Matrix computations for detecting and visualizing outlier clusters

MEI KOBAYASHI, MASAKI AONO, HIRONORI TAKEUCHI, HIKARU SAMUKAWA
IBM Research, Tokyo Research Laboratory, IBM Japan, Ltd.
1623-14 Shimotsuruma, Yamato-shi, Kanagawa-ken 242-8502 Japan
e-mail: mei@jp.ibm.com, tel: 81+462-15-4934, fax: 81+462-73-6428

Abstract - We propose two novel algorithms for detecting *major clusters* (i.e., clusters that are comprised of more than 4% of the documents in a database) and *outlier clusters* or *outliers* (i.e., clusters that are comprised of 3% to 4% of the documents in a database). And we introduce a visualization system to enable users to view, manipulate and understand output from our algorithms through a simple 3-dimensional graphical user interface. Some fairly successful techniques have been developed to identify major clusters, however these techniques often fail to identify outliers. Outliers in very large databases can represent valuable information [2], for example: unusual spending patterns due to fraudulent use of credit cards, customers who have a high probability of defaulting on loan payments, and small but emerging trends in customer claim and satisfaction. Our two algorithms are based on information retrieval algorithms which use vector space modeling: the *latent semantic indexing* (LSI) algorithm of Deerwester et al. [4] and the *covariance matrix analysis* (COV) algorithm of Kobayashi et al. [7].

Key-Words: - cluster, outlier, information retrieval, data mining, covariance matrix

1. Introduction

In recent years the volume of data stored in electronic databases has become so massive that development of systems to enable fast and accurate retrieval of information that are tailored to the interests of individual users has become imperative. Several mathematical approaches are being taken in the race to build fast, accurate and intelligent knowledge mining and management systems, one of which is vector space modeling [3], introduced by Salton and his colleagues [11] over a quarter century ago. The relevancy ranking of a document with respect to a query is determined by its so-called "*distance*" to the query vector, e.g., the angle defined by the query and each document vector. This method for ranking is impractical for very large databases since there are too many computations and subsequent comparisons.

In the late 1980's Deerwester et al. [4] proposed the *latent semantic indexing* (LSI) algorithm as a means of reducing the dimension of the document-attribute matrix to enable real-time *information retrieval* (IR) from very large databases. The fundamental idea in LSI is to model a database by an M-by-N document-attribute matrix A (the rows of which are vectors each of which represents a documents in the database) and to reduce the dimension of the IR problem to k, where $k \ll \min(M, N)$, by projecting the problem into the subspace spanned by the rows of the closest rank-k matrix to A in the Frobenius norm [5]. One of the major bottlenecks in applying LSI to massive databases (with hundreds of thousands of documents) is the need to compute the largest few hundred singular values and corresponding singular vectors of the document-attribute matrix for a database. Even though document-attribute matrices that appear in IR tend to be very sparse (usually 0.2% to 0.3% non-zero), computation of the top 200-300 singular triplets of the matrix using a powerful desktop PC becomes impossible when the number of documents exceeds several hundred thousand [6].

In this paper we review *COV*, an IR algorithm based on spectral analysis of the covariance matrix for the document vectors, that reduces the dimension of IR problems and overcomes the scalability problem associated with LSI, and we present two new algorithms for detecting major and outlier clusters in massive databases: one is based on LSI and the other on COV. The remainder of this paper is organized as follows. In the next section we review COV and discuss the underlying theoretical concepts. The third section presents applications of our work to outlier cluster detection and the fourth presents results from implementation studies and visualization of the data.

2. COVARIANCE MATRIX IR

Given a database modeled by an M-by-N document-attribute term matrix A, with M row vectors $\{d_i \mid i = 1, 2, \ldots, M\}$ representing documents, each having N dimensions representing attributes, the *covariance matrix* of the document vectors is defined as

$$C \equiv \frac{1}{M} \sum_{i=1}^{M} d_i d_i^T - \bar{d}\,\bar{d}^T \ ,$$

where d_i represents the i-th document vector and $\bar{d}$ is the component-wise average over the set of all document vectors [8], i.e., $\bar{d} = [\bar{d}_1 \ \bar{d}_2 \ \cdots \ \bar{d}_N]^T$; $d_i = [a_{i,1} \ a_{i,2} \ \cdots \ a_{i,N}]^T$, and

$$\bar{d}_j = \frac{1}{M} \sum_{i=1}^{M} a_{i,j} \ .$$

Since the covariance matrix is symmetric, positive, semi-definite, it can be decomposed into the product $C = V \, \Sigma \, V^T$, where V is an orthogonal matrix that diagonalizes C so that the diagonal entries of Σ are in monotone decreasing order going from top to bottom, i.e., diag(Σ) = $(\lambda_1, \lambda_2, \ \ldots, \lambda_N)$. To reduce the dimension of the IR problem to $k \ll M, N$, we project all of the document vectors and the query vector into the subspace spanned by the k eigenvectors $\{v_1, v_2, \ \ldots, v_k\}$ corresponding to the largest k eigenvalues $\{\lambda_1, \lambda_2, \ \ldots, \lambda_k\}$ of the covariance matrix C. Similarity ranking with respect to the modified query and document vectors is performed in a manner analogous to that before dimensional reduction, e.g., by computing the cosine of the angle defined by the query and document vectors.

Covariance matrix-based IR is similar to LSI in that it projects a very high dimensional problem into a subspace small enough to speed up computations to determine basis vectors to represent the subspace and determine relevancy rankings for IR, but large enough to retain enough information about different features of documents to facilitate accurate retrieval. The LSI and COV algorithms use different criteria to determine a subspace; LSI uses the subspace spanned by the rows of the closest rank-k matrix to A in the Frobenius norm, while COV uses the k-dimensional subspace that best represents the full data with respect to the minimum square error. Furthermore, COV shifts the origin of the coordinate system to the "center" of the subspace to spread apart documents as much as possible so that documents can be more easily be distinguished from one another.

When we performed numerical experiments in information retrieval using LSI and COV with the Reuters-21578 news database [1] [10], our results from both algorithms were very close, as expected, since successful algorithms should have similar outputs [7]. For instance, the top 10 relevancy rankings are identical and the relevance scores are within 1%-2%. Some of the rankings are switched from the 11-th ranking, but the relevancy are still very close and are within 2%-3%.

3. OUTLIER DETECTION

We present two new algorithms for detecting both major and outlier clusters in databases that are significant enhancements of the LSI and COV algorithms. Implementation studies have shown that LSI and COV are fairly successful at identifying major clusters, however, they usually fail to identify outliers. In fact the algorithms often delete information in outliers, because major clusters and their large sub-clusters dominate the subjects that will be preserved during dimensional reduction. Recently, Ando [1] proposed an algorithm that overcomes this problem in limited contexts. The main intended idea in

[1] The documents in the Reuters-21578 collection appeared on the Reuters newswire in 1987 and have been made publically available for academic research purposes.

her algorithm is to prevent major themes from dominating the process of selecting the basis vectors for the reduced dimensional subspace. This supposed to be carried out during the basis vector selection process by introducing a negative bias to documents that belong to clusters that are well-represented by basis vectors that have already been selected. The negative bias is imparted by computing the magnitude (i.e., the length in the Euclidean norm) of the *residual* of each document vector (i.e., the proportion of the document vector which cannot been represented by the basis vectors that have been selected thus far), then re-scaling the magnitude of each document vector by a power q of the magnitude of its residual. Ando's algorithm is somewhat successful in detecting clusters, however, the following problems can occur: all outliers clusters may not be identified; the procedure for finding eigenvectors may become unstable when the scaling factor is large; the basis vectors are not always orthogonal; and if the number of documents in the database is very large, the eigenvector cannot be computed on an ordinary PC.

We propose two new algorithms for detecting major and outlier clusters which overcome some of the problems associated with Ando's algorithm. The first is a significant modification of Ando's algorithm and the second is based on COV.

ALGORITHM 1 (LSI-based)

```
for $(i = 1; i \leq k; i++)\{$
    $t_{\max} = \max(|r_1|,\ |r_2|,\ \ldots,\ |r_M|)$ ;
    $q = \text{func}\ (t_{\max})$ ;
    $R_s = [\ |r_1|^q\ r_1,\ |r_2|^q\ r_2,\ \ldots,\ |r_M|^q\ r_M\ ]^T$ ;
    SVD $(R_s)$ ;   (singular value decomposition)
    $b_i' =$ the first row vector of $V^T$ ;
    $b_i =$ MGS $(b_i')$ ;   (modified Gram-Schmidt)
    $R = R - R\ b_i b_i^T$ ;   (residual matrix)
}
```

The input parameters are the document-term matrix A, the scale factor q, and the dimension k to which the IR will be reduced. The *residual matrices* are denoted by R and R_s. We set R to be A initially. After each iterative step the residual vectors are updated to take into account the new basis vector b_i.

Algorithm 1 is based on the observation that re-scaling document vectors after the computation of each basis vector in Ando's algorithm leads to the rapid diminution of documents which have even a moderate-size component in the direction of one of the first few document vectors. To understand how negative biasing can obliterate these vectors, consider the following scenario. Suppose that a document has a residual of 90% after one basis vector is computed, and q is set to be one. Before the next iteration, the vector is re-scaled to length 0.81, after two more iterations $0.81 \times 0.81 < 0.66$, and after n more iterations, less than 0.81 to the n-th power. We recognize that biasing can be useful, however the bias factor should dynamically change to take into account the length of the residual vectors after each iterative step to prevent over-biasing. More specifically, in the first step of the iteration we compute the maximum length of the residual vectors and use it to define the scaling factor q which appears in the second step.

$$q = \begin{cases} t_{\max}^{-1} & \text{if} \quad t_{\max} > 1 \\ 1 + t_{\max} & \text{if} \quad t_{\max} \approx 1 \\ 10^{t_{\max}^{-2}} & \text{if} \quad t_{\max} < 1 \end{cases}$$

As a second modification, we replace the computation of eigenvectors in Ando's algorithm with the computation of the SVD for robustness. Our third modification is the introduction of modified Gram-Schmidt orthogonalization [6] of the basis vectors b_1.

Our second algorithm for outlier detection is a modification of COV that is analogous to the modification of LSI to produce Algorithm 1. Results from our implementation studies given below indicate that our second algorithm is better than Ando's, LSI, COV, and Algorithm 1 at identifying large and multiple outlier clusters.

ALGORITHM 2 (COV-based)

```
for $(i = 1; i \leq k; i++)\{$
    $t_{\max} = \max(|r_1|,\ |r_2|,\ \ldots,\ |r_M|)$ ;
    $q = \text{func}\ (t_{\max})$ ;
    $R_s = [|r_1|^q\ r_1,\ |r_2|^q\ r_2,\ \ldots,\ |r_M|^q\ r_M]^T$ ;
    C = COV $(R_s)$ ;   (covariance matrix)
```

```
  SVD $(C)$ ;   (singular value decomposition)
  $b'_i$ = the first row vector of $V^T$ ;
  $b_i$ = MGS $(b'_i)$ ;   (modified Gram-Schmidt)
  $R = R - R\, b_i b_i^T$ ;   (residual matrix)
}
```

4. OUTLIER VISULIZATION

To test and compare the quality of results from the algorithms discussed above, we constructed a data set consisting of two large clusters (each of which have three subclusters), four outlier clusters and noise. Each large cluster has two subclusters that are twice as large as the outliers and a subcluster that is the same size as the outliers, as shown below.

CLUSTER STRUCTURE OF DATA

(total: 140 documents, 40 terms)
25 docs (Clinton cluster) - *major*
  10 docs (Clinton + Gore only) - *subcluster*
  10 docs (Clinton + Hillary only) - *subcluster*
  10 docs (Clinton + Gore + Hillary) - *subcluster*
25 docs (Java cluster) - *major*
  10 docs (Java + JSP only) - *subcluster*
  5 docs (Java + Applet only) - *subcluster*
  10 docs (Java + JSP + Applet) - *subcluster*
5 docs (Bluetooth cluster) - *outlier*
5 docs (Soccer cluster) - *outlier*
5 docs (Matrix cluster) - *outlier*
5 docs (DNA cluster) - *outlier*
70 docs *noise*

We implemented five algorithms to reduce the dimension of the document-term space: LSI, COV, Ando, and Algorithms 1 and 2. The 40-dimensional term space was reduced to six dimensions, i.e., we set $k = 6$. Table 1 summarizes clusters that were detected as basis vectors were computed. LSI did not find any outlier clusters. COV picked up some information in outliers, But failed to detect specific outliers. Ando's algorithm detected two outlier clusters: **B** (Bluetooth) and **S** (Soccer) in b_4 and the two remaining outliers **M** (Matrix) and **D** (DNA) in b_5 and b_6. Our results indicate that after the fourth iteration the lengths of the residual vectors for documents on subjects other than **M** and **D** have been given too much of a negative bias so that information in them cannot be recovered. Furthermore, they show why re-scaling using a constant factor q does not work well in the presence of multiple outliers. In contrast, Algorithms 1 and 2 successfully detect all outlier clusters. Results from using Algorithm 1 are as follows: **M** and **D** are detected by b_4; **B** and **S** by b_5; and **O** – all outliers together – by b_6. In short, all outlier clusters are detected. Results for Algorithm 2 are: **M** and **D** are detected by b_3; **B** and **S** by b_5; and **O** by b_2, b_4 and b_6, i.e., all outliers are detected, as in Algorithm 1.

TABLE 1: CLUSTERING RESULTS

	LSI	*COV*	*Ando*	*Alg.*1	*Alg.*2
b_1	**C**	**C, J**	**J**	**J**	**C, J**
b_2	**J**	**C, J**	**C**	**C**	**N, O, C, J**
b_3	**N**	**N, O**	**N**	**N**	**M, D**
b_4	**C**	**O, N**	**B, S**	**M, D**	**O**
b_5	**J**	**O, N**	**M, D**	**B, S**	**B, S**
b_6	**N**	**O, N**	**M, D**	**O**	**N, O**

C	Clinton (major) cluster
J	Java (major) cluster
N	Noise
O	Outlier clusters (all)
B	Bluetooth (outlier) cluster
S	Soccer (outlier) cluster
M	Matrix (outlier) cluster
D	DNA (outlier) cluster

Three-dimensional slices of results from Algorithm 1 are shown in Figures 1 and 2. Results from Algorithm 2 are shown in Figures 3 and 4. To enable better visualization of clusters and noise, we computed the convex hull of sets of documents which appear close together. In Figure 1 the $x-$, $y-$ and $z-$axes are the basis vectors b_1, b_2 and b_3, respectively. Both major clusters (i.e., *Clinton* and *Java*) and *noise* can be clearly seen. The coordinate axes in Figure 2 are the basis vectors b_4, b_5 and b_6, respectively. All four outlier clusters (i.e., *Bluetooth*, *Soccer*, *Matrix*, and *DNA*) can be clearly seen. In Figure 3 the coordinate axes are the basis vectors b_1, b_2 and b_3, respectively. As in Figure 1, both major clusters

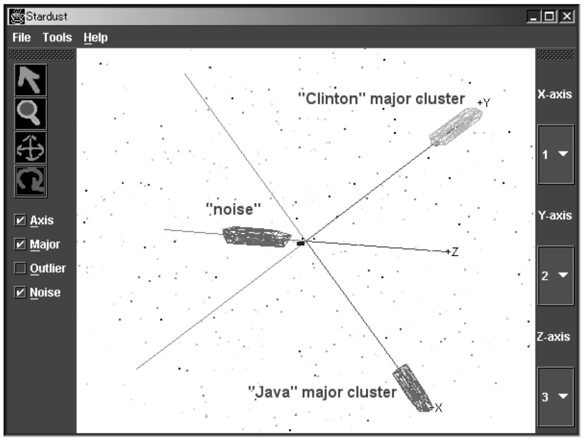


Figure 1: Three-dimensional image of major clusters and noise detected using Algorithm 1. The x-, y- and z-axes are the basis vectors b_1, b_2 and b_3, respectively.

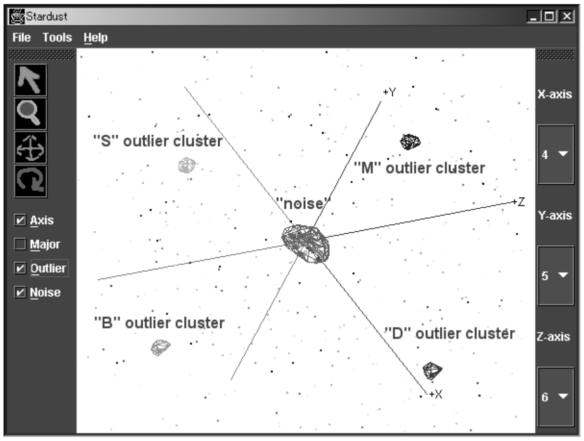


Figure 2: Three-dimensional image of outlier clusters detected using Algorithm 1. The x-, y- and z-axes are the basis vectors b_4, b_5 and b_6.

and *noise* can be clearly seen. The coordinate axes in Figure 4 are the basis vectors b_3, b_4 and b_5, respectively. All four outliers and *noise* can be clearly seen even without information from the sixth basis vector b_6.

Our visualization system allows users to select the basis vectors to be used as the x-, y-, and z-axes using the menu on the RHS, and it also includes a service to recommend sets of coordinate axes (basis vectors) that would lead to display of information about clusters – a useful option for very large databases. Going from top to bottom, the icons on the LHS allow users to: view further information (e.g., title, abstract) about specific documents, magnify/contract the graphics image, shift the image up/down or to the left/right, and rotate the image.

References

[1] R. Ando, Latent semantic space, *Proc. ACM SIGIR*, July 2000, pp. 216–223.

[2] V. Barnett, T. Lewis, *Outliers in Statistical Data*, third ed., John Wiley and Sons, 1994.

[3] M. Berry, S. Dumais, G. O'Brien, Using linear algebra for intelligent information retrieval, *SIAM Review*, Vol. 37, Dec. 1995, pp. 571–595.

[4] S. Deerwester et al., Indexing by latent semantic analysis, *J. American Society Info. Science*, Vol. 41, 1990, pp. 391–407.

[5] C. Eckart, G. Young, A principal axis transformation for non-Hermitian matrices, *Bulletin of the American Mathematical Society*, Vol. 45, 1939, pp. 118–121.

[6] G. Golub, C. Van Loan, *Matrix Computations*, third ed., John Hopkins Univ. Press, 1996.

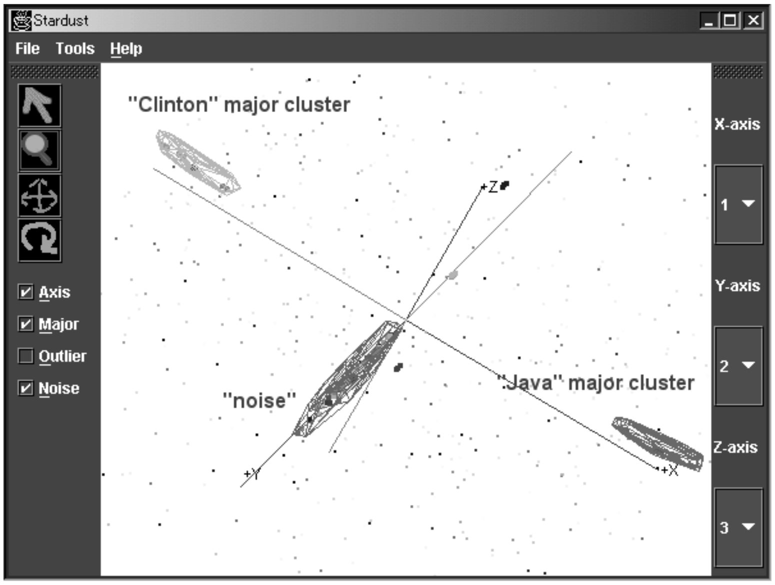
[7] M. Kobayashi, M. Aono, H. Takeuchi, H. Samukawa, Visualization and discovery of major and outlier clusters in very large databases, *IBM/TRL Research Report*, No. RT-5139, 8 pages, April 1, 2001.

[8] K. Mardia, J. Kent, J. Bibby, *Multivariate Analysis*, Academic Press, 1979.

[9] E. Rasmussen, Clustering algorithms, W. Frakes, R. Baeza-Yates (eds.), *Information Retrieval*, Prentice-Hall, 1992, Ch. 16.

[10] Reuters-21578 news data base: *http://www.research.att.com/l̃ewis*

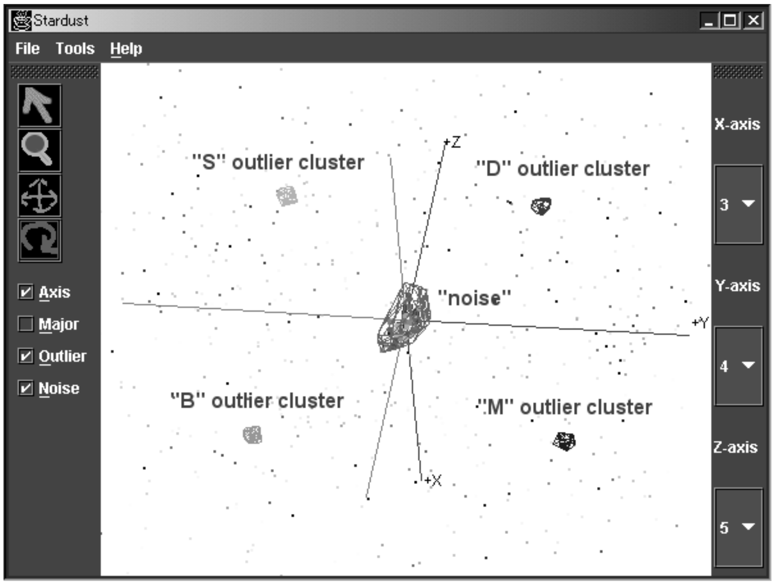
[11] G. Salton, *The SMART Retrieval System*, Prentice-Hall, 1971.



☑ Axis
☑ Major
☐ Outlier
☑ Noise

X-axis 1
Y-axis 2
Z-axis 3

Figure 3: Three-dimensional image of major clusters and noise detected using Algorithm 2. The x-, y- and z-axes are the basis vectors b_1, b_2 and b_3.



☑ Axis
☐ Major
☑ Outlier
☑ Noise

X-axis 3
Y-axis 4
Z-axis 5

Figure 4: Three-dimensional image of outlier clusters detected using Algorithm 2. The x-, y- and z-axes are the basis vectors b_3, b_4 and b_5.