

# オブジェクトの相対位置関係に基づく検索要求の視覚的 記述とその処理方法

渡辺正裕 吉川正俊 植村俊亮

奈良先端科学技術大学院大学 情報科学研究科

あらゆる分野で莫大な情報が生成、蓄積されているが、本研究では、巨大な情報集合から迅速かつ的確に必要なものを取り出すという要求を満たすために、直観的に情報を提供する手法を提案する。オブジェクト-特徴行列の特異値分解を利用し、データベースに格納されているオブジェクト集合を多次元ベクトル空間として視覚化する。本手法では利用者が“yardstick objects”とともに検索目標を2次元のインタフェースに記述する。これらのオブジェクトの相対位置関係が検索条件として解釈され、検索結果は特異値分解で得られる直交基底を合成して得られる2次元平面に出力される。本手法を用いると、キーワードのand/orで記述した場合に複雑になるような問合せ簡単に記述することが可能になる。

## Spatial Indication Mechanism for Exploratory Information Retrieval in Multidimensional Space

Masahiro WATANABE, Masatoshi YOSHIKAWA and Shunsuke UEMURA

Graduate School of Information Science  
Nara Institute of Science and Technology (NAIST)

In this paper, we propose a new approach to visual information retrieval. In this approach, first, an object-attribute matrix generated from an object set is decomposed using the *Singular Value Decomposition (SVD)* method. In the second step, users place “yardstick objects” by which they can specify the retrieval conditions in the 2D-space. And then, users can indicate a query object in the same 2D-space. The spatial relationships between the query object and “yardstick objects” are interpreted as the definition of the plane, or as the search conditions. Then the object information is visualized in a 2D-space using the result of SVD.

With this approach, users can easily specify retrieval conditions which will be complicated when expressed in Boolean expressions of keywords. And relevant objects that do not contain the attribute of the query can be properly characterized and identified. Such visual method realizes smooth and effective interactions between the system and users.

## 1. はじめに

これまで研究されてきたベクトルモデルを用いたフィルタリング手法では、指定された query vector と距離が一定以下のオブジェクトを、あるいは近いオブジェクトから順に検索結果とする方法が一般的であった [FD92]。

本研究では、利用者プロフィールを記述する際に利用者にとって既知であるオブジェクト (目安オブジェクトと呼ぶ) を目安にして 2 次元入力インタフェースで検索目標を指定する。この指定と目安オブジェクトとの相互位置関係を利用し、オブジェクト特徴プロフィールを射影する方法を決定する。

検索対象となるオブジェクト集合の情報内容を、多次元のベクトルを並べた行列で表現する。これに特異値分解を用いて  $k$  次元直交ベクトル空間に正規化したものをオブジェクト特徴プロフィールとする。これに対して利用者は 2 次元の入力インタフェースを用いて検索要求を記述し、利用者プロフィールとする。この二つのプロフィールを比較・照合するために、オブジェクト特徴プロフィールを利用者プロフィールと同じ 2 次元に射影する。

既存の方法では特徴空間での「距離」のみに着目し、「方向」に関しては無視していたが、本稿で提案する方法は距離だけでなく方向についても考慮に入れて指定できることを目標としている。

まず、本稿で提案する検索要求の記述方法の概要を 1. 節で説明し、2. 節で情報フィルタリングと情報検索の関係、ベクトル検索モデル、特異値分解、LSI など、本稿で提案する検索要求の記述方法の要素技術について説明する。その上で 3. 節で提案する検索要求の記述方法について述べる。

## 2. 準備

大量の情報の中から利用者の興味に合った情報を選別する技術が情報フィルタリングである [FO95][有吉 97]。情報フィルタリングは、次のように分類することができる [森田 96][MGT+87]。

### 1. Cognitive filtering

情報の内容と、利用者の情報に対する要求を記述したプロフィールを比較することでフィルタリングを行う。たとえば、自分が現在行っている仕事に関連する情報を優先的に入手するといったことがこれに該当する。

### 2. Social filtering

利用者個人と、個人の属する組織との関連に基づくフィルタリング。つまり情報の内容ではなく、情報の発信者の特徴や利用者との間の関係に基づいてフィルタリングを行う。たとえば、自分の上司からのメッセージは重要なものとして扱うといった場合がこれに該当する。

### 3. Economic filtering

情報を得ることによる利益と、情報を得るために必要な費用の比に基づいてフィルタリングを行う。ここでいう費用とは、たとえばメッセージに対する課金などのような明示的なものだけではなく、メッセージの長さやその他心理的な要因も含まれる。

本稿では文献を対象としてキーワードや単語出現頻度などを利用する実験を行い、提案する手法の有効性について

議論する。

## 2.1 情報検索と情報フィルタリング

情報フィルタリングシステムを構成する要素技術は情報検索技術と類似する点が多く、情報フィルタリングは情報検索の一形態であると考えられる [JC92]。このような前提を元に、情報検索手法の一つである、Latent Semantic Indexing を情報フィルタリングに適用した研究 [FD92][DDF+90] がいくつか知られている。LSI は、特異値分解 (Singular Value Decomposition; SVD) を文献のベクトル検索モデルに応用した索引手法である。

本研究で提案する検索要求の記述手法は、LSI で用いられているアイデアを改良したものである。

## 2.2 ベクトル検索モデル

オブジェクトや検索要求をベクトルで表現する検索モデルをベクトル検索モデルという。とくにオブジェクトが文書である場合には、出現する単語から文書のベクトルを生成する様々な方法が研究されている。検索要求ベクトル (query vector) とオブジェクトベクトルの類似度を余弦などを用いて計算し、検索要求ベクトルと類似度の大きいものから順にランク付けして検索結果とする [小川 96][FBY92]。

## 2.3 特異値分解

特異、またはそれに近い連立方程式や行列を扱う強力な手法が存在する。これらの手法は特異値分解 (Singular Value Decomposition; SVD) として知られているが、ガウス消去法や LU 分解では満足な結果が得られない場合の多くに、何が問題であるのかを診断できる [FBY92]。いくつかの場合には、診断するだけでなく数値的に解いて近似的な解を与えることもある。

特異値分解 (Singular Value Decomposition; SVD) は主成分分析、パターン認識、逆問題分析など多くの分野で用いられる手法である [岡本 92]。

$m$  個のオブジェクトと  $t$  個の特徴を仮定する。オブジェクト特徴行列  $X_0$  は  $t$  行 (オブジェクト集合に出現する各成分に対応) と、 $m$  列 (集合中の各オブジェクトに対応) から構成される。任意の行列  $X_0$  は、 $X_0 = T_0 S_0 D_0^T$  と書くことができる (図 8)。分解後の三つの行列は、次のようになっている。

- $t \times k$  行列  $T_0$ : 左特異ベクトル行列と呼ばれる列直交行列  
各特徴が  $k(X_0$  の階数) 次元の直交ベクトル (orthogonal attribute vector) で表現されている。つまり、各特徴を  $k$  個のオブジェクトとの相関で表現している。
- $k \times k$  行列  $S_0$ : 対角要素が降順に並べられた正の特異値 (singular value) であるような対角行列  
各特異値が該当する次元の「重要度」を表現している。
- $m \times k$  行列  $D_0$ : 右特異ベクトル行列と呼ばれる列直交行列  
各オブジェクトが  $k(X_0$  の階数) 次元の直交ベクトル (orthogonal object vector) で表現されている。

$T_0$ 、 $S_0$ 、 $D_0$  によって  $X_0$  を正確に再現できる。ベクトル空間内の特徴を扱う場合には各行列の小さい特異値に対応す

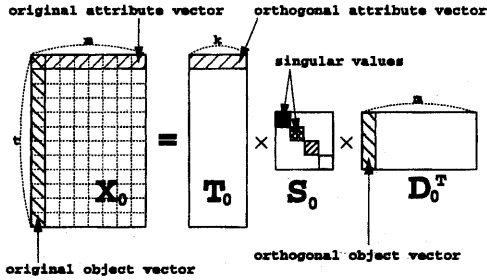


図 1 Singular Value Decomposition of  $X_0$

る部分は、不要であるばかりかノイズにもなる。つまり、数値的に小さい特異値が存在すると行列が ill-conditioned になる。LU 分解やガウス消去法を用いると正式な解を与える。しかし、誤差による影響が大きくなる。小さい特異値を 0 にして計算した方が残差 (residual) が小さい。そこで、 $S_0$  の対角成分になっている特異値のうち、小さいものを 0 で近似して、上位  $k' (\leq k)$  個の特異値を残す。0 である部分は計算上存在しないのと等価であるので、省略することができる。そうして得られる  $k' \times k'$  行列を  $S$  とすると、図 8 の右辺の  $S_0$  が図 2 の左辺の  $S$  で近似される。行列の積の定義より、 $S$  の 0 成分と積をとることを考慮すると、 $T_0$  や  $D_0^T$  の列数  $k$  についても  $k'$  に削減しても等価である。したがって、図 2 のようにして  $X$  を構成し、 $X_0$  の代わりに用いることが一般的である。 $X_0$  から  $X$  に近似するとき、行列の階数が  $k$  から  $k'$  に削減される。これは、行列に非常に小さい特異値が存在して丸め誤差の影響が大となるような場合で、安定した性能を発揮するようにするための手法である [PTVF88]。

各オブジェクトの間の相関を調べる際に、自己相関行列  $X_0^T X_0$  を求めることが一般に行われるが、これは  $X_0$  の次元数を削減した行列  $X$  の自己相関行列  $X^T X = DS^2 D^T$  を求めることで近似できる。同様に、与えられたオブジェクト集合中での特徴と特徴の相関を  $XX^T = TS^2 T^T$  で求めることができる。数万～数十万であった  $X_0$  の階数 (rank) が  $X$  を用いることによって数十～数百に削減され、このような計算を効率的に処理することができる。ベクトルを用いた近傍検索の問合せに対するオブジェクトのランキングやオブジェクト集合のクラスタリングなどにおいて、余弦類似尺度の計算やオブジェクトベクトル間の距離の計算が一般に求められるが、階数を削減した行列  $X$  を利用することにより、少ない計算量で近似的な解を求めることが可能である。

## 2.4 LSI

LSI (Latent Semantic Indexing) [FO95][FD92] は、ベクトル検索モデルにおけるオブジェクト特徴行列に特異値分解を適用することで、オブジェクト特徴空間を 200 程度の正規直交基底で索引付けするアプローチである。LSI によってオブジェクトと特徴の相関の潜在的な構造を捉え、検索の性能を改良しようとするものである。これまでの研究によって、LSI は情報検索の性能を改良することが判明している。

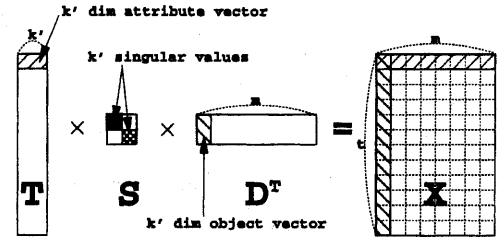


図 2 Composition of  $X$  with the reduced number of dimensions

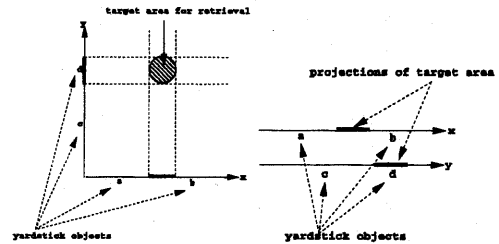


図 3 利用者の yardstick objects の指定と、縦軸・横軸への射影

## 3. 提案する入力インタフェース

### 3.1 入力インタフェース

本節では、検索目標オブジェクトを指定入力する際に、利用者にとって既知であるオブジェクトを目安にする入力インタフェースの構成方法を説明する。まず、利用者の把握のし易さと入力デバイスの都合を考慮して 2 次元の入力インタフェースを仮定するものとする。

利用者の念頭にあるオブジェクト特徴空間を 2 次元平面に射影して検索システムに入力する。このとき、2 次元平面をシステムが設定する場合と、利用者が自由に設定する場合を考えることができる。本稿では後者のタイプのインタフェースを提案する。このタイプのインタフェースを用いる場合、利用者にとって所望の情報がその 2 次元上で存在すると予想される位置に検索目標オブジェクトを指定、配置する。次にシステムがその入力を解釈して検索を行い、利用者の指定条件を満たすと考えられるオブジェクトを検索結果として出力する。

今回はオブジェクト二つを用いて、それらを両端とする線分を基準に 2 次元の座標軸を決定するアイデアを採用する。

以下では、視覚的に利用者の要求を記述するための 2 次元インタフェース (図 3 左) を構成する際に目安となるオブジェクトに付随する多次元ベクトルを用いる方法を説明する。

まず、検索対象となるオブジェクト集合から、『オブジェ

クト特徴行列」を生成する。オブジェクト特徴行列は検索のときの基準となるオブジェクト集合や問合せの検索目標オブジェクトを記述するための、文脈・背景を表現したものと考えられることができる。具体的には、オブジェクトを多次元ベクトルで表現し、そのベクトルを並べることによってオブジェクト集合を行列で表現する。この行列を特異値分解することによって、多次元オブジェクト特徴ベクトル空間を構成する正規直交基底を得る。これらの正規直交基底を用いて利用者の要求を記述する入力インタフェースを設計する。

ここで重要な点は、特異値分解によって保証されるのは各次元が直交しているということにすぎず、利用者にとって意味のある次元に分解されるわけではないという点である。また、個々の直交基底の表現能力は、オブジェクト集合の構造全体を表現する情報に比較して非常に小さいものである。そこで本研究では特異値分解で得られた直交基底をいくつか組み合わせると一つの軸として、疑似的に利用者が理解できる意味を持たせることを目標とする。このとき組み合わせる直交基底の選択する指針をいくつか提案し、それぞれ比較検討する。そのようにして構成された軸を2本組み合わせると2次元の入力インタフェースを仮定する。

注目している直交基底成分が縦軸または横軸の *yardstick objects* 線分を記述するのに向いているかどうかを判断する基準としては、以下の点が挙げられる。

1. *yardstick objects* 線分をその基底に射影したときの順序 (図3下)
2. 縦軸と横軸の基底成分が重複することを許可するかどうか
3. *yardstick objects* 線分をその基底に射影したときの大きさ

以上のことを考慮に入れると、次のような場合が考えられる。

まず、*yardstick objects* 線分を射影したときに大きさが一定のしきい値に満たないような基底を不採用とする。さらに順序に関して不適切である基底成分を次のようにして除去する。

- 射影の順序が利用者の指定と一致しないものは不適切であるとして利用しないことにする場合
  1. 縦軸と横軸の両方に対して不適
  2. 縦軸と横軸のどちらかに対して不適
  3. 縦軸と横軸のどちらに対しても適する

ということが起こり得る。ある軸に対して不適であると判断された基底は却下してその軸の成分として採用しないとする。

- 射影の順序が利用者指定と一致しないものも符号を操作することによってその軸の成分として採用することにする場合
  1. この段階で不採用になる基底成分はない

次に重複と基底の大きさについて、

- 重複を許さない場合で *yardstick objects* 線分をそれぞれ縦軸と横軸に射影したときの大きさの差が微妙である場合
  1. その直交基底成分は却下して、縦軸にも横軸にも用いない

2. 微妙であってもその違いを尊重して、その直交基底を縦軸か横軸のどちらか一方の成分として採用する

- 重複を許す場合

1. その直交基底を縦軸と横軸の両方の成分として採用する

という方針で各基底成分を分類する。

以下、3.2節で縦軸と横軸で構成する基底成分に重複がないようにする方法について説明し、3.3節で縦軸と横軸で構成する基底成分に重複を許すような方法について説明する。

### 3.2 縦軸と横軸で重複がないような解釈

本節のアルゴリズムでは、各  $l$  について、まず *yardstick objects* 線分を射影したときに大きさが一定値 ( $\sigma$ ) に達しないような基底  $bas_l$  を集合  $E$  に分類し、軸の構成に用いないようにする。次に、横軸と縦軸を構成する基底成分が重複しないようにするために、基底  $bas_l$  に対して横軸に対応する *yardstick objects* 線分を射影したときの大きさと縦軸に対応する *yardstick objects* 線分を射影したときの大きさを比較し、射影が大きい方の軸の成分として基底  $bas_l$  を用いる。基底  $bas_l$  を横軸の成分として用いる場合は集合  $A$  または  $B$  に分類し、縦軸の成分として用いる場合は集合  $C$  または  $D$  に分類する。射影の大きさが微妙な場合について縦軸と横軸のどちらにも基底  $l$  を採用しない場合は基底  $bas_l$  を集合  $E$  に分類する。

具体的には図4のアルゴリズムで直交基底成分を分類する。

#### 3.2.1 重複を許さず大小が一致しない基底は捨てる方式

図4のアルゴリズムで得られた各直交基底集合  $A, \dots, E$  は相互に重複する要素を含まない。*yardstick objects* 線分を基底  $l$  に対して射影したときの大きさに基づいて  $E$  の要素以外の直交基底ベクトルを集合  $A, B$  と  $C, D$  に分類した。さらに、*yardstick objects* を射影したときに利用者の指定の大小関係を保存するかどうかで、直交基底を分類する。

直交基底集合  $A, B$  の基底要素  $l$  に対してオブジェクトペア線分を射影したときに利用者の指定の大小関係を保存するような基底  $l$  を集合  $A$  に、保存しないような基底  $l$  を集合  $B$  に分類する。

同様にして直交基底集合  $C, D$  を集合  $C$  と集合  $D$  に分類する。

これらのうち、集合  $A$  を利用して横軸を、 $C$  を利用して縦軸を構成する。

この方法では縦軸と横軸について、基底ベクトルを選択するときに重複を許さず、利用者の指定とオブジェクトの大小関係が逆になるような基底を却下する方針 (図5) になる。

- $bas_i$  は図3で利用者が指定したオブジェクトを十分に識別し、大小関係も指定通りなので、採用
- $bas_j$  は図3で利用者が指定したオブジェクトを十分に識別するが、大小関係が逆転しているため、不

input : a set of objects( $Obj_1, \dots, Obj_N$ )  
output:  
5 sets of objects  $A = \{bas_{a_1}, bas_{a_2}, \dots, bas_{a_p}\}$ ,  $B = \{bas_{b_1}, bas_{b_2}, \dots, bas_{b_q}\}$   
,  $C = \{bas_{c_1}, bas_{c_2}, \dots, bas_{c_r}\}$   
,  $D = \{bas_{d_1}, bas_{d_2}, \dots, bas_{d_s}\}$   
,  $E = \{bas_{e_1}, bas_{e_2}, \dots, bas_{e_t}\}$

```

for l = 1 to 200
  if dist(l, Obj $_{\alpha}$ , Obj $_{\beta}$ ) <  $\sigma$  and
    dist(l, Obj $_{\gamma}$ , Obj $_{\delta}$ ) <  $\sigma$ 
    then bas $_l$  を集合 E に入れる
  else if dist(l, Obj $_{\alpha}$ , Obj $_{\beta}$ ) < dist(l, Obj $_{\gamma}$ , Obj $_{\delta}$ )
    then if (l, Obj $_{\gamma}$  > Obj $_{\delta}$ )
      then bas $_l$  を集合 A に入れる
      else bas $_l$  を集合 B に入れる
    else if (l, Obj $_{\alpha}$  > Obj $_{\beta}$ )
      then bas $_l$  を集合 C に入れる
      else bas $_l$  を集合 D に入れる

```

ここで,  $dist(l, Obj_x, Obj_y)$  は, SVD の  $l$  番目の成分に関する  $Obj_x$  と  $Obj_y$  の距離,  $(l, Obj_x > Obj_y)$  は SVD の  $l$  番目の成分に関する比較演算  $>$ ,  $\sigma$  は適当なしきい値.

図 4 各直交基底を分類するアルゴリズム

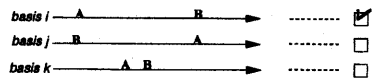


図 5 大小関係が指定入力と一致

採用

- $basis\ k$  は図 3 で利用者が指定したオブジェクトを十分に識別しないので不採用

3.2.2 重複を許さず大小が一致しない基底は符号を操作して利用する方式

3.2節のアルゴリズムで得られた直交基底集合  $A, B$  を利用して横軸を,  $C, D$  を利用して縦軸を構成する.

この方法では縦軸と横軸について, 重複を許さず, 利用者の指定とオブジェクトの大小関係が逆になっても (集合  $B, D$ ) 符号を反転させて利用する方針を採用する (図 6).

- $basis\ i$  は図 3 で利用者が指定したオブジェクトを十分に識別し, 大小関係も指定通りなので, 採用
- $basis\ j$  は図 3 で利用者が指定したオブジェクトを十分に識別するが, 大小関係が逆転しているので, 不採用
- $basis\ k$  は図 3 で利用者が指定したオブジェクトを十分に識別しないので不採用

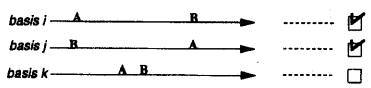


図 6 大小関係は符号操作で対応

input : a set of objects( $Obj_1 \dots Obj_N$ )  
output:  
sets of objects  $F = \{bas_{f_1}, bas_{f_2}, \dots, bas_{f_w}\}$ ,  $G = \{bas_{g_1}, bas_{g_2}, \dots, bas_{g_v}\}$   
,  $H = \{bas_{h_1}, bas_{h_2}, \dots, bas_{h_w}\}$

```

for l = 1 to 200
  if dist(l, Obj $_{\alpha}$ , Obj $_{\beta}$ ) <  $\sigma$ 
    then bas $_l$  を集合 H に入れる
  else if (l, Obj $_{\alpha}$  > Obj $_{\beta}$ )
    then bas $_l$  を集合 F に入れる
    else bas $_l$  を集合 G に入れる

```

ここで,  $dist(l, Obj_x, Obj_y)$  は, SVD の  $l$  番目の成分に関する  $Obj_x$  と  $Obj_y$  の距離,  $(l, Obj_x > Obj_y)$  は SVD の  $l$  番目の成分に関する比較演算  $>$ ,  $\sigma$  は適当なしきい値.

図 7 各直交基底を分類するアルゴリズムその 2

3.3 縦軸と横軸で重複を許すような解釈

本節のアルゴリズムでは, 各  $l$  について, まず *yardstick objects* 線分を射影したときに大きさがある一定値 ( $\sigma$ ) に達しないような基底  $bas_l$  を集合  $H$  に分類し, 軸の構成に用いないようにする. 残りの直交基底集合に対してオブジェクトペア線分を射影したときに利用者の指定の大小関係を保存するような基底  $l$  を集合  $F$  に, 保存しないような基底  $l$  を集合  $G$  に分類する.

具体的には図 7 のアルゴリズムで直交基底成分を分類する.

3.3.1 重複を許し大小が一致しない基底は捨てる方式

図 7 のアルゴリズムで得られた直交基底集合  $F$  を利用して一方の軸を構成する. 他方の軸は図 7 のアルゴリズムをもう一組の *yardstick objects* について実行して構成する.

3.3.2 重複を許し大小が一致しない基底も利用する方式

図 7 のアルゴリズムで得られた直交基底集合  $F, G$  を利用して一方の軸を構成する. 他方の軸は 3.3 節のアルゴリズムをもう一組の *yardstick objects* について実行して構成する. この方法では利用者の指定と *yardstick objects* の大小関係が逆になっても (集合  $G$ ) 符号を反転させて利用

する方針を採用する。

### 3.4 各方法の比較

#### Lemma (Lower-Bounding)[Fal96]

健全な照合問合せにおいて、誤棄却がないことを保証するためには特徴抽出関数 (feature extraction function)  $\mathcal{F}$  が、次の式を満たす必要がある。

$$D_{\text{feature}}(\mathcal{F}(O_1), \mathcal{F}(O_2)) \leq D(O_1, O_2)$$

重複を許さない 3.2.1 節, 3.2.2 節では、縦軸と横軸が特徴空間上でも直交しているため、インタフェースから受ける印象と特徴空間上のオブジェクトの配置が比較的一致すると考えられる。重複を許す 3.3.1 節, 3.3.2 節では、二次元インタフェース上では直交しているように表示されていても、特徴空間上では直交しておらず、インタフェースと実際の特徴空間に認識のずれが生じる可能性がある。また、重複を許さない 3.2.1 節, 3.2.2 節は比較的軸の構成に用いられる基底が少ないので、オブジェクトの識別力・表現力が小さくなる可能性がある。しかし、これらの 3.2.1 節, 3.2.2 節の写像は、Lower-Bounding の Lemma を満たす。それに比較して 3.3.1 節, 3.3.2 節の写像は Lower-Bounding の Lemma を満たさないため、特徴空間上の距離に基づいた検索やフィルタリングのときに検索もれの危険性を持っている。

### 3.5 座標軸のベクトルとオブジェクトのベクトル

ある一つの軸を構成するのに選択された直交基底成分の (符号つき) 総和をその軸の方向ベクトルとする。つまり、 $Axis = \sum_{k=1}^p F(a_k) \text{bas}_{a_k}$  がその軸を表すベクトルとなる。

ここで  $F(a_k)$  は yardstick objects の、基底  $a_k$  方向の大小関係で決定される、+1 か -1 の値をとる関数である。

あるオブジェクトの多次元ベクトルを  $Obj_m$ 、二次元配置したときの二次元ベクトルを  $f(Obj_m)$  とすると、 $f(Obj_m) = (\frac{Obj_m \cdot Axis_1}{|Axis_1|}, \frac{Obj_m \cdot Axis_2}{|Axis_2|})$  となる。

## 4. 結果と考察

本稿で提案した手法を文献検索に適用した結果を次に示す。

文献のベクトルが横軸と縦軸の近傍に偏って分布しているが、これは対象にした文献集合が小さすぎること、各文献が短すぎることが原因であると考えられる。今後は、もっと長い文献の大きな集合を対象にして、インタフェースの構築アルゴリズムを改良する必要がある。

### 参考文献

- [DDF+90] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman: "Indexing by Latent Semantic Analysis", in *Journal of the American Society for Information Science*, Vol. 41-6, pp. 391-407, 1990.
- [Fal96] C. Faloutsos: *Searching Multimedia Databases by Content*, Kluwer Academic, 1996.
- [FBY92] W. B. Frakes and R. Baeza-Yates eds.: *Information Retrieval - Data Structures & Algorithms* -,

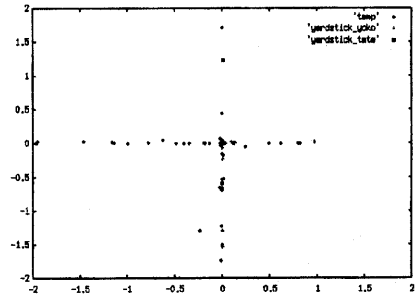


図 8 the result of our method for document retrieval

- Prentice-Hall, 1992.
- [FD92] P. W. Foltz and S. T. Dumais: "PERSONALIZED Information Delivery: AN ANALYSIS of Information Filtering Methods", in *COMMUNICATIONS OF THE ACM*, pp. 50-60, 1992.
- [FO95] C. Faloutsos and D. W. Oard: "A Survey of Information Retrieval and Filtering Methods", Technical Report CS-TR-3514, University of Maryland, August 1995.
- [JC92] N. J. Belkin and W. Croft: "Information FILTERING and Information RETRIEVAL: Two Sides of the Same Coin?", in *COMMUNICATIONS OF THE ACM*, pp. 29-38, 1992.
- [MGT+87] T. W. Malone, K. R. Grant, F. A. Turbak, S. A. Brobst, and M. D. Cohen: "Intelligent Information-Sharing Systems", *Communications of the ACM*, 30(5):390-402, May 1987.
- [PTVF88] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery: *Numerical Recipes in C*, Cambridge University Press, 1988.
- [岡本 92] 岡本 良夫: 逆問題とその解き方, オーム社, 1992.
- [小川 96] 小川 泰嗣: "情報検索の最近の動向", 日本ソフトウェア科学会 インターネットと情報検索 講習会資料, pp. 1-16, 1996.
- [森田 96] 森田 昌宏, 速水 治夫: "情報フィルタリング", 情報処理, 37(8):751-758, August 1996.
- [渡辺 96] 渡辺 正裕, 石川 佳治, 吉川 正俊, 植村 俊亮: "多次元ベクトルの視覚的探索機能を有する情報検索", 情報処理学会データベースシステム研究会研究報告, 第 96-DBS-109 巻, pp. 7-12, July 1996.
- [有吉 97] 有吉 勇介, 市山 俊治: "情報の内容と他者の評価を利用した情報フィルタリング方式", 電子情報通信学会 第 8 回 データ工学ワークショップ (DEWS'97), pp. 1-8, 1997.