Random projection in dimensionality reduction: Applications to image and text data

Ella Bingham and Heikki Mannila*
Laboratory of Computer and Information Science
Helsinki University of Technology
P.O. Box 5400, FIN-02015 HUT, Finland
ella@iki.fi, Heikki.Mannila@hut.fi

ABSTRACT

Random projections have recently emerged as a powerful method for dimensionality reduction. Theoretical results indicate that the method preserves distances quite nicely; however, empirical results are sparse. We present experimental results on using random projection as a dimensionality reduction tool in a number of cases, where the high dimensionality of the data would otherwise lead to burdensome computations. Our application areas are the processing of both noisy and noiseless images, and information retrieval in text documents. We show that projecting the data onto a random lower-dimensional subspace yields results comparable to conventional dimensionality reduction methods such as principal component analysis: the similarity of data vectors is preserved well under random projection. However, using random projections is computationally significantly less expensive than using, e.g., principal component analysis. We also show experimentally that using a sparse random matrix gives additional computational savings in random projection.

Keywords

random projection, dimensionality reduction, image data, text document data, high-dimensional data

1. INTRODUCTION

In many applications of data mining, the high dimensionality of the data restricts the choice of data processing methods. Such application areas include the analysis of market basket data, text documents, image data and so on; in these cases the dimensionality is large due to either a wealth of alternative products, a large vocabulary, or the use of large image windows, respectively. A statistically optimal way of dimensionality reduction is to project the data onto a lower-dimensional orthogonal subspace that captures as much of the variation of the data as possible. The best (in mean-square sense) and most widely used way to do this is principal component analysis (PCA); unfortunately it is quite expensive to compute for high-dimensional data sets. A computationally simple method of dimensionality reduction that does not introduce a significant distortion in the data set would thus be desirable.

In random projection (RP), the original high-dimensional data is projected onto a lower-dimensional subspace using a random matrix whose columns have unit lengths. RP has been found to be a computationally efficient, yet sufficiently accurate method for dimensionality reduction of high-dimensional data sets. While this method has attracted lots of interest, empirical results are sparse.

In this paper we give experimental results on using RP as a dimensionality reduction tool on high-dimensional image and text data sets. In both application areas, random projection is compared to well known dimensionality reduction methods. We show that despite the computational simplicity of random projection, it does not introduce a significant distortion in the data.

The data sets used in this paper are of very different natures. Our image data is from monochrome images of natural scences. An image is presented as a matrix of pixel brightness values, the distribution of which is generally approximately Gaussian: symmetric and bell-shaped. Text document data is presented in vector space [25], in which each document forms one d-dimensional vector where d is the vocabulary size. The i-th element of the vector indicates (some function of) the frequency of the i-th vocabulary term in the document. Document data is often highly sparse or peaked: only some terms from the vocabulary are present in one document, and most entries of the document vector are zero. Also, document data has a nonsymmetric, positively skewed distribution, as the term frequencies are nonnegative. It is instructive to see how random projection works as a dimensionality reduction tool in the context of these two very different application areas.

We also present results on images corrupted by noise, and our experimental results indicate that random projection is not sensitive to impulse noise. Thus random projection is a promising alternative to some existing methods in noise reduction (e.g. median filtering), too.

This paper is organized as follows. At the end of this introduction we discuss related work on random projections and similarity search. Section 2 presents different dimensionality

*On leave at Nokia Research Center

KDD 01 San Francisco CA USA

reduction methods. Section 3 gives the experimental results of dimensionality reduction on image data, and Section 4 on text data. Finally, Section 5 gives a conclusion.

1.1 Related work

Papadimitriou et al. [22] use random projection in the preprocessing of textual data, prior to applying LSI. They present experimental results on an artificially generated set of documents. In their approach, the columns of the random projection matrix are assumed strictly orthogonal, but actually this need not be the case, as we shall see in our experiments.

Kaski [17, 16] has presented experimental results in using the random mapping in the context of the WEBSOM[1] system. Kurimo [20] applies random projection to the indexing of audio documents, prior to using LSI and SOM. Kleinberg [19] and Indyk and Motwani [14] use random projections in nearest-neighbor search in a high dimensional Euclidean space, and also present theoretical insights. Dasgupta [6, 7] has used random projections in learning high-dimensional Gaussian mixture models. Other applications of random projection include e.g. [4, 28].

The problems of dimensionality reduction and similarity search have often been addressed in the information retrieval literature, and other approaches than random projection have been presented. Ostrovsky and Rabani [21] give a dimension reduction operation that is suitable for clustering. Agrawal et al. [3] map time series into frequency domain by the discrete Fourier transform and only retain the first few frequencies. Keogh and Pazzani [18] reduce the dimension of time series data by segmenting the time series into sections and indexing only the section means. Aggarwal et al. [2] index market basket data by a specific signature table, which easens the similarity search. Wavelet transforms ([12, 27] etc.) are a common method of signal compression.

2. METHODS FOR DIMENSIONALITY REDUCTION

2.1 Random projection

In random projection, the original d-dimensional data is projected to a k-dimensional ($k << d$) subspace through the origin, using a random $k \times d$ matrix R whose columns have unit lengths. Using matrix notation where $X_{d \times N}$ is the original set of N d-dimensional observations,

$$X^{RP}_{k \times N} = R_{k \times d} X_{d \times N} \quad (1)$$

is the projection of the data onto a lower k-dimensional subspace. The key idea of random mapping arises from the Johnson-Lindenstrauss lemma [15]: if points in a vector space are projected onto a randomly selected subspace of suitably high dimension, then the distances between the points are approximately preserved. For a simple proof of this result, see [10, 8].

Random projection is computationally very simple: forming the random matrix R and projecting the $d \times N$ data matrix X into k dimensions is of order $O(dkN)$, and if the data matrix X is sparse with about c nonzero entries per column, the complexity is of order $O(ckN)$ [22].

Strictly speaking, (1) is not a projection because R is generally not orthogonal. A linear mapping such as (1) can cause significant distortions in the data set if R is not orthogonal. Orthogonalizing R is unfortunately computationally expensive. Instead, we can rely on a result presented by Hecht-Nielsen [13]: in a high-dimensional space, there exists a much larger number of almost orthogonal than orthogonal directions. Thus vectors having random directions might be sufficiently close to orthogonal, and equivalently $R^T R$ would approximate an identity matrix. In our experiments, the mean squared difference between $R^T R$ and an identity matrix was about $1/k$ per element.

When comparing the performance of random projection to that of other methods of dimensionality reduction, it is instructive to see how the similarity of two vectors is distorted in the dimensionality reduction. We measure the similarity of data vectors either as their Euclidean distance or as their inner product. In the case of image data, Euclidean distance is a widely used measure of similarity. Text documents, on the other hand, are generally compared according to the cosine of the angle between the document vectors; if document vectors are normalized to unit length, this corresponds to the inner product of the document vectors.

We write the Euclidean distance between two data vectors x_1 and x_2 in the original large-dimensional space as $||x_1 - x_2||$. After the random projection, this distance is approximated by the scaled Euclidean distance of these vectors in the reduced space:

$$\sqrt{d/k}\,||Rx_1 - Rx_2|| \quad (2)$$

where d is the original and k the reduced dimensionality of the data set. The scaling term $\sqrt{d/k}$ takes into account the decrease in the dimensionality of the data: according to the Johnson-Lindenstrauss lemma, the expected norm of a projection of a unit vector onto a random subspace through the origin is $\sqrt{k/d}$ [15].

The choice of the random matrix R is one of the key points of interest. The elements r_{ij} of R are often Gaussian distributed, but this need not be the case. Achlioptas [1] has recently shown that the Gaussian distribution can be replaced by a much simpler distribution such as

$$r_{ij} = \sqrt{3} \cdot \begin{cases} +1 & \text{with probability } \frac{1}{6} \\ 0 & \text{with probability } \frac{2}{3} \\ -1 & \text{with probability } \frac{1}{6}. \end{cases} \quad (3)$$

In fact, practically all zero mean, unit variance distributions of r_{ij} would give a mapping that still satisfies the Johnson-Lindenstrauss lemma. Achlioptas' result means further computational savings in database applications, as the computations can be performed using integer arithmetics. In our experiments we shall use both Gaussian distributed random matrices and sparse matrices (3), and show that Achlioptas' theoretical result indeed has practical significance. In context of the experimental results, we shall refer to RP when the projection matrix is Gaussian distributed and SRP when the matrix is sparse and distributed according to (3). Otherwise, the shorthand RP refers to any random projection.

2.2 PCA, SVD and LSI

In principal component analysis (PCA), the eigenvalue decomposition of the data covariance matrix is computed as $\mathrm{E}\{XX^T\} = E \Lambda E^T$ where the columns of matrix E are the eigenvectors of the data covariance matrix $\mathrm{E}\{XX^T\}$ and Λ is a diagonal matrix containing the respective eigenvalues.

[1] See `http://websom.hut.fi/websom/`

If dimensionality reduction of the data set is desired, the data can be projected onto a subspace spanned by the most important eigenvectors:

$$X^{PCA} = E_k^T X \tag{4}$$

where the $d \times k$ matrix E_k contains the k eigenvectors corresponding to the k largest eigenvalues. PCA is an optimal way to project data in the mean-square sense: the squared error introduced in the projection is minimized over all projections onto a k-dimensional space. Unfortunately, the eigenvalue decomposition of the data covariance matrix (whose size is $d \times d$ for d-dimensional data) is very expensive to compute. The computational complexity of estimating the PCA is $O(d^2N) + O(d^3)$ [11]. There exists computationally less expensive methods [26, 24] for finding only a few eigenvectors and eigenvalues of a large matrix; in our experiments, we use appropriate Matlab routines to realize these.

A closely related method is singular value decomposition (SVD): $X = USV^T$ where orthogonal matrices U and V contain the left and right singular vectors of X, respectively, and the diagonal of S contains the singular values of X. Using SVD, the dimensionality of the data can be reduced by projecting the data onto the space spanned by the left singular vectors corresponding to the k largest singular values:

$$X^{SVD} = U_k^T X \tag{5}$$

where U_k is of size $d \times k$ and contains these k singular vectors. Like PCA, SVD is also expensive to compute. There exists numerical routines such as the power or the Lanczos method [5] that are more efficient than PCA for sparse data matrices X, and that is why we shall use SVD instead of PCA in the context of sparse text document data. For a sparse data matrix $X_{d \times N}$ with about c nonzero entries per column, the computational complexity of SVD is of order $O(dcN)$ [22].

Latent semantic indexing (LSI) [9, 22] is a dimensionality reduction method for text document data. Using LSI, the document data is presented in a lower-dimensional "topic" space: the documents are characterized by some underlying (latent, hidden) concepts referred to by the terms. LSI can be computed either by PCA or SVD of the data matrix of N d-dimensional document vectors.

2.3 Discrete cosine transform

Discrete cosine transform (DCT) is a widely used method for image compression and as such it can also be used in dimensionality reduction of image data. DCT is computationally less burdensome than PCA and its performance approaches that of PCA. DCT is also optimal for human eye: the distortions introduced occur at the highest frequencies only, and the human eye tends to neglect these as noise. DCT can be performed by simple matrix operations [23, 27]: an image is transformed to the DCT space and dimensionality reduction is done in the inverse transform by discarding the transform coefficients corresponding to the highest frequencies. Computing the DCT is not data-dependent, in contrast to PCA that needs the eigenvalue decomposition of data covariance matrix; that is why DCT is orders of magnitude cheaper to compute than PCA. Its computational complexity is of the order $O(dN \log_2(dN))$ for a data matrix of size $d \times N$ [27].

3. RESULTS ON IMAGE DATA

The data set consisted of $N = 1000$ image windows drawn from 13 monochrome images[2] of natural scenes.The sizes of the original images were 256×256 pixels, and windows of size 50×50 were randomly drawn from the images. Each image window was presented as one d-dimensional column vector ($d = 2500$).

3.1 Noiseless image data

When comparing different methods for dimensionality reduction, the criteria are the amount of distortion caused by the method and its computational complexity. In the case of image data we measure the distortion by comparing the Euclidean distance between two dimensionality reduced data vectors to their Euclidean distance in the original high-dimensional space. In the case of random projection, the Euclidean distance in the reduced space is scaled as shown in (2); with other methods, no scaling is performed.

We first tested the effect of the reduced dimensionality using different values of k in $[1, 800]$. At each k, the dimensionality reducing matrix operation was computed anew. Figure 1 shows the error in the distance between members of a pair of data vectors, averaged over 100 pairs. The results of random projection with a Gaussian distributed random matrix (RP), random projection with a sparse random matrix as in (3) (SRP), principal component analysis (PCA) and discrete cosine transform (DCT) are shown, together with their 95 per cent confidence intervals.

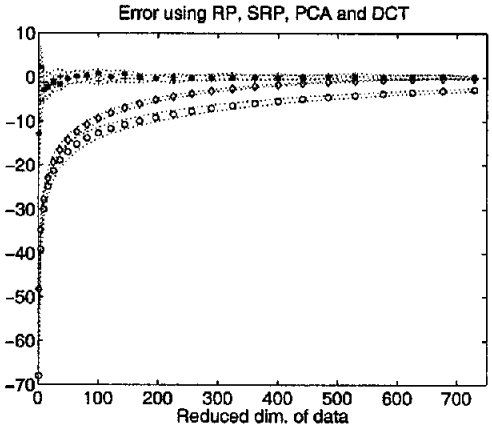

Figure 1: The error produced by RP (+), SRP (∗), PCA (◇) and DCT (○) on image data, and 95 % confidence intervals over 100 pairs of data vectors.

In Figure 1 it is clearly seen that random projection (RP and SRP) yields very accurate results: dimensionality reduction by random projection does not distort the data significantly more than PCA. At dimensions $k > 600$, random projection and PCA give quite accurate results but the error produced by DCT is clearly visible. At smaller dimensions also PCA distorts the data. This tells us that the variation in the data is mostly captured by the first 600 principal components, because the error in PCA is dependent on the sum of omitted eigenvalues, and k is equal to the number of eigen-

[2]Available from
`http://www.cis.hut.fi/projects/ica/data/images/`

values retained. In contrast, the random projection method continues to give accurate results until $k = 10$. One explanation for the success of random projection is the J-L scaling term $\sqrt{d/k}$ (Formula (2)), which takes into account the decrease in the dimensionality. In PCA, such scaling would only be useful in the smallest dimensions but a straightforward rule is difficult to give.

Another point of interest is the computational complexity of the methods. Figure 2 shows the number of Matlab's floating point operations needed when using RP, SRP, PCA or DCT in dimensionality reduction, in a logarithmic scale. It can be seen that PCA is significantly more burdensome than random projection or DCT. (In the case of DCT, only the chosen data vectors were transformed instead of the whole data set; this makes the number of floating point operations rather small.)

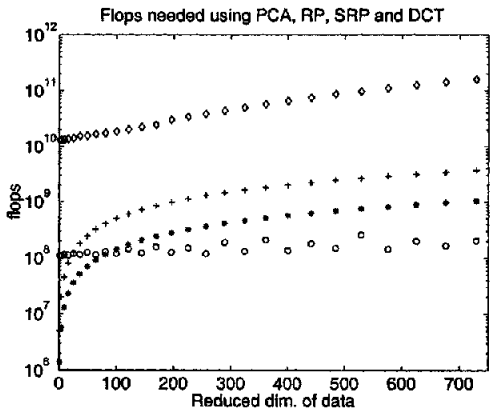

Figure 2: Number of Matlab's floating point operations needed when reducing the dimensionality of image data using RP (+), SRP (∗), PCA (◇) and DCT (○), in a logarithmic scale.

From Figures 1–2 we can conclude that random projection is a computationally inexpensive method of dimensionality reduction while preserving the distances of data vectors practically as well as PCA and clearly better than DCT. Even more, at smallest dimensions RP outperforms both PCA and DCT.

Dimensionality reduction on image data differs slightly from another common procedure, image compression, in which the image is transformed into a more economical form for e.g. transmission, and then transformed back into the original space. The transformation is often chosen so that the resulting image looks as similar as possible to the original image, to a human eye. In this respect, the discrete cosine transform has proven optimal. To see how an image whose dimensionality is reduced by RP would look like, the random mapping should be inverted. The pseudoinverse of R is expensive to compute, but since R is almost orthogonal, the transpose of R is a good approximation of the pseudoinverse, and the image can be computed as $X^{new}_{d\times N} = R^T_{d\times k} X^{RP}_{k\times N}$ where X^{RP} is the result of the random projection (1). Nonetheless, the obtained image is visually worse than a DCT compressed image, to a human eye. Thus random projection is successful in applications where the distance or similarity between data vectors should be preserved under dimensionality reduction as well as possible, but where the data is not intended to be visualized for the human eye. These applications include, e.g., machine vision: it would be possible to automatically detect whether an (on-line) image from a surveillance camera has changed or not.

3.2 Noise reduction in images

In our second set of experiments we considered noisy images. The images were corrupted by salt-and-pepper impulse noise: with probability 0.2, a pixel in the image was turned black or white. We wanted to project the data in such a way that the distance between two data vectors in the reduced noisy data space would be as close as possible to the distance between these vectors in the high-dimensional *noiseless* data space, even though the dimensionality reduction was applied to high-dimensional noisy images.

A simple yet effective way of noise reduction especially in the case of salt-and-pepper impulse noise is median filtering (MF) where each pixel in the image is replaced by the median of the pixel brightnesses in its neighborhood. The median is not affected by individual noise spikes and so median filtering eliminates impulse noise quite well [27]. A common neighborhood size is 3×3 pixels which was also used in our experiments. MF is computationally very efficient, of order $O(dmN)$ for N image windows of d pixels, where m denotes the size of the neighborhood (in our case, $m = 9$). Also, MF does not require dimensionality reduction; thus its result can be used as a yardstick when comparing methods for dimensionality reduction and noise cancellation.

Figure 3 shows how the distance between two noisy image windows is distorted in dimensionality reduction, compared to their distance in the original high-dimensional, noiseless space. Here we can compare different dimensionality reduction methods with respect to their sensitivity to noise. We can see that median filtering introduces quite a large distortion in the image windows, despite that to a human eye it removes impulse noise very efficiently. The distortion is due to blurring: pixels are replaced by the median of their neighborhood, eliminating noise but also small details. PCA, DCT and random projection perform quite similarly to the noiseless case. From Figure 3 we can conclude that random projection is a promising alternative to dimensionality reduction on noisy data, too, as it does not seem to be sensitive to impulse noise. There exists of course many other methods for noise reduction, too. Here our interest was mainly in dimensionality reduction and not noise reduction.

4. RESULTS ON TEXT DATA

Next, we applied dimensionality reduction techniques on text document data from four newsgroups of the 20 newsgroups corpus[3]: sci.crypt, sci.med, sci.space and soc.religion.christian. The documents were converted into term frequency vectors and some common terms were removed using McCallum's Rainbow toolkit[4] but no stemming was used.

The data was not made zero mean, nor was the overall variance of entries of the data matrix normalized. The document vectors were only normalized to unit length. This kind of preprocessing was different from that applied to im-

[3]Available from `http://www.cs.cmu.edu/~textlearning`
[4]Available from `http://www.cs.cmu.edu/~mccallum/bow`

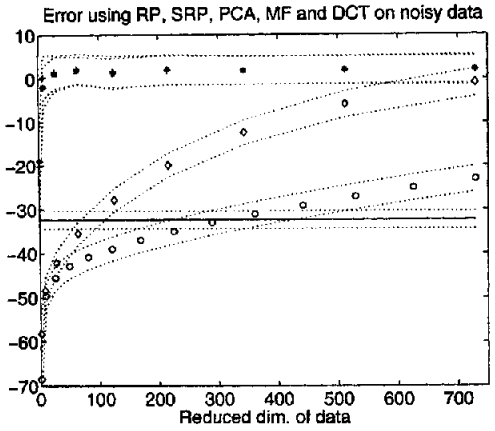

Figure 3: The error produced by RP (+), SRP (∗), PCA (◇), DCT (○) and MF (−) on noisy image data, with 95% confidence intervals over 100 pairs of image windows. In MF dimensionality is not reduced.

age data. Together with the distinct natures of image and text data, differences in preprocessing yielded slightly different results on these different data sets. The size of the vocabulary was $d = 5000$ terms and the data set consisted of $N = 2262$ newsgroup documents.

We randomly chose pairs of data vectors (that is, documents) and computed their similarity as their inner product. The error in the dimensionality reduction was measured as the difference between the inner products before and after the dimensionality reduction.

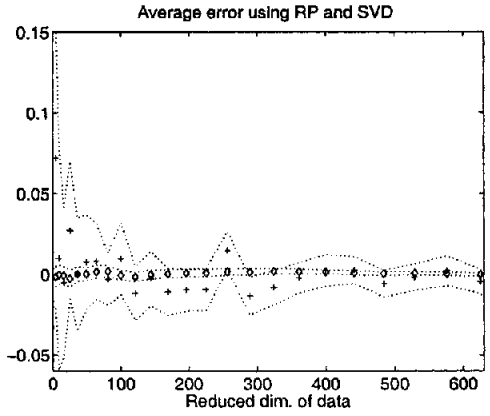

Figure 4: The error produced by RP (+) and SVD (◇) on text document data, with 95% confidence intervals over 100 pairs of document vectors.

Figure 4 shows the error introduced by dimensionality reduction. The results are averaged over 100 document pairs. The results of SVD and random projection with a Gaussian distributed random matrix are shown, together with 95 per cent confidence intervals. The reduced dimensionality k took values in $[1, 700]$. It is seen that random projection is not quite as accurate as SVD but in many applications the error may be neglectable. The Johnson-Lindenstrauss result [15] states that Euclidean distances are retained well in random projection. The case of inner products is a different one — Euclidean distances of document vectors would probably have been preserved better. It is a common practice to measure the similarity of document vectors by their inner products; thus we present results on them.

Despite using efficient SVD routines for finding a few singular vectors of a sparse matrix, SVD is still orders of magnitude more burdensome than RP.

Our results on text document data indicate that random projection can be used in dimensionality reduction of large document collections, with less computational complexity than latent semantic indexing (SVD). Similarly to what was presented in [22], RP can speed up latent semantic indexing (LSI): the dimensionality of the data is first reduced by RP and the burdensome LSI is only computed in the new low-dimensional space. In [22] the documents were generated artificially and the random matrix R was assumed strictly orthogonal; our experiments show that neither of these restrictions is actually necessary. Another common problem in text document retrieval is query matching. Random projection might be useful in query matching if the query is long, or if a set of *similar* documents instead of one particular document were searched for.

5. CONCLUSIONS

We have presented new and promising experimental results on random projection in dimensionality reduction of high-dimensional real-world data sets. When comparing different methods for dimensionality reduction, the criteria are the amount of distortion caused by the method and its computational complexity. Our results indicate that random projection preserves the similarities of the data vectors well even when the data is projected to moderate numbers of dimensions; the projection is yet fast to compute.

Our application areas were of quite different natures: noisy and noiseless images of natural scenes, and text documents from a newsgroup corpus. In both application areas, random projection proved to be a computationally simple method of dimensionality reduction, while still preserving the similarities of data vectors to a high degree.

We also presented experimental results of random projection using a sparsely populated random matrix introduced in [1]. It is in fact not necessary to use a Gaussian distributed random matrix but much simpler matrices still obey the Johnson-Lindenstrauss lemma [15], giving computational savings.

One should emphasize that random projection is beneficial in applications where the distances of the original high-dimensional data points are meaningful as such — if the original distances or similarities are themselves suspect, there is little reason to preserve them. For example, consider using the data in neural network training. Projecting the data onto a lower dimensional subspace speeds up the training only if the training is based on interpoint distances; such problems include clustering and k Nearest Neighbors etc. Also, consider the significance of each of the dimensions of a data set. In a Euclidean space, every dimension is equally important and independent of the others, whereas e.g. in a process monitoring application some measured quantities (that is, dimensions) might be closely related to others, and the interpoint distances do not necessarily bear a clear meaning.

A still more realistic application of random projection would be to use it in a data mining problem, e.g. clustering, and compare the results and computational complexity of mining the original high-dimensional data and dimensionality reduced data; this is a topic of a further study.

An interesting open problem concerns k, the number of dimensions needed for random projections. The Johnson-Lindenstrauss result [15, 10, 8] gives bounds that are much higher than the ones that suffice to give good results on our empirical data. For example, in the case of our image data, the lower bound for k on $\epsilon = 0.2$ is 1600 but in the experiments, $k \approx 50$ was enough. The Johnson-Lindenstrauss result, of course, is a worst-case one, and it would be interesting to understand which properties of our experimental data make it possible to get good results by using fewer dimensions.

We conclude that random projection is a good alternative to traditional, statistically optimal methods of dimensionality reduction that are computationally infeasible for high dimensional data. Random projection does not suffer from the curse of dimensionality, quite contrary to the traditional methods.

6. REFERENCES

[1] D. Achlioptas. Database-friendly random projections. In *Proc. ACM Symp. on the Principles of Database Systems*, pages 274–281, 2001.

[2] C. C. Aggarwal, J. L. Wolf, and P. S. Yu. A new method for similarity indexing of market basket data. In *Proc. 1999 ACM SIGMOD Int. Conf. on Management of data*, pages 407–418, 1999.

[3] R. Agrawal, C. Faloutsos, and A. Swami. Efficient similarity search in sequence databases. In *Proc. 4th Int. Conf. of Data Organization and Algorithms*, pages 69–84. Springer, 1993.

[4] R. I. Arriaga and S. Vempala. An algorithmic theory of learning: robust concepts and random projection. In *Proc. 40th Annual Symp. on Foundations of Computer Science*, pages 616–623. IEEE Computer Society Press, 1999.

[5] M.-W. Berry. Large-scale sparse singular value computations. *International Journal of Super-Computer Applications*, 6(1):13–49, 1992.

[6] S. Dasgupta. Learning mixtures of Gaussians. In *40th Annual IEEE Symp. on Foundations of Computer Science*, pages 634–644, 1999.

[7] S. Dasgupta. Experiments with random projection. In *Proc. Uncertainty in Artificial Intelligence*, 2000.

[8] S. Dasgupta and A. Gupta. An elementary proof of the Johnson-Lindenstrauss lemma. Technical Report TR-99-006, International Computer Science Institute, Berkeley, California, USA, 1999.

[9] S. Deerwester, S.T. Dumais, G.W. Furnas, and T.K. Landauer. Indexing by latent semantic analysis. *Journal of the Am. Soc. for Information Science*, 41(6):391–407, 1990.

[10] P. Frankl and H. Maehara. The Johnson-Lindenstrauss lemma and the sphericity of some graphs. *Journal of Combinatorial Theory, Ser. B*, 44:355–362, 1988.

[11] G.H. Golub and C.F. van Loan. *Matrix Computations*. North Oxford Academic, Oxford, UK, 1983.

[12] A. Graps. An introduction to wavelets. *IEEE Computational Science and Engineering*, 2(2):50–61, 1995.

[13] R. Hecht-Nielsen. Context vectors: general purpose approximate meaning representations self-organized from raw data. In J.M. Zurada, R.J. Marks II, and C.J. Robinson, editors, *Computational Intelligence: Imitating Life*, pages 43–56. IEEE Press, 1994.

[14] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proc. 30th Symp. on Theory of Computing*, pages 604–613. ACM, 1998.

[15] W.B. Johnson and J. Lindenstrauss. Extensions of Lipshitz mapping into Hilbert space. In *Conference in modern analysis and probability*, volume 26 of *Contemporary Mathematics*, pages 189–206. Amer. Math. Soc., 1984.

[16] S. Kaski. Data exploration using self-organizing maps. In *Acta Polytechnica Scandinavica, Mathematics, Computing and Management in Engineering Series*, number 82. 1997. Dr.Tech. thesis, Helsinki University of Technology, Finland.

[17] S. Kaski. Dimensionality reduction by random mapping. In *Proc. Int. Joint Conf. on Neural Networks*, volume 1, pages 413–418, 1998.

[18] E. J. Keogh and M. J. Pazzani. A simple dimensionality reduction technique for fast similarity search in large time series databases. In *4th Pacific-Asia Conf. on Knowledge Discovery and Data Mining*, 2000.

[19] J.M. Kleinberg. Two algorithms for nearest-neighbor search in high dimensions. In *Proc. 29th ACM Symp. on Theory of Computing*, pages 599–608, 1997.

[20] M. Kurimo. Indexing audio documents by using latent semantic analysis and SOM. In E. Oja and S. Kaski, editors, *Kohonen Maps*, pages 363–374. Elsevier, 1999.

[21] R. Ostrovsky and Y. Rabani. Polynomial time approximation schemens for geometric k-clustering. In *Proc. 41st Symp. on Foundations of Computer Science*, pages 349–358. IEEE, 2000.

[22] C.H. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala. Latent semantic indexing: A probabilistic analysis. In *Proc. 17th ACM Symp. on the Principles of Database Systems*, pages 159–168, 1998.

[23] K.R. Rao and P. Yip. *Discrete Cosine Transform: Algorithms, Advantages, Applications*. Academic Press, 1990.

[24] S. Roweis. EM algorithms for PCA and SPCA. In *Neural Information Processing Systems 10*, pages 626–632, 1997.

[25] G. Salton and M.J. McGill. *Introduction to modern information retrieval*. McGraw-Hill, 1983.

[26] L. Sirovich and R. Everson. Management and analysis of large scientific datasets. *Int. Journal of Supercomputer Applications*, 6(1):50–68, spring 1992.

[27] M. Sonka, V. Hlavac, and R. Boyle. *Image processing, analysis, and machine vision*. PWS Publishing, 1998.

[28] S. Vempala. Random projection: a new approach to VLSI layout. In *Proc. 39th Annual Symp. on Foundations of Computer Science*. IEEE Computer Society Press, 1998.