE に含まれている 1033 件の文書全体から、前処理として、“a” や “about” などの一般的な 439 個の英単語を不要語リストに指定して、文書の内容と関係のほとんどない単語は削除した。この後、接辞処理を行い、残った英単語を語幹に変換する処理を行った。この前処理の結果、文書全体に 5526 個あった単語から、4329 個の単語が索引語として抽出され、実験データとして用いた。

4.2 検索実験方法

実験では、MEDLINE から前処理により得られた索引語を要素とする文書ベクトルと検索要求ベクトルを作成し、比較することで検索スコアを計算する。文書ベクトルを作成するとき、ベクトルの要素には局所的、大域的な索引語の分布を考慮するために、索引語の頻度に重み付けした数値が用いられる。数多く提案されている重みづけ手法で、今回の実験では以下の式で定義された対数エントロピー重み (Chicholm and Kolda 1998) を用いた。 L_{ij} は j 番目の文書に対する i 番目の索引語への重み、 G_i は文書全体に対する i 番目の索引語への重みを表す。

$$L_{ij} = \begin{cases} 1 + \log f_{ij} & (f_{ij} > 0) \\ 0 & (f_{ij} = 0) \end{cases} \quad (13)$$

$$G_i = 1 + \sum_{j=1}^n \frac{\frac{f_{ij}}{F_i} \log \frac{f_{ij}}{F_i}}{\log n} \quad (14)$$

ここで、 n は全文書数、 f_{ij} は j 番目の文書に出現する i 番目の索引語の頻度、 F_i は文書集合全体における i 番目の索引語の頻度を表す。これより、 j 番目の文書から得られる文書ベクトルの i 番目の要素 d_{ij} は、

$$d_{ij} = L_{ij} \times G_i \quad (15)$$

となる。

得られた文書ベクトルから、球面 k 平均アルゴリズムを用い、これらの文書ベクトルより指定された数の概念ベクトルを作成する。作成した概念ベクトルを結合した行列に対し、ランダム・プロジェクションを行い、文書ベクトル、検索要求ベクトルの次元を削減する。次元の削減されたベクトルを用いて、内積の計算を行い、その値を各文書に対する検索スコアとする。これらのスコアのうち、上位 50 文書を検索結果として出力する。

検索システムの評価には、一般的に用いられている正解率 (Precision) と再現率 (Recall) を用いた (Lewis 1991)(Witten, Moffat, and Bell 1994)。

$$\text{Recall} = \frac{\text{システムが出力した正解文書数}}{\text{全正解文書数}} \quad (16)$$

表 1 各次元数における平均正解率
表 1 Average precision at each number of dimensions

次元数	ランダム・プロジェクション	平均正解率
100	あり	0.3982
200	あり	0.4711
300	あり	0.5154
400	あり	0.5231
500	あり	0.5673
600	あり	0.5748
700	あり	0.5822
800	あり	0.5979
900	あり	0.6037
1033	なし	0.4936

$$\text{Precision} = \frac{\text{システムが出力した正解文書数}}{\text{システムが出力した文書数}} \quad (17)$$

再現率と正解率は、それぞれ個別に用いて、システム評価を行うことができるが、本実験では、一般にランクづけ検索システムの評価に用いられる再現率・正解率曲線を用い、システムの評価を行った。この曲線は、各質問に対しひとつの曲線が作成されるが、本稿の検索システム評価には、全 30 個の質問に対する各再現率での平均を計算した再現率・正解率曲線を用いた。

5 実験結果および考察

5.1 次元数による比較

本実験では、ランダム・プロジェクションにより、ベクトルの次元を 100 から 900 まで圧縮した検索モデルについて、検索実験を行った。その結果、各次元における平均正解率は表 1 のようになった。平均正解率は、ベクトルの次元が大きくなるにつれて増加し、次元数 300 において、次元圧縮を行わないベクトル空間モデルよりも良い結果となった。また、次元数が 400 から 500 に変化させたときの平均正解率の増加が最も大きく、それ以降は変化の割合が少なくなっている。次元数を大きくすれば、検索に必要な計算量が増加する。このことから、効果的な検索を行うためには、全文書数の約半分に次元圧縮を行う必要があることが分かった。

5.2 検索モデル作成時間

検索モデルを作成する時間、および、一つの検索要求に対し、検索を行うために必要な時間を測定した結果を述べる。検索実験には、Ultra Sparc(330MHz) のマシンを使用し、ベクトル

表 2 モデル作成時間とひとつの検索要求に対する検索時間

表 2 Processing time for making a retrieval model and retrieving one query

手法	モデル作成時間	検索時間
ランダム・プロジェクション	約 2 分	4 秒
LSI	約 24 分	4 秒

表 3 文書数の変化によるモデル作成時間

表 3 Processing time for making a retrieval model and retrieving one query

データ	ランダム・プロジェクション	LSI
MEDLINE	約 2 分	約 24 分
MEDLINE+CISI	約 14 分	約 26 分
MEDLINE+CISI+CRANFIELD	約 34 分	約 43 分

の次元を 500 とした結果、表 2 に示すように、ランダム・プロジェクションを用いた場合、モデルを作成する時間は約 11 分必要であった。LSI の場合、SVD の計算については SVDPACK の中で最も高速な Lanczos 法を利用し、同様にベクトルの次元を 500 とした結果、モデルを作成する時間は約 24 分で必要であった。この結果、ランダム・プロジェクションは LSI に比べ、高速に検索モデルを構築することができた。

このモデル作成時間においては、メモリサイズの大きさによる、SVD の計算時間に与える影響が考えられる。スワップ領域を用いるほどの大規模なデータについては大きな影響を及ぼし、モデル作成の時間を多く必要とするが、本実験において用いたマシンには 640M バイトのメモリを搭載しているため、MEDLINE コレクションのような規模のデータに対しては、メモリサイズの影響はほとんどないと考えられる。

本実験で用いた MEDLINE には収録されているデータは 1033 件と比較的少ない。このため、文書数を変化させたときの検索モデル構築時間の変化について比較を行った。文書数を増加させるために、MEDLINE と同様なテストコレクションである CISI を併せた 2493 記事、さらに CRANFIELD を併せた 3893 記事について、それぞれの検索モデル作成時間を測定した。その結果、ランダム・プロジェクションと LSI のモデル作成時間は表 3 のようになった。これより、文書数が増加に対して球面 k 平均アルゴリズムの 1 回の反復による計算量が大きくなるのであるが、ランダム・プロジェクションが検索時間に関しても有効であることが分かる。しかし、非常に大規模な文書数に対しては、より 1 回の反復による計算量が増加するため、反復計算を必要とせずに、球面 k 平均アルゴリズム並の概念ベクトルを得ることが課題となった。

図 1 モデルに対する再現率・正解率曲線

図 1 Recall-Precision curve for comparison between models

5.3 他の検索モデルとの比較

ランダム・プロジェクションを用いた検索モデルに対して、モデルとしての有効性について評価をする。この評価をするために、次元圧縮をしていない元のベクトル空間モデルと特異値分解を用いた LSI による検索モデルについての検索実験も同時に行い、性能を比較した。このとき、比較として用いた LSI は、次元数 100 として次元圧縮した検索モデルを用いている。これらの検索モデルについて、同様に検索実験を行い、すべての検索質問の平均を求めた再現率・正解率曲線を図 1 に示す。図 1 において、横軸は再現率を表し、縦軸は正解率を表す。またグラフの 'LSI100' は次元数 100 の LSI, 'VSM' は次元圧縮なしのベクトル空間モデル, 'RP500', 'RP700', 'RP900' はランダム・プロジェクションによるそれぞれに示された次元数に圧縮したモデルの実験結果である。

その結果、ベクトル空間モデルと比較して、ランダム・プロジェクションを用いた検索モデルは、大幅に性能が改善されていることが分かった。また、次元数 100 の LSI と比較すると、ランダム・プロジェクションは LSI に比べ少し下がりつつはいるものの、ほぼ同じ程度に検索精度が改善されていることを示している。このことから、ランダム・プロジェクションが検索モデルとして、LSI と同等の性能を持っていることが分かる。

5.4 概念ベクトルの有効性

ランダム・プロジェクションで次元圧縮に用いられる概念ベクトルが有効であるかを評価するために、他のベクトルを用いて次元圧縮が行われた場合との検索結果の比較を行った。ベクトルには、乱数を用いて、全要素の平均が0、分散が1の正規分布 $N(0, 1)$ となるベクトルと、指定された数の文書ベクトルを任意に抽出して得られた部分集合からなるベクトルを、それぞれ次元圧縮に用いた。この結果、再現率・正解率曲線は図2となった。ここで、‘Random’は正規分布となるベクトル、‘Subset’は文書ベクトルの部分集合を表し、共にベクトルの次元数は500として、次元圧縮を行ったモデルの実験結果である。また、サンプルに使った文書集合の偏りを考慮するため、グラフに示した実験でのベクトルの他にいくつかのサンプルを用意し、同様の実験を行い、平均的な検索精度を求めた。その結果、正規分布による任意のベクトルにおける平均正解率の平均値は0.38、文書ベクトルの部分集合における平均値は0.47となった。

このグラフと平均値から、正規分布の性質を持つ任意のベクトルや文書ベクトルの部分集合を用いて次元圧縮を行った結果とそれぞれ比較すると、概念ベクトルを用いて次元圧縮を行った結果が、明らかに優れていることが分かる。乱数により生成したベクトルを用いた場合、これらのベクトルの各要素には、索引の重要度や索引語間の関連性はほとんど存在しない。このようなベクトルにより次元圧縮を行う場合、ベクトルの要素には文書の内容を表すような潜在的な意味がほとんど含まれていないために、検索性能が下がってしまったと考えられる。

文書ベクトルの部分集合を用いた場合は、次元圧縮後、ベクトル中のいくつかの要素が似通った意味を持っているために、検索性能が下がったと考えられる。概念ベクトルは、内容の似通った文書がクラスタリングによりひとつにまとめられ、それらの重心を求めることで、文書の内容を端的に表すことができる。また、クラスタリングを行うことで似通った内容を持つ概念ベクトルが少なくなるため、内容がほとんど変わらない概念ベクトルを重複して生成する可能性が少ない。しかし、文書の部分集合では、内容の重複した文書が複数存在する可能性がある。このため、次元圧縮後のベクトル空間モデルに意味の重なった要素が存在し、検索性能が下がってしまう可能性が大きくなってしまふと考えられる。これらのことにより、情報検索に対してランダム・プロジェクションを用いて次元圧縮を行う場合、内容の近い文書や同義語などのような索引語の特徴を表した概念ベクトルを用いることにより、優れた検索性能が得られることが示された。

6 おわりに

本論文では、ベクトル空間モデルの次元圧縮手法として、ランダム・プロジェクションを用いた検索モデルを提案した。このモデルの有効性を評価するために、MEDLINEを利用した検索実験を行った。その結果、次元圧縮していない元のベクトル空間モデルと比べ検索精度が改

図 2 概念ベクトルに対する再現率・正解率曲線
 図 2 Recall-Precision curve for comparison between vectors

善されていることが分かった。また、LSI と比較しても、検索精度の差は少なく、ランダム・プロジェクションが LSI と同程度の次元圧縮性能を持っていることが分かった。LSI とランダム・プロジェクションのモデル作成、検索に必要な時間を比較すると、LSI は特異値分解を行うこともあり、ランダム・プロジェクションは LSI に比べ約半分の時間で検索を行うことができた。また、MEDLINE よりも大規模な文書集合に対しても、ランダム・プロジェクションが高速に検索モデルが構築することができる。これらのことから、ランダム・プロジェクションは LSI に比べ、高速、かつ有効な次元圧縮手法であることが分かった。

また、ランダム・プロジェクションで次元圧縮に必要な行列を得るために、球面 k 平均アルゴリズムで得られる概念ベクトルの利用を提案し、その有効性を検索実験にて評価した。その結果、乱数により生成したベクトルや文書ベクトルの部分集合を用いた場合に比べ、検索精度が優れていた。文書間の内容などの特徴を表した概念ベクトルを用いることで、その概念における索引語の分布を、ベクトルのひとつの要素として表現することができる。これより、ランダム・プロジェクションを用いて検索モデルを構築するとき、概念ベクトルが潜在的な意味を有効にとらえることができることが分かった。

今後の研究課題としては、まず、球面 k 平均アルゴリズムは初期段階での分割に非常に大きな影響を及ぼす可能性があるため、初期分割に依存しない有効な概念ベクトルの生成方法を考慮し、より有効な次元圧縮を実現が可能であると考えられる。さらに、より有効な次元圧縮

を行うために、評価用データの解答やユーザの評価をフィードバック情報として、概念ベクトルの調節を行った検索モデル (Vogt, Cottrell, BeLew, and Bartell 1999)(Tai and Kita 2000) を構築することが挙げられる。

参考文献

- Arriaga, R. I. and Vempala, S. (1999). "An Algorithmic Theory of Learning: Robust Concepts and Random Projection." In *Proceedings of the 40th Foundations of Computer Science*, pp. 616–623.
- Blum, A., Konjevod, G., Ravi, R., and Vempala, S. (1998). "Semi-Definite Relaxations for Minimum Bandwidth and Other Vertex-Ordering Problems." In *Proceedings of the 30th ACM Symposium on the Theory of Computing*, pp. 90–99.
- Chicholm, E. and Kolda, T. G. (1998). "New Term Weighting Formulas for the Vector Space Method in Information Retrieval." Tech. rep., Oak Ridge National Laboratory, Oak Ridge, Tennessee.
- Deerwester, S., Dumais, S. T., Landauer, T. K., Furnas, G. W., and Harshman, R. A. (1990). "Indexing by Latent Semantic Analysis." *American Society for Information Science*, **41**(6), 391–407.
- Dhillon, I. S. and Modha, D. S. (1999). "Concept Decompositions for Large Sparse Text Data using Clustering." Tech. rep., IBM Almaden Research Center.
- Dumais, S. T. (1995). "Using LSI for information filtering: TREC-3 experiments." In *D. Harman (Ed.), Overview of The Third Text REtrieval Conference (TREC3) National Institute of Standards and Technology Special Publication*, pp. 219–230.
- Faloutsos, C. and Lin, K. (1995). "FastMap: A Fast Algorithm for Indexing, Data-Mining and Visualization of Traditional and Multimedia Datasets." In *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, pp. 163–174.
- Feige, U. (1998). "Approximating the Bandwidth via Volume Respecting Embeddings." In *Proceedings of the 30th ACM Symposium on the Theory of Computing*.
- Frankl, P. and Maehara, H. (1988). "The Johnson-Lindenstrauss Lemma and the Sphericity of some graphs." *Combinatorial Theory*, **B 44**, 355–362.
- Johnson, W. B. and Lindenstrauss, J. (1984). "Extensions of Lipshitz Mapping into Hilbert Space." *Contemporary Mathematics*, **26**, 189–206.
- Kleinberg, J., Papadimitriou, C., and Raghavan, P. (1998). "Segmentation problems: A micro-economic view of data mining." In *Proceedings of the 30th ACM Symposium on Theory of Computing*, pp. 473–482.

- Kolda, T. G. and O'Leary, D. P. (1998). "A Semidiscrete Matrix Decomposition for Latent Semantic Indexing in Information Retrieval." In *Proceedings of ACM Transaction on Information Systems*, Vol. 16, pp. 322–346.
- Lewis, D. D. (1991). "Evaluating Text Categorization." In *Proceedings of Speech and Natural Language Workshop*, pp. 312–318.
- Papadimitriou, C., Raghavan, P., Tamaki, H., and Vempala, S. (1998). "Latent Semantic Indexing: A Probabilistic Analysis." In *Proceedings of the 17th ACM Symposium on the Principles of Database Systems*, pp. 159–168.
- Salton, G. and McGill, M. (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill.
- Tai, X. and Kita, K. (2000). "教師あり学習によるベクトル空間モデルの精度改善." 自然言語処理研究会, 情報処理学会.
- Vempala, S. (1998). "Random Projection: A New Approach to VLSI Layout." In *Proceedings of the 39th Foundations of Computer Science, Palo Alto*, pp. 389–395.
- Vogt, C. C., Cottrell, G. W., BeLew, R. K., and Bartell, B. T. (1999). "User Lenses - Achieving 100% Precision on Frequently Asked Questions." In *Proceedings of User Modeling '99, Banff*, pp. 87–96.
- Witten, I. H., Moffat, A., and Bell, T. C. (1994). *Managing Gigabytes: Compressing and Indexing Documents and Images*. Van Nostrand Reinhold, New York.

略歴

佐々木 稔: 1973 年生. 1996 年徳島大学工学部知能情報工学科卒業. 1998 年徳島大学大学院博士前期課程修了. 同年, 徳島大学大学院博士後期課程入学, 現在に至る. 機械学習, 情報検索等の研究に従事. 情報処理学会会員.

北 研二: 1957 年生. 1981 年早稲田大学理工学部数学科卒業. 1983 年から 1992 年まで沖電気工業 (株) 勤務. この間, 1987 年から 1992 年まで ATR 自動翻訳電話研究所に出向. 1992 年 9 月から徳島大学工学部勤務. 現在, 同教授. 工学博士. 確率・統計的自然言語処理, 情報検索等の研究に従事. 情報処理学会, 電子情報通信学会, 日本音響学会, 日本言語学会, 計量国語学会, ACL 各会員.

(2000 年 4 月 14 日 受付)

(2000 年 6 月 26 日 再受付)

(2000 年 10 月 10 日 採録)