高階ランダムプロジェクションを用いた情報検索

横林　亮平†　　白井　匡人††　　三浦　孝夫†

† 法政大学 理工学部創生科学科　〒 184–8584 東京都小金井市梶野町 3-7-2
†† 島根大学大学院総合理工学研究科　情報システム学領域　〒 690–8504 島根県松江市西川津町 1060
E-mail: †ryohei.yokobayashi.2j@stu.hosei.ac.jp, ††shirai@cis.shimane-u.ac.jp, †††miurat@hosei.ac.jp

あらまし　本研究では，テンソルデータモデルを確立し，データ操作として高階データ構造に対する高速な情報検索手法を提案する．情報検索では，2 階形式 (行列形式) データを仮定することが多いが，OLAP などの実務分野では高階形式 (テンソル) を仮定する．高階データは，一般に探索データ空間が膨大となり，効率よい検索が必要である．これまで、空間の削減と検索精度の維持を目的として潜在意味索引付けやランダムプロジェクション技術が提案されている．これらは行列形式データの次元数を縮小して高速検索機構を実現し，また縮小後のデータ間誤差を一定以下に保証する．本稿では，高階データ構造にランダムプロジェクションを導入し，効率よい検索機構が実現できることをいう．同時に次元縮小後のデータ距離の誤差保証が達成できることを示す.

キーワード　情報検索，ランダムプロジェクション，次元縮小，高階データ構造，誤差保証

1. 前　書　き

近年，インターネット上のテキストを中心に様々な情報を簡単かつ迅速にアクセスできる．しかしデータが膨大であるため，これらを一貫して効率的に探索することは容易ではない．故に，多くの有用な情報はほとんど抽出されず直ちに消えていく．情報検索では従来ベクトル空間モデル（VSM）を想定してきた．データ情報の多くは，単語および文書情報をベクトル形式で記述したテキスト文から成る．このような枠組みでは情報検索においてカテゴリ，オントロジー，時間的な側面などさまざまな特性が反映されない．このため，単語の多義性に起因して，ユーザの意図と異なる文書が検索される可能性がある．例えば「APPLE」という単語には，果物または会社という異なる解釈が存在する．高階データ (テンソル) を用いてカテゴリ情報を付加すると (APPLE，果物) と (APPLE，会社) を区別でき，情報表現の幅を広げることが可能である．しかし，テンソルはデータ量が膨大であり，検索操作の実行が高価となる．このため，高階データに対する効率的な検索手法が必要となる．

文書集合に対して効率的に検索を行う方法として次元縮小がある．一般的に文書集合の単語次元は数万から数十万といった高次元となる．このため，単語次元を縮小することで効率的に検索が行える．データ集合の次元数を縮小する手法としてランダムプロジェクション [5] がある．この手法は，データ間の空間距離を誤差内に止め，その空間次元を縮小する技法である [3]. しかし 2 階形式データを対象としているため，高階データ構造に適用することができない．高階データ構造に対する次元縮小手法として高階特異値分解 (HOSVD) [7] と高階ランダムプロジェクション (HORP) [10] がある．HOSVD は，各モードに対して行列化し特異値分解を行うことで高階データの各次元を縮小する．HOSVD は，高階データに対して効率的に文書検索が行えるが，データに依存して特異値ベクトルを求める．特異値ベクトルは，特異値が少量変化するだけで大幅に変化するため，差分的に求めることができない．このため，HOSVD では分解時に用いたデータと異なるデータを扱うことができない．一方，HORP は高階データ構造の内容に独立に次元縮小を行うことができる．本研究では，高階データ構造に対する高速な検索手法を提案する．提案手法では，高階データに対して HORP を実行することで次元縮小を行う．このとき，次元縮小に伴うデータ間距離の誤差保証を証明する．

本研究の貢献は以下の 2 点である．第 1 にテンソルデータモデルを確立し，これに対する新たなデータ操作を提案している点である．第 2 に，高階データ構造の縮小に対する誤差保証を行っている点である．本研究は，提案手法を用いて情報検索を行うことで，高階データ構造に対して次元縮小後のデータ距離の誤差が保証され，効率的に情報検索が行えることを示す.

第 2 章では，テンソルデータモデルとモデル操作について述べ，第 3 章で行列の次元縮小について述べる．さらに第 4 章でテンソル次元縮小について，第 5 章では情報検索について言及する．第 6 章では実験により提案手法の有効性を示す．第 7 章で結論とする．

2. データモデル

2.1 高階データ表現

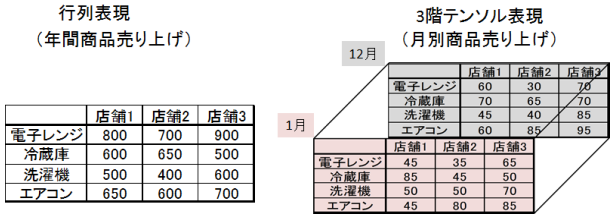
高階（テンソル）データとは，通常階数が 3 以上となるデータ構造を指す．また，1 階のテンソルをベクトル，2 階のテンソルを行列と呼ぶ．インデックス数をテンソルの次元数と呼び，階数 N のテンソル X は以下のようになる．

$$X = \{X_{e_1,..,e_N} | e_j \in I_j, j = 1,..,N\} \subseteq \mathcal{R}^{I_1 \times I_2 \times ... \times I_N}$$

k 番目のインデックス I_k を k 番目の次元と呼ぶ．また，テンソルは $X \in \mathcal{R}[I_1 \times I_2 \times ... \times I_N]$ とも表現でき，$I_1 \times I_2 \times ... \times I_N$ 上で一意に決まる．

テンソルを用いることの利点は，行列データよりも詳細な情報表現を行うことができる点とデータ操作が単純にできる点に

ある．例えば，ある店の商品売上情報をみる場合，行列ではひと月のみのデータあるいは 1 年といったまとまったデータ情報を表現することになる．しかし，テンソル表現をする場合，年間売上を月別に表示することができる．図 1 に行列とテンソルのデータ例を示す．



	店舗1	店舗2	店舗3
電子レンジ	800	700	900
冷蔵庫	600	650	500
洗濯機	500	400	600
エアコン	650	600	700

	店舗1	店舗2	店舗3
電子レンジ	60	30	70
冷蔵庫	70	65	70
洗濯機	45	40	85
エアコン	60	85	95

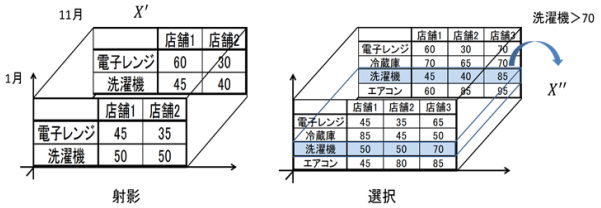
	店舗1	店舗2	店舗3
電子レンジ	45	35	65
冷蔵庫	85	45	50
洗濯機	50	50	70
エアコン	45	80	85

図 1　店舗 1,2 の商品売り上げ

2.2　テンソルデータ操作

テンソルデータ操作 [6] には，テンソル射影，テンソル選択，テンソルどうしの加算と減算，さらに乗算操作がある．また，行列およびベクトルとの変換，およびこれらとの積を導入する．以下，個々の操作を定義する．射影とは，指定した条件に従い部分的なテンソルを取り出す．例えば，図 1 のテンソルの月別売上データにおいて，電子レンジと洗濯機が 12 月以外に売れた店舗情報を取り出す．この操作は，$X' = \pi_{(\sharp 1(\text{電子レンジ, 洗濯機})\sharp 2(*)\sharp 3(12\ \text{月})^{-1})}(X)$ で表す．

選択では，更に限定的にデータの一部分を抽出する．例えば，図 1 のテンソルデータより洗濯機が 70 台以上売れた店を抽出する．データ操作は，$X'' = \sigma_{\sharp 2.\text{洗濯機}>70}(X)$ となる．射影，選択によって得られるデータ例を図 2 に示す．



	店舗1	店舗2
電子レンジ	60	30
洗濯機	45	40

	店舗1	店舗2
電子レンジ	45	35
洗濯機	50	50

	店舗1	店舗2	店舗3
電子レンジ	60	30	70
冷蔵庫	70	65	70
洗濯機	45	40	85
エアコン	60	85	95

	店舗1	店舗2	店舗3
電子レンジ	45	35	65
冷蔵庫	85	45	50
洗濯機	50	50	70
エアコン	45	80	85

図 2　射影と選択操作

次に，テンソルの演算操作について述べる．階数と次元数の等しい 2 つのテンソル $X, Y \in \mathcal{R}[I_1 \times I_2 \times \ldots \times I_N]$ についてそれぞれの要素を $x_{i_1,i_2,\ldots,i_N}$，$y_{i_1,i_2,\ldots,i_N}$ と表す．2 テンソルの和と差は，$X+Y$，$X-Y$ と表し，その要素は

$$X + Y = x_{i_1,i_2,\ldots,i_N} + y_{i_1,i_2,\ldots,i_N} \tag{1}$$

$$X - Y = x_{i_1,i_2,\ldots,i_N} - y_{i_1,i_2,\ldots,i_N} \tag{2}$$

となる．また，2 つのテンソルの積は

$$\langle X, Y \rangle = X \cdot Y = \sum_{i=1}^{I_1} \sum_{i=2}^{I_2} \ldots \sum_{i=N}^{I_N} x_{i_1,i_2,\ldots,i_N} y_{i_1,i_2,\ldots,i_N} \tag{3}$$

となる．X のノルム $||X||$ は $\sqrt{X \cdot X}$ である．

テンソルは，行列やベクトルの集合であり，ベクトル表現ができる．そこで，ベクトルの取り方を形式化したファイバとモードについて言及する [6]．ファイバは，テンソルをベクトル表現したものである．3 階のテンソルを例に挙げる．このとき，ベクトルの取り方には 3 種類存在する．3 種の表現を図 3 に示す．固定したテンソルに対して，列方向 (column)，行方向 (row)，奥行き方向 (tube) にファイバをとり，それぞれモード 1 ファイバ，モード 2 ファイバ，モード 3 ファイバと呼ぶ．テンソルを行列で表現するときは，モード n のファイバを横に並べることで行列化を行い，これを n-モード行列化と呼ぶ．n-モード行列化の例を図 4 に示す．モード n のファイバから構成された n-モード行列は，$X_{(n)}$ で表現する．n-モード行列を用いることで，テンソルは行列で表せる．

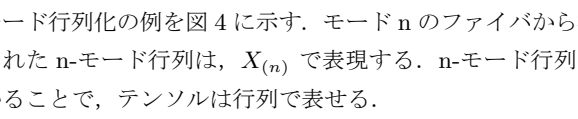


図 3　ファイバとモード

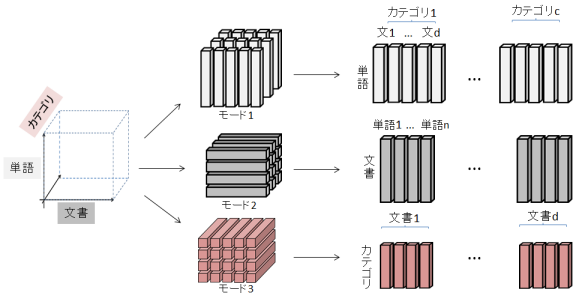


図 4　n-モード行列化

ここで，行列とテンソルの乗算 (行列テンソル積) 操作について述べる．階数 N のテンソル X ($X \in \mathcal{R}[I_1 \times I_2 \times \ldots \times I_N]$)，行列 M $\left(M \in R^{J_n \times I_n}\right)$ とし，行列テンソル積を Y とする．$Y_{(n)} = X \times_n M \in \mathcal{R}[I_1 \times I_2 \times \ldots \times I_{n-1} \times J_n \times I_{n+1} \times \ldots \times I_N]$，$j_n = 1, .., J_n$ とおくと，

$$X \times_n M = ((Y_{i_1,..,i_{n-1},j_n,i_{n+1},..,i_N}))$$

$$Y_{i_1,..,i_{n-1},j_n,i_{n+1},..,i_N} = \sum_{i_n=1}^{I_n} X[i_1,..,i_{n-1},i_n,i_{n+1},..,i_N] M[j_n,i_n]$$

となる．このとき，$J_n \leqq I_n$ であれば次元数は小さくなる．$Y_{(n)} = X \times_n M$ は以下の特性を持つ．

$$X \times_m A \times_n B = X \times_n B \times_m A, \quad m \neq n$$

$$X \times_n A \times_n B = X \times_n (BA)$$

	1月			2月		
	店舗1	店舗2	店舗3	店舗1	店舗2	店舗3
電子レンジ	45	35	65	60	30	70
冷蔵庫	85	45	50	70	65	70
洗濯機	50	50	70	45	40	85
エアコン	45	80	85	60	85	95

図 5　1,2 月店舗売り上げ

$X \in \mathcal{R}[2\times3\times2]$ のテンソル (図 5) に対して行列テンソル積を行う場合について述べる．行列 $M \in R^{3\times4}$ を用いると，

$$\begin{bmatrix} 0 & 1 & 2 & 3 \\ 3 & 2 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 45 & 35 & 65 & 60 & 30 & 70 \\ 85 & 45 & 50 & 70 & 65 & 70 \\ 50 & 50 & 70 & 45 & 40 & 85 \\ 45 & 80 & 85 & 60 & 85 & 95 \end{bmatrix}$$

$$= \begin{bmatrix} 320 & 385 & 445 & 340 & 400 & 525 \\ 335 & 245 & 365 & 365 & 260 & 435 \end{bmatrix}$$

となり，次元数が小さくなる．次に，テンソルとベクトルの乗算 (ベクトルテンソル積) について言及する．テンソル $X \in R^{I_1\times I_2\times\ldots\times I_N}$ とベクトル $V \in R^{I_n}$ とすると，ベクトルテンソル積は $Y_{(n)} = X\times_n V \in \mathcal{R}[I_1\times I_2\times\ldots\times I_{n-1}\times I_{n+1}\times\ldots\times I_N]$ であり，

$$X\times_n V = ((Y_{i_1,..,i_{n-1},i_{n+1},..,i_N}))$$

$$Y_{i_1,..,i_{n-1},i_{n+1},..,i_N} = \sum_{i_n=1}^{I_n} X[i_1,..,i_{n-1},i_n,i_{n+1},..,i_N]V[i_n]$$

となる．このとき，Y は X に対して階数が 1 つ低くなる．テンソル (図 5) に対してベクトルテンソル積を行う場合について述べる．ベクトル [1 2 3] のとき，ベクトルテンソル積を以下に示す．

$$\begin{bmatrix} 45 & 35 & 65 & 60 & 30 & 70 \\ 85 & 45 & 50 & 70 & 65 & 70 \\ 50 & 50 & 70 & 45 & 40 & 85 \\ 45 & 80 & 85 & 60 & 85 & 95 \end{bmatrix} \times \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 310 & 210 \\ 325 & 410 \\ 360 & 380 \\ 1160 & 535 \end{bmatrix}$$

また，ファイバ長により正規化した乗算を $Y = X\times_n^\circ V$ で表す．

$$X\times_n^\circ V = ((Y_{i_1,..,i_{n-1},i_{n+1},..,i_N}))$$

$$Y_{i_1,..,i_{n-1},i_{n+1},..,i_N} = (\sum_{i_n=1}^{I_n} X[i_1,..,i_{n-1},i_n,i_{n+1},..,i_N]V[i_n])/L$$

$$L = \sqrt{(\sum_{i_n=1}^{I_n} X[i_1,..,i_{n-1},i_n,i_{n+1},..,i_N])^2)||V||}$$

3.　行列次元縮小

次元縮小とはデータを低次元に射影し，データの空間量と複雑さを大幅に削減する．行列データに対してデータ構造を維持したまま射影を行う方法として，潜在意味索引付 (LSI) [8] とランダムプロジェクション (RP) [5] について述べる．

3.1　次元縮小

LSI は特異値分解 (SVD) を用いて, 元のオリジナルの行列を特異値と特異ベクトルによって近似する手法である．SVD を用いると，行列 X は以下のように分解される．

$$X^{N\times d} = U\Sigma V^t \tag{4}$$

ここで，Σ は対角行列であり，特異値の大きな順に対角要素が並ぶ．元行列 X が単語×文書の行列のとき，Σ は記事に対する単語の重み（潜在意味）を表す．小さな特異値要素は行列の近似に対する影響が少ないため削除することで低次元に射影することが可能になる．このため，LSI は精度を保ったまま大幅な次元縮小できる．しかし，特異値・特異ベクトルはデータに依存して決定するため，異なるデータを扱うことができない．

一方 RP では，データ行列 X に対して射影行列 R との積を取ることでデータ構造を保ったまま次元縮小を行う．射影行列 R は，乱数を基に構築する大きさ $k\times N$ の行列である．文書行列に対して RP を用いると，以下のように次元縮小が行われる．

$$X_{RP}^{k\times d} = R^{k\times N}X^{d\times N} \tag{5}$$

RP 行列の要素構成は以下の条件を満たす必要がある. [1] [5]

- 各要素が平均 0，分散 1 の正規分布に従う.
- 列ベクトルの長さが 1 である.
- R は直交行列である.

行列の直交化には大きな計算量を必要とする．そこで，上記の条件を近似的に満たすような RP 要素の生成方法が提案されている．RP 行列の要素 r_{ij} が

$$r_{ij} = \sqrt{3}\begin{cases} +1 & \text{確率 } 1/6 \\ 0 & \text{確率 } 2/3 \\ -1 & \text{確率 } 1/6 \end{cases} \tag{6}$$

に従うように決定される．RP 行列は，文書データに独立して要素決定が行われる．そのため，文書が変更・追加されても射影行列を変更する必要がない．また，LSI と比較して高速に次元縮小を行うことができる．

3.2　データ間距離の誤差保証

RP では，次元縮小後に得られるデータ間の距離の誤差保証がなされている [3]．例として，単語 (N) と文書 (d) の行列データを挙げる．行列データに対し，RP 行列を用いて単語の次元縮小を行う．次元縮小前の文書ベクトル u,v のユークリッド距離を $\| u-v \|^2$ で表す．また，次元縮小後の文書ベクトル Ru と Rv のユークリッド距離を，$\| Ru-Rv \|^2$ とする．次元縮小後に得られる単語の次元数が

$$k \geqq 4\left(\epsilon^2 - 2/\epsilon^3\right)^{-1}\ln N \tag{7}$$

かつ $N \geqq k$ を満たすとき，

$$(1-\epsilon)\| u-v \|^2 \leqq \| R_1u - R_1v \|^2 \leqq (1+\epsilon)\| u-v \|^2 \tag{8}$$

が成り立つ．このとき $0 < \epsilon < 1$ である．

4. テンソル次元縮小

本章では，テンソルデータにおける空間量を削減する方法について述べる．ここでは行列の次元縮小技術をテンソルに拡張し，次元縮小したテンソルのデータ間距離の誤差保証を行う．

4.1 テンソル次元縮小操作

次元縮小はデータを低次元に射影する方法である．よって，テンソルデータモデルにおいて，行列とテンソルの乗算 (行列テンソル積) 操作は次元縮小の意味をもつ．テンソルに対してデータ構造を維持したまま次元縮小を行う方法としては HOSVD [7] がある．HOSVD は，LSI 同様の分解をテンソルデータに対して行う分解手法であり，n モード行列化を用いて各モードに対し SVD を行う．このため，LSI と同様にデータ内容に依存することが問題となる．

4.2 高階ランダムプロジェクション

データ内容に独立したテンソル次元縮小方法に高階ランダムプロジェクション (HORP) がある．従来のランダムプロジェクションは，1 つのモードに対してのみ次元縮小を行う．HORP では，n 階テンソルの各モード行列に対して RP 行列をかけるため，全てのモードを低次元に射影できる．HORP によるモードの次元縮小順序は独立である．つまり，モードとそのモードにかける RP 行列が決定していれば，どのモードから次元縮小しても最終的に得られる値は一意に定まる．また，次元縮小操作が単純で高速に処理を行うことが可能である．

単語 (N)，文書 (d)，カテゴリ (c) で構成された 3 階テンソルでは，以下の式で次元縮小される．$k \ll N, l \ll d, m \ll c$ とすると，

$$X_{klm} = X \times_N R1 \times_d R2 \times_c R3 \tag{9}$$

となる．このとき，$R1 \times R2 \times R3$ の表現方法により次元縮小手法は変化する．

次に，次元縮小されたテンソルを復元する操作について提案する．HOSVD や HORP では，(9) の $R1, R2, R3$ が正規直交行列を仮定しているため，以下の式が成り立つ．

$$R^t R = I \quad (I \text{ は単位行列})$$

よって，次元縮小したテンソルに対して R^t との積をとることで次元復元が可能となる．

$$X^{N \times d \times c} \approx R_1^t \times X^{k \times d \times c} \tag{10}$$

4.3 HORP によるデータ間距離の誤差

HORP を用いて次元縮小を行う際，RP 同様データ間の距離に誤差が生じる．HORP では，RP 行列による次元縮小を複数回実行するため誤差が大きくなる．RP により次元縮小した文書ベクトルを $u' = R_1 u$，$v' = R_1 v$ としたとき，u'，v' に対してさらに RP を実行する．よって，データ間距離の誤差の上限は

$$\begin{aligned} \| R_2 u' - R_2 v' \|^2 &\leqq (1+\epsilon) \| R_1 u - R_1 v \|^2 \\ &\leqq (1+\epsilon)^2 \| u - v \|^2 \end{aligned}$$

となる．また，下限は，

$$\begin{aligned} \| R_2 u' - R_2 v' \|^2 &\geqq (1-\epsilon) \| R_1 u - R_1 v \|^2 \\ &\geqq (1-\epsilon)^2 \| u - v \|^2 \end{aligned}$$

である．従って，RP 行列を用いて次元縮小を N 回実行したとき，

$$(1-\epsilon)^N \| u - v \|^2 \leqq \| R_N a - R_N b \|^2 \leqq (1+\epsilon)^N \| u - v \|^2$$

と表すことができる．

5. テンソル情報検索

ここではテンソルにおける情報検索を形式化し，次元縮小を用いた高速な情報検索手法を提案する．また，次元縮小による類似度の誤差保証を行う．

5.1 テンソル情報検索操作

テンソルに対する情報検索は，モード n ファイバまたは n-モード行列化を用いて，検索質問との類似度を求めることにより行う．検索質問は，索引語の集合からなり，類似度は一般的に内積や余弦類似度を用いる．したがって，テンソルとベクトルの乗算操作 (ベクトルテンソル積) は類似度操作の意味を持つ．また，ファイバ長で正規化するとベクトルテンソル積は余弦類似度となる．要素が類似度となるため，要素の高い部分の属性データをランキングし表示することがテンソル情報検索となる．テンソルにおける情報検索操作の例を図 6 に示す．

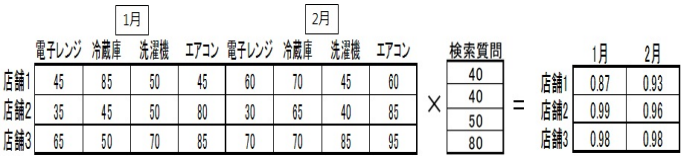


図 6　テンソル情報検索

5.2 HORP を用いたテンソル情報検索

HORP を用いた情報検索を提案する．行列データでは，次元縮小したデータを用いて情報検索を行うことで，検索精度を保ちつつ高速に検索が行えることが知られている [11]．したがって，テンソルデータに対して HORP を用いることで次元縮小を行い高速な情報検索を行う．まず，HORP で全てのモードに対して次元縮小を行いデータの次元数を大幅に圧縮する．検索を行う際は，検索に必要なモードのみを復元操作を行う．そうすることで，保持しておくべき記憶量を減少させると共に高速な検索が行える．例えば文書検索では，全軸次元縮小されたテンソルから文書軸のみ復元する．このとき検索質問とテンソルの次元数が異なるため乗算ができない．よって検索を行う際は，検索質問にあらかじめ RP 行列による射影を行うことで，類似度の計算を行う．

RP 行列が厳密な直交行列であれば $R^t R = I$ となるが，近似的に RP 行列の要素決定を行うことで誤差が生じる．RP 行列の直交性に関する誤差を ϵ とすると $R^t R = (1 \pm \epsilon) I$ と表せる．次元縮小前の類似度が，$\langle u, v \rangle = u^t v$ のとき，次元縮小後の類似度は，

$$\langle Ru, Rv\rangle = (Ru)^t Rv = u^t R^t Rv = u^t\left((1\pm\epsilon)I\right)v = u^t v \pm \epsilon\left(u^t v\right)$$

となる．よって次元縮小後の類似度は，次元縮小前の類似度に対して，

$$\frac{\langle Ru, Rv\rangle}{\langle u, v\rangle} = 1\pm\epsilon \tag{11}$$

の誤差に収まる．

6. 実　　験

実験では，NTCIR-3 を用いて文書検索に関する実験を行い，次元縮小を用いた情報検索手法の有効性を示す．テンソルには，階数 3 のテンソル (単語，文書，カテゴリ) を用いる．それぞれの軸を HORP を用いて次元縮小した後，RP 行列の転置を用いて操作に必要な軸を復元する．テンソルでは，軸 (階) 数が複数あるため，どの軸に対して検索を行うのか検索の仕方が複数存在する．本実験では，簡単のため文書軸に対して検索を行う文書検索に限定して実験を行う．

RP 行列の要素は平均 0, 分散 1 の正規分布に従い，正規直交行列になることが理想的である．しかし，近似的に RP 行列を構成するため次元縮小の際に差が生じる．よって，実データを用いて多項分布で構成した RP 行列による次元縮小操作および復元操作を行い，データ間距離誤差保証と検索精度の保証を行う．

第 1 の実験では，次元縮小したテンソルと元のテンソルデータそれぞれの文書間のユークリッド距離を比較し，誤差を測定する．単語，文書，カテゴリを次元縮小した後，文書軸を復元する．3 回の次元縮小によりデータ間距離の誤差許容範囲は $(1\pm\epsilon)^3$ である．

第 2 の実験では，HORP を用いた情報検索の検索精度の実験を行う．HORP を用いて次元縮小を行っても，精度を保ったまま元のテンソル同様に情報検索が行えることを示す．

6.1 実 験 準 備

実験に用いる NTCIR-3 は，特許検索タスクで用いた特許データコレクションである．公開特許公報全文データの kkh,JAPIO 出願抄録データの jsh, 日本国英語特許出願抄録データの paj がある．kkh と jsh は日本語で構成されており，paj は英語で構成されている．

本実験では，paj の中の 1995 年に収録された 317441 文書からランダムに抜粋した 5000 記事を使用する．5000 記事の内，妙録部分を索引語として不要語の消去及び単語のステミングを行う．さらに，本実験では出現頻度 40 以上の語だけを抽出する．最終的に得られた索引語は 1562 語となる．提案手法は，高階データを対象とするため，索引語・文書のほかに 3 つ目の階としてカテゴリを用いる．カテゴリは，各記事に記載される国際分類カテゴリに従い各文書が属するカテゴリを決定する．各記事には一つ以上のカテゴリが存在し，複数のカテゴリに属することを許す．実験に用いる 5000 記事中には，118 のカテゴリが存在する．含まれる記事数別にカテゴリを分け，平均索引語数をとると以下のようになる．

最も使用頻度の低いカテゴリには 2 つの記事が含まれ，最も使用頻度の高いカテゴリには 943 件の記事が含まれる．また，同時に出現し易いカテゴリを表 2 に示す．

表 1　カテゴリの記事数べつ平均索引語数

記事数	カテゴリ	平均索引語数
2-100 件	81 件	36.9
101-200 件	18 件	36.16
201-300 件	8 件	35.11
301-400 件	4 件	37.26
401-500 件	2 件	37.58
501-600 件	1 件	36.41
601-700 件	1 件	37.06
701-800 件	0 件	0
801-900 件	1 件	26
901-943 件	2 件	31.22

表 2　重複し易いカテゴリ

同時に出現するカテゴリ	出現頻度
A61　と　C07	85 件
C08　と　C09	48 件
B01　と　C07	32 件
G01　と　H01	28 件
B29　と　B32	24 件
G02　と　G09	24 件
B60　と　B62	23 件
B21　と　B23	22 件
B01　と　C02	21 件
C21　と　C22	21 件

以上により得られた，索引語 (単語)，文書，カテゴリを基にテンソルデータを構築する．各文書が属するカテゴリに単語の出現頻度を挿入し，それ以外の部分の要素は全て 0 とする．本実験におけるテンソルは，0 以外の要素が 257551 存在する．また，1 記事 1 カテゴリにおける最大索引語数は 67 語，最小索引語数は 4 語である．

6.2 評 価 方 法

第 1 の実験では，誤差保証した範囲内に何%の確率で収まるかで評価を行う．第 1 の実験におけるユークリッド距離の誤差許容範囲は，$(1\pm\epsilon)^3$ である．本実験データにおいて，(7) の条件を満たす ϵ は，$0.521 < \epsilon < 1$ である．従って ϵ を 0.6 から 0.1 ずつ増やし，各々のときの最小テンソルでデータ間距離を測定して元テンソルと比較する．

第 2 の実験では，次元縮小前のテンソルに対して検索質問を行う．その結果，類似度 0.7 以上となった文書を選びこれを適合文書 (正解) とする．同様に，次元縮小したテンソルに対しても検索質問を行う．類似度が 0.7 以上となる文書を選び適合文書と比較することで次元縮小による検索精度の影響を測ることができる．検索質問は 5000 記事からランダムに選んだ 300 記事を使用する．次元縮小したときのテンソルに対しては，RP 行列を用いて次元縮小した検索質問を用いる．この検索質問は，単語に関してのみ次元縮小を行う．検索精度の評価には 11 点平均適合率を適用する．

$$\text{適合率} = \frac{\text{検索した内の適合文書の数}}{\text{検索でヒットした文書}}$$

は，検索結果に含まれる正解の割合を示す．

$$\text{再現率} = \frac{\text{検索した内の適合文書の数}}{\text{適合文書}}$$

は，正解をどのくらい網羅しているかを表す．情報検索では，適合率と再現率が共に 1 となるのが理想である．しかし，現実には適合率を上げようと再現率は下がり，再現率を上げようとすれば適合率が低下する．

6.3 実験結果

第 1 の実験結果を表 3 に示す．

表 3　データ間距離の誤差

ε	平均	標準偏差	許容範囲 min max	誤差範囲外 (%)
0.6	1.353	1.038	0.064 4.096	187,462 件 (1.50)
0.7	1.355	1.042	0.027 4.913	95,105 件 (0.76)
0.8	1.355	1.044	0.008 5.832	46,990 件 (0.38)
0.9	1.355	1.048	0.001 6.859	24,245 件 (0.19)

第 1 の実験では，3 回の次元縮小と 1 回の次元復元により，次元縮小されたデータ間距離の誤差範囲は $(1 \pm \epsilon)^3$ である．データ間距離数は 12,497,500 件存在する．各々の ϵ における次元縮小可能な最も小さなテンソルは $\epsilon = 0.6$ で $137 \times 158 \times 89$, $\epsilon = 0.7$ で $112 \times 131 \times 74$, $\epsilon = 0.8$ で $99 \times 115 \times 64$, $\epsilon = 0.9$ で $91 \times 105 \times 59$ となる．データ間距離の平均誤差は，ϵ =0.6 において 1.353，ϵ =0.7 から 0.9 において 1.355 となる．

第 2 の実験における次元圧縮率と 11 点平均適合率の関係を表 4 に示す．

表 4　圧縮率と検索精度

(単語×カテゴリ) 次元数	次元縮小率	11 点適合率
1562 × 118	100%	1
1100 × 110	34.35%	0.78
900 × 100	51.17%	0.787
700 × 90	65.82%	0.785
500 × 80	78.30%	0.525
300 × 70	88.30%	0.44
100 × 60	96.20%	0.052

単語とカテゴリの次元数が 900×100 のとき，最大 11 点平均適合率 78.7 となる．また，このときの縮小率は元のテンソルの 48.83%(圧縮率 51.17%) のテンソルとなる．さらに，元のテンソルの 34.18%(圧縮率 65.82%) のテンソルにおいても 11 点平均適合率は 78%を上回る．

6.4 考　　察

第 1 の実験では，データ間距離の誤差範囲を超過するものが $\epsilon = 0.6$ のとき 1.503%，$\epsilon = 0.7$ のとき 0.761%,$\epsilon = 0.8$ のとき 0.376%,$\epsilon = 0.9$ のとき 0.194% 存在する．この原因は，RP 行列要素を近似的に構成していることにある．本来，RP 行列は正規分布を仮定しているが，多項分布を用いているため誤差範囲を超越した．

HORP を用いた検索精度では，次元数を 65.82%圧縮しても 78.5%の精度で情報検索が行える．次元数が大いほど検索精度は高くなる傾向にある．次元縮小したテンソルによる検索では，検索結果中に正解とならないものが存在する．これをノイズ (偽陽性) と呼ぶ．このノイズの内繰り返しノイズとして現れている文書があり，その文書が検索精度を低下させている．例えば，単語とカテゴリの次元数を 100×60 に次元縮小したときノイズは 113 件存在する．その中で繰り返しノイズとして出現する 8 文書あり，これを表 5 に示す．

表 5　ノイズ文書と出現回数

文書番号	ノイズとしての出現回数
212	30
170	16
295	14
231	12
293	8
174	6
253	5
240	5
8 文書	96

この 8 文書は全ノイズ中の 85%(113 件中 96 件) に影響している．この 8 件の文書の特徴は，使われている索引語が高出現頻度語となっていることである．例として，文書 174 を図 7 に示す．文書 174 では，索引語が 4 単語 (obtain, mold, base, couple) のみしか使われていない．このうち obtain，mold，base は高出現頻度順位 30 以内の単語である．高出現頻度の単語は，その他の多くの文書で使用される単語である．文書中に高出現頻度語以外の索引語が少ない場合，類似度を計算すると高出現頻度部分のみ乗算が行われ類似度が高くなる傾向にある．よって，ノイズとして繰り返し出現し，適合率の低下原因となる．

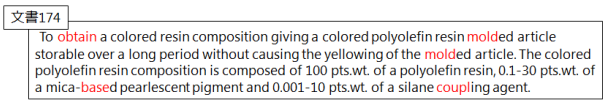


図 7　ノイズ文書の例

7. 結　　論

高階 (テンソル) データモデルを提案した．また，データ操作の一つとして高階ランダムプロジェクション (HORP) を用いた情報検索手法の提案を行った．次元縮小したテンソルを用いて，次元縮小前と同様のデータ操作が可能であることを示した．ただし，次元縮小前後のデータには誤差が存在する．よって，HORP による次元縮小したときの誤差保証を行った．次元縮小を N 回行ったときのデータ間距離は $(1 \pm \epsilon)^N$ となる．実データを用いて 3 回の次元縮小と 1 回の次元復元を行ったときのデータ間距離は，$(1 \pm \epsilon)^3$ である．近似的に要素決定を行った HORP を用いても，誤差保証範囲内におよそ 99%収まる．

テンソルの類似度操作は，次元縮小前後で誤差が $1 \pm \epsilon$ 内に収まることを保証した．HORP を用いた情報検索では，元テンソルの 34.18%(圧縮率 65.82%) のテンソルで検索精度は 78.5%となる．故に，精度を維持したまま高速な情報検索を行うことが可能となった．

文　　献

[1] Bingham, Ella, and Heikki Mannila. "Random projection in dimensionality reduction: applications to image and text data." Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2001.

[2] Bogdanova, Iva, Francisco Rincn, and David Atienza. "A multi-lead ECG classification based on random projection features." 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2012.

[3] Dasgupta, Sanjoy, and Anupam Gupta. "An elementary proof of a theorem of Johnson and Lindenstrauss." Random Structures and Algorithms 22.1 (2003): 60-65.

[4] De Lathauwer, Lieven, Bart De Moor, and Joos Vandewalle. "A multilinear singular value decomposition." SIAM journal on Matrix Analysis and Applications 21.4 (2000): 1253-1278.

[5] Kaski, Samuel. "Dimensionality reduction by random mapping: Fast similarity computation for clustering." Neural Networks Proceedings, 1998. IEEE World Congress on Computational Intelligence. The 1998 IEEE International Joint Conference on. Vol. 1. IEEE, 1998.

[6] Kolda, Tamara G., and Brett W. Bader. "Tensor decompositions and applications." SIAM review 51.3 (2009): 455-500.

[7] De Lathauwer, Lieven, Bart De Moor, and Joos Vandewalle. "A multilinear singular value decomposition." SIAM journal on Matrix Analysis and Applications 21.4 (2000): 1253-1278.

[8] Papadimitriou, Christos H., et al. "Latent semantic indexing: A probabilistic analysis." Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems. ACM, 1998.

[9] Sun, Jimeng, Dacheng Tao, and Christos Faloutsos. "Beyond streams and graphs: dynamic tensor analysis." Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2006.

[10] Sanguansat, P., "Higher-Order Random Projection for Tensor Object Recognition.", Intn'l Symp. on Communications and Information Technologies (ISCIT). IEEE, 2010.

[11] 大内　浩二, 三浦　孝夫, 塩谷　勇, "頻度分布に基づくプロジェクションを用いた文書検索," DEWS, 1C-i8, 2005.