November 1, 2001
RT0436
Computer Science; Mathematics 17 pages

Research Report

Covariance Matrix Analysis for Outlier Detection

Mei Kobayashi, Masaki Aono, Hironori Takeuchi, Hikaru Samukawa

IBM Research, Tokyo Research Laboratory
IBM Japan, Ltd.
1623-14 Shimotsuruma, Yamato
Kanagawa 242-8502, Japan

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Covariance Matrix Analysis for Outlier Detection

Mei Kobayashi [*] Masaki Aono, Hironori Takeuchi and Hikaru Samukawa
{ mei, aono, hironori, samukawa }@jp.ibm.com
IBM Research, Tokyo Research Laboratory, IBM Japan, Ltd.
1623-14 Shimotsuruma, Yamato-shi, Kanagawa-ken 242-8502 Japan

Abstract

In this paper we introduce *COV*, a novel information retrieval and data mining algorithm that uses vector space modeling and spectral analysis of the document vector covariance matrix to map the retrieval/mining problem into a lower dimensional space. Since the dimension of the covariance matrix depends on that of the attribute space and is independent of the number of documents, COV can be applied to databases that are too massive to be processed by methods based on the ***singular value decomposition*** (SVD) of the document-attribute matrix, such as ***latent semantic indexing*** (LSI). In addition to improved scalability, COV selects basis vectors for the lower dimensional space and shifts the origin so that subtle differences in document vectors can be more readily detected than LSI. We demonstrate the significance of this feature of COV through an important application in data mining, known as outlier cluster detection. We propose two new algorithms for detecting major and outlier clusters in databases – the first is based on LSI, the second on COV – and show through implementation studies that our algorithm based on COV outperforms the one based on LSI.

keywords: covariance matrix analysis, information retrieval, outlier cluster

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 tailored to the interests of individual users has become imperative [17]. Several mathematical approaches are being taken in the race to build fast, accurate and intelligent knowledge mining and management systems [3], [5], [14] one of which is vector space modeling [4], introduced by Salton and his colleagues [16] over

[*] contact author, e-mail: mei@jp.ibm.com, tel: 81+462-15-4934, fax: 81+462-73-6428

a quarter century ago. In vector space modeling, each document in a database is modeled by a vector, each coordinate of which represents an attribute. In *boolean* models, a coordinate of a vector is naught (when the corresponding attribute is absent) or unity (when the attribute is present). *Term weighting* is a refinement of this model that takes into account the frequency of appearance of words and their location of appearance, e.g., in the title, abstract, or section header. 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 [1]. 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. [6] 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 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 that represent the documents in the database) and to reduce the dimension of the relevancy ranking 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 [8]. One of the major bottlenecks in extending LSI to mine information from massive databases is the need to compute the largest few hundred singular values and corresponding singular vectors of the document-attribute matrix for a database [7], [9]. Even though document-attribute matrices that appear in information retrieval 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 powerful computers becomes difficult, if not impossible, when the number of documents exceeds a few hundred thousand.

In this paper we propose *COV* an information retrieval and data mining algorithm based on spectral analysis of the document vector covariance matrix, that reduces the dimension of the relevancy ranking problem and overcomes scalability problems associated with LSI [13]. Our algorithm depends on the computations of the largest few hundred eigenvalues and corresponding eigenvectors of the covariance matrix, whose dimension depends on the number of attributes (and is independent of the number of documents). In most applications the dimension of the attribute space is at most ten thousand so the spectral computations can be carried out using methods described in standard texts, such as [7], [9]. The remainder of this paper is organized as follows. In the next section we present our COV algorithm, discuss the underlying theoretical concepts and present results from our implementation studies comparing results from COV with those from LSI. In the third section we present applications of our work to data mining systems. We present two

[1] This method of measurement does not not follow the formal mathematical definition of distance.

new algorithms (one based on LSI, and a second based on COV) for detecting major and outlier clusters in massive databases and show through implementation studies that our algorithm based on COV outperforms the one based on LSI [12].

Table 1: LSI and COV query = { japan 1 car 1 market 1 } .

rel rank	% rel LSI	doc # LSI	% rel COV	doc # COV	title of document retrieved using COV
1	80.74	208	80.74	208	Japan to try to open market to US car parts
2	79.34	8647	79.32	8647	Suzuki Motor plans Hungarian joint car venture
3	72.39	19637	72.41	19637	Japan sets condition for car plant loan to Poland
4	66.88	287	66.88	287	Japan distrib may import Mazda US-made cars
5	66.10	17195	66.08	17195	EC says Japan car export restraint not enough
6	65.80	13295	65.79	13295	Oki Electric studies import of US car phones
7	65.35	10740	65.32	10740	Nissan Mexicana to take over car engine exports
8	64.34	20022	64.28	20022	Nissan starts to market remoldelled 4WD vehicles
9	63.38	16790	63.37	16763	US urges Japan to open farm market further
10	63.38	16763	62.09	16790	US urges Japan to open farm market further
11	62.82	8090	62.05	10131	Honda exports first US-made cars
12	62.10	10131	61.55	8090	Foreign share of German car market falls
13	61.58	17282	61.55	17282	EC says Japan car export restraint not enough
14	60.20	19021	60.21	19021	Japan still asks inst. to limit speculative dlrdeals
15	59.05	5326	59.03	5326	Endotronics heavy losses from withdrwl Japan
16	58.70	21276	58.61	21276	Takeshita chosen as next Japan Prime Minister
17	58.53	4329	58.46	4329	Porsche recalls 892 os its 1987 model cars
18	57.95	7907	57.94	7907	Japan opens home market to U.S. fish
19	57.84	19436	57.84	19436	Audi of America lists car prices
20	57.74	19046	57.67	13546	Japan machinery orders fall in April

2 Covariance matrix analysis for relevancy ranking

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 [13], i.e., $d_i = [a_{i,1} \; a_{i,2} \; \cdots \; a_{i,N}]^T$; $\bar{d} = [\bar{d}_1 \; \bar{d}_2 \; \cdots \; \bar{d}_N]^T$; 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 relevancy ranking 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 angle defined by the query and document vectors.

COV-based similarity ranking can be applied to much larger databases than LSI since the dimension of both the row and column of the covariance matrix is equal to the dimension of the attribute space, which is at most 10,000 or so for most databases. Computation of the largest few hundred eigenvalues and eigenvectors of a dense, symmetric, positive semi-definite matrix can be carried out using methods described in standard texts [7], [9]. COV can also be applied to databases with more than 10,000 attributes; for these databases, the covariance matrix is constructed implictly using vector-matrix multiplication followed by a symmetric Lanczos Algorithm and Sturm Sequencing [7], [9].

Covariance matrix-based information retrieval 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, but large enough to retain enough information about different features of documents to facilitate accurate relevancy ranking. 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 and clusters of documents as much as possible so that subtle differences in closely related documents can be more easily distinguished from one another, as illustrated in Figure 1. (*Clusters* are sets of documents that are grouped together based on their computed similarities; documents within the same cluster are similar to each other, while those in different clusters are less similar or dissimilar [10], [11]). Note that any pair of documents in clusters **A**, **B** and **C** define an angle less than or equal to 20-degrees in the subspace defined by LSI. Also note that the origin of the subspace defined by COV is shifted to the "center" of all of the clusters to give a more even distribution of document vectors in the subspace. In the subspace defined by COV, any pair of documents that define an angle less than or equal to 20-degrees must belong to the same cluster. Furthermore, for very large clusters, such as major cluster **A**, some pairs of document vectors belonging to different sub-clusters, e.g., $\mathbf{A}_1$ and $\mathbf{A}_2$, define an angle greater than 20 degrees.

We performed numerical relevancy ranking experiments using LSI and COV with the Reuters-21578 news database [15]. (Documents in the Reuters-21578 collection appeared on the Reuters newswire in 1987 and have been made publically available for academic research purposes.) The format of the query vector is: term 1 followed by its weighting factor, term 2 followed by its weighting factor, term 3 followed by its weighting factor. Results from both algorithms are very close, as expected, since successful algorithms should have similar outputs. In results from a representative example, shown in Table 1, 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 Major and outlier cluster detection

Clustering is the automatic creation of groups of documents, known as *clusters*, based on the computed similarities between documents in a database [11]. One of the goals of clustering is to improve the efficiency and effectiveness of information retrieval. Another goal – found in data mining applications – is the successful identification of smaller clusters which correspond to abnormalities in time-series data [2], e.g., identifying unusual spending patterns by credit card customers due to fraudulent use by unauthorized persons, identifying customers likely to default on payments, or identifying emerging trends in customer preferences and/or customer claims. Some fairly successful techniques have been developed to identify major clusters (i.e., clusters that are comprised of more than 4% of the documents in a database), however these techniques often fail to identify smaller, so-called *outlier clusters* or *outliers* (i.e., clusters that are comprised of 3% to 4% of the documents in a database). We present two new algorithms for detecting both major and outlier clusters in databases. Our algorithms are significant enhancements of the LSI and COV algorithms that are usually successful at identifying major clusters.

3.1 Prior art

Our implementation studies show that the basic LSI and COV algorithms usually fail to identify outliers, and, in fact, often delete information in these clusters, because major clusters and their

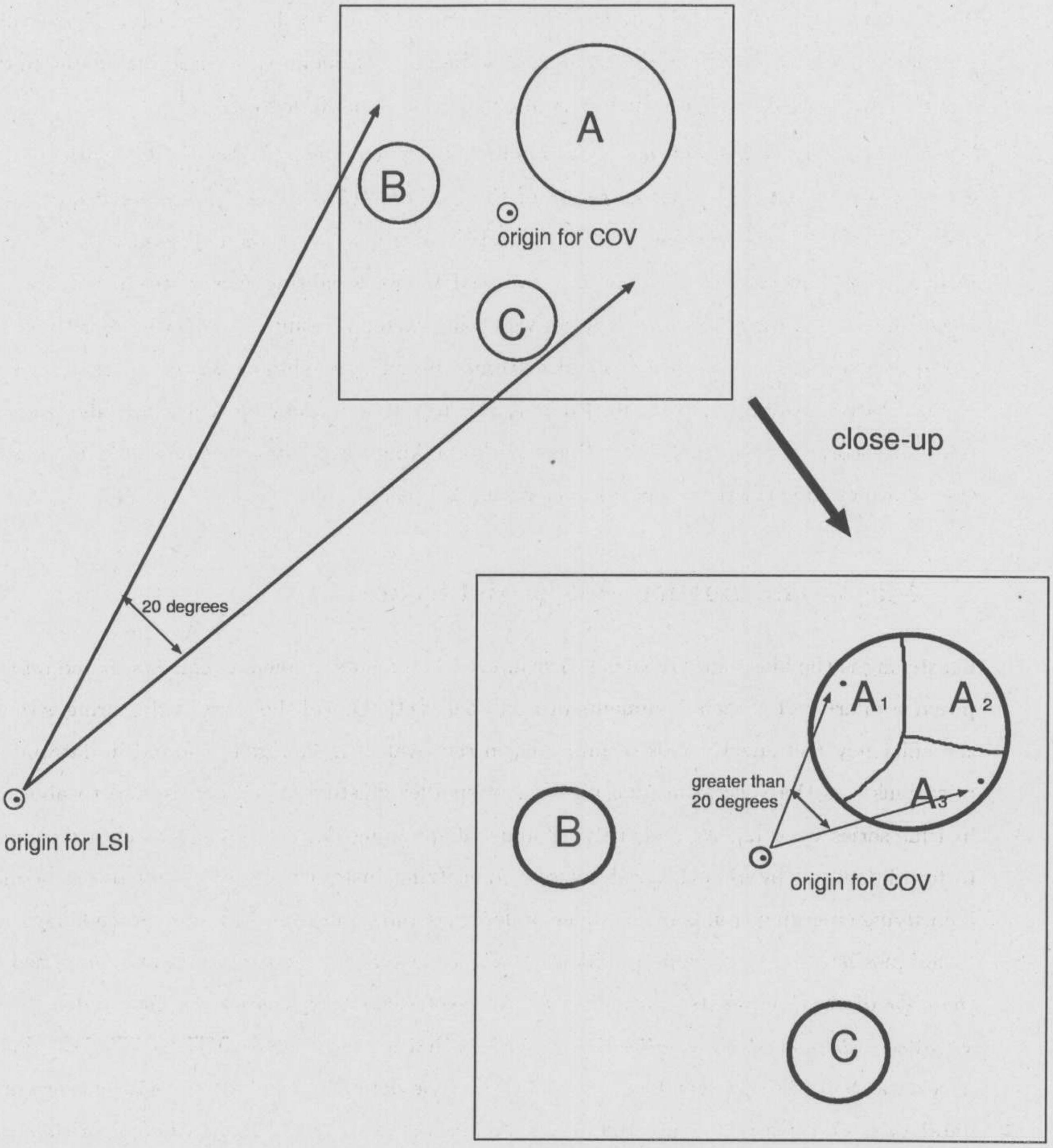


Figure 1: LSI vs. COV: Both algorithms map the relevancy ranking problem into a proper subspace of the document-attribute space. LSI does not move the origin. COV shifts the origin to the "center" of the set of basis vectors in the subspace so that document vectors are more evenly spaced apart, enabling finer detection of differences between document vectors in different clusters and subclusters.

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 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 that has not 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

```
$R = A$;
for $(i = 1; i \leq k; i++)\{$
    $R_s = [\ |r_1|^q\, r_1,\ |r_2|^q\, r_2,\ \ldots,\ |r_M|^q\, r_M\ ]^T$ ;
    $b_i$ = the first eigenvector of $R_s^T R_s$ ;
    $R = R - R\, b_i b_i^T$ ;        (the residual matrix)
$\}$
```

The input parameters are the document-term matrix A, the constant scale factor q, and the dimension k to which the relevancy ranking problem 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. After the k-th basis vector is computed, each document vector d_j in the original ranking problem is mapped to its counterpart $\hat{d_j}$ in the k-dimensional subspace as follows: $\hat{d_j} = [b_1, b_2, \ldots, b_k]^T d_j$. Similarly, the query vector is mapped to the k-dimensional subspace before relevance ranking is performed. 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 q is large; the basis vectors b_i 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.

3.2 Our algorithms

We propose two new algorithms for detecting major and outlier clusters that 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

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

Our first algorithm 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 that 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 it is re-scaled by $0.81 \times 0.81 < 0.66$, and after n more iterations it is re-scaled to 0.81 to the n-th power. Our algorithm recognizes that use of 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 that 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 [9] of the basis vectors b_1.

Algorithm 2

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

Our second algorithm for detecting outlier clusters is a modification of COV that is analogous to the modification of LSI that was performed to produce Algorithm 1. In this second algorithm, the SVD of the covariance matrix associated with the document vectors is computed (in lieu of the residual matrix in 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.

3.3 Implementation experiments

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

140 documents, 40 terms

- 25 docs (Clinton cluster) - *major*
  - 10 docs (Clinton + Al Gore only) - *subcluster*
  - 10 docs (Clinton + Hillary only) - *subcluster*
  - 10 docs (Clinton + Al 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*

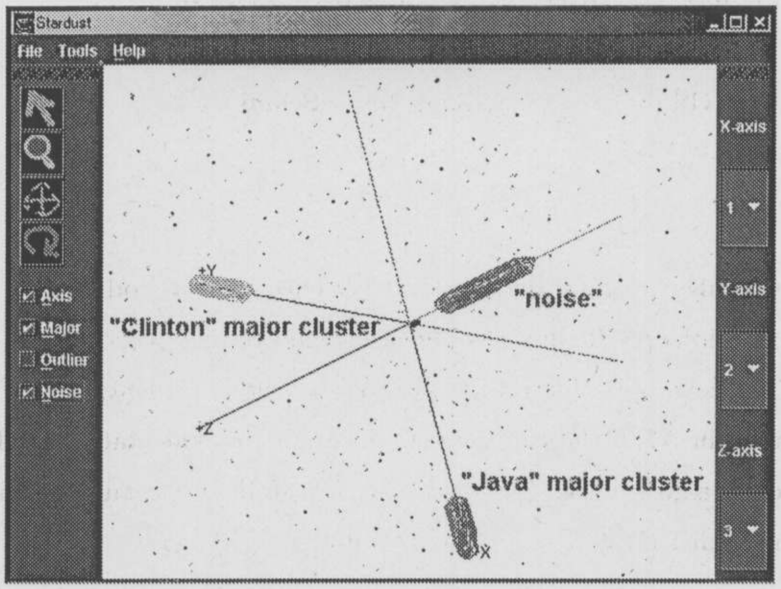


Figure 2: Three-dimensional graph of major clusters and noise detected using Algorithm 1. The x-, y- and z-axes are the basis vectors b_1, b_2 and b_3, respectively (as listed in Table 2).

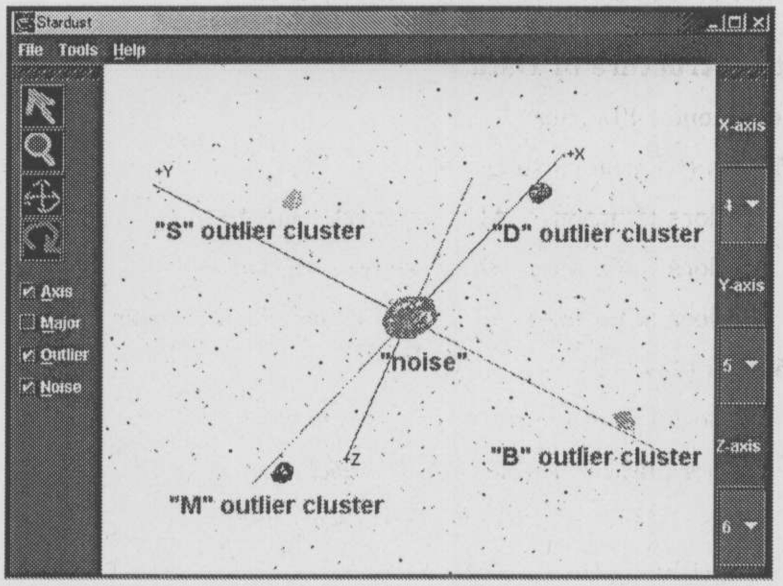
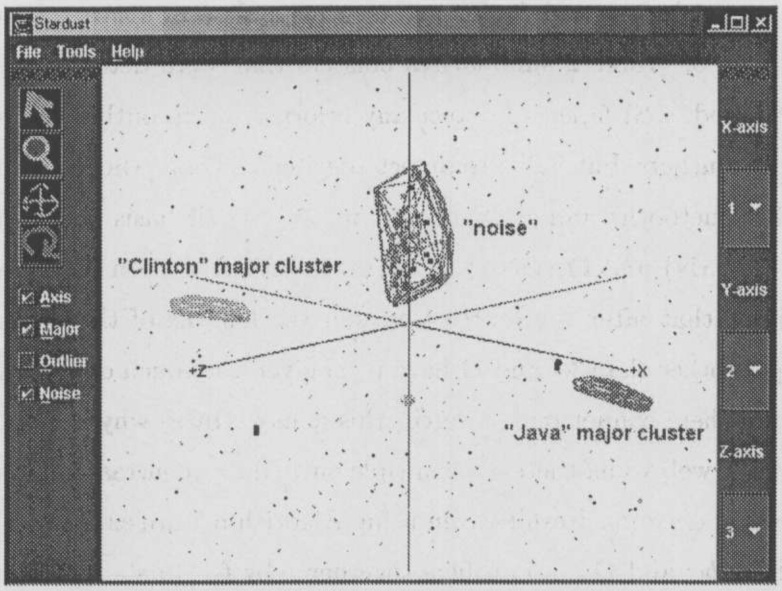


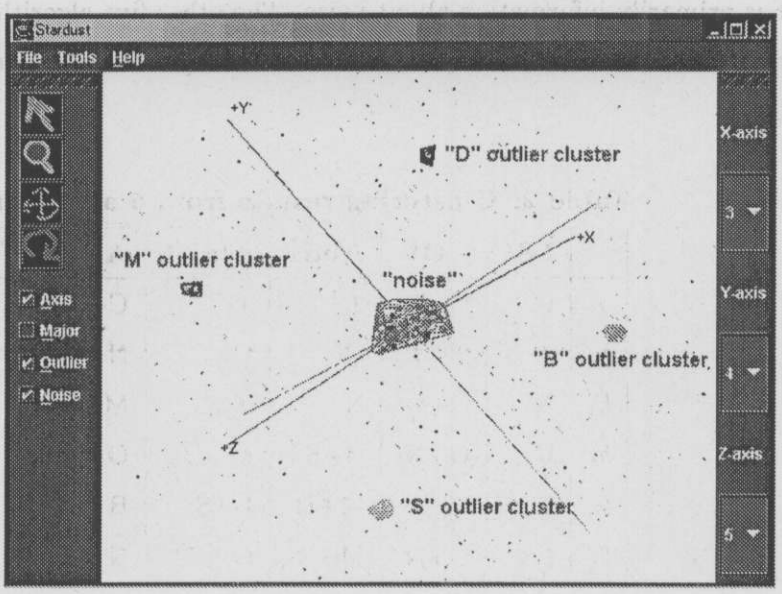
Figure 3: Three-dimensional graph of outlier clusters detected using Algorithm 1. The x-, y- and z-axes are the basis vectors b_4, b_5 and b_6, respectively (as listed in Table 2).



☑ Axis
☑ Major
□ Outlier
☑ Noise

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

Figure 4: Three-dimensional graph 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, respectively (as listed in Table 2).



☑ Axis
□ Major
☑ Outlier
☑ Noise

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

Figure 5: Three-dimensional graph of outlier clusters detected using Algorithm 2. The x-, y- and z-axes are the basis vectors b_3, b_4 and b_5, respectively (as listed in Table 2).

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 2 summarizes clusters that were detected as basis vectors b_1, b_2, ... , b_6 were computed. LSI failed to detect any information in 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 in the fourth basis vector b_4 and the two remaining outliers **M** (Matrix) and **D** (DNA) in both the fifth and sixth basis vectors b_5 and b_6. Our also results indicate that after the fourth iteration the lengths of the residual vectors for documents covering topics other than **M** and **D** have been given too much of a negative bias so any remaining information in them cannot be recovered; this demonstrates why re-scaling using a constant factor q does not work well when there are 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 evenly. 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 evenly detected, as in Algorithm 1.

Algorithm 2 is best at selecting basis vectors that contain cluster information early on: the first two basis vectors contain information about all of the major and outlier clusters and noise; the third, fourth and fifth basis vectors have more specific information about outliers; and the sixth basis vector is primarily information about noise. The other five algorithms do not detect detailed information on all outliers until six basis vectors are computed (if information on outliers was not decimated).

Table 2: Clustering results from 5 algorithms

	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

In Table 2 (above), **C** represents the major cluster *Clinton*, **J** the major cluster *Java*, **N** *Noise*, **B** the outlier cluster *Bluetooth*, **S** the outlier cluster *Soccer*, **M** the outlier cluster *Matrix*, **D** the outlier cluster *DNA*, and **0** the set of all outlier clusters. Three-dimensional slices of results from Algorithm 1 are shown in Figures 2 and 3. Results from Algorithm 2 are shown in Figures 4 and

5. To enable better visualization of clusters and noise, we computed the convex hull of sets of documents which appear close together. In Figure 2 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 $x-$, $y-$ and $z-$axes in Figure 3 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 4 the $x-$, $y-$ and $z-$axes are the basis vectors b_1, b_2 and b_3, respectively. As in Figure 2, both major clusters and *noise* can be clearly seen. The $x-$, $y-$ and $z-$axes in Figure 5 are the basis vectors b_3, b_4 and b_5, respectively. All four outlier clusters and *noise* can be clearly seen even without information from the sixth basis vector b_6.

4 Conclusion

In this paper we propose three new algorithms: one named COV for information retrieval and relevancy ranking based on analysis of the covariance matrix for documents vectors in a database, and two new algorithms for the detection of multiple outlier clusters. 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. The cluster detection algorithms, which are based on latent semantic indexing (LSI) and COV, reduce the dimension of document-term space to speed up ranking, retrieval and clustering. LSI and COV successfully detect large clusters, but they inadvertently discard information in outlier clusters during dimensional reduction. Recently Ando proposed an algorithm that modifies LSI so that outliers will be retained by introducing a negative bias to directions that are already represented during the computation of basis vectors of the subspace to which the relevancy ranking problem will be mapped. However, the algorithm has some drawbacks. For instance, it can fail to find all clusters because the bias factor is not well-controlled; it may be numerically unstable; the basis vectors it computes may not be orthogonal; and it cannot process massive document-term matrices. The algorithms we propose overcome these problems. Our implementation studies which compare LSI, COV, Ando's algorithm and our two algorithms using data with known cluster structure indicate that our algorithms are better at finding all major and outlier clusters.

Acknowledgements: The authors would like to acknowledge many helpful conversations with our colleagues in the Unstructured Information Management and Graphics Projects at IBM Research.

References

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

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

[3] M. Berry (ed.), SIAM, Philadelphia, PA (2001).

[4] M. Berry, S. Dumais, G. O'Brien, *Computational Information Retrieval*, SIAM Review, **37**, pp. 571–595 (Dec. 1995).

[5] F. Crestani, M. Lalmas, C. van Rijsbergen, I. Campbell, *'Is this document relevant ? ... probably': a survey of probablistic models in information retrieval*, ACM Computing Surveys, **30**, 4 (Dec. 1998).

[6] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, R. Harshman, *Indexing by latent semantic analysis*, J. Amer. Soc. Info. Science, **41**, pp. 391–407 (1990).

[7] J. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, PA (1997).

[8] C. Eckart, G. Young, *A principal axis transformation for non-Hermitian matrices*, Bull. Amer. Math. Soc., **45**, pp. 118–121 (1939).

[9] G. Golub, C. Van Loan, *Matrix Computations*, third ed., John Hopkins Univ. Press, Baltimore, MD (1996).

[10] A. Jain, R. Dubes, *Algorithms for Clustering Data*, Prentice-Hall, Englewood Cliffs, NJ (1988).

[11] L. Kaufman, P. J. Rousseeuw, *Finding Groups in Data: an Introduction to Cluster Analysis*, John Wiley and Sons, NY (1990).

[12] M. Kobayashi, M. Aono, H. Samukawa, H. Takeuchi, *Information retrieval apparatus for accurately detecting multiple outlier clusters*, patent, filed (July 5, 2001).

[13] M. Kobayashi, L. Malassis, H. Samukawa, *Retrieval and ranking of documents from a database*, patent, filed (June 12, 2000).

[14] M. Kobayashi, K. Takeda, *Information retrieval on the Web*, ACM Computing Surveys, **32** (2), pp. 144–173 (June 2000).

[15] Reuters-21578 document set: *www.research.att.com/l̃ewis*

[16] G. Salton, *A comparison between manual and automatic indexing methods*, American Documentation, **20** (1), pp. 61–71 (1969).

[17] I. Witten, A. Moffat, T. Bell, *Managing Gigabytes*, Van Nostrand Reinhold, NY (1994).