Dimensionality Reduction Using a Randomized Projection Algorithm: Preliminary Results

Travis Atkison, Hillol Kargupta, Charles Nicholas

Computer Science and Electrical Engineering Department
University of Maryland, Baltimore County
1000 Hilltop Circle, Baltimore, MD 21250
{atkison, hillol, nicholas}@cs.umbc.edu

UMBC Technical Report TR-CS-01-11

Abstract

We describe an implementation and experiments with a low-distortion randomized projection algorithm [LINI94] that can reduce the number of dimensions in the data by a considerable amount. The performance of the randomized algorithm is compared with that of a popular technique---Principal Component Analysis (PCA). The experiments show that the randomized projection algorithm consistently outperforms the PCA.

1 Introduction

Mining high-dimensional data often requires construction of a low-dimensional embedding that preserves the underlying "structure" hidden in the data. Randomized projection techniques [LINI94, MANN01] offer one way to do that. These randomized algorithms can guarantee that the distances between points in the original dataset remain almost invariant in the projected dataset. These projections usually reduce the number of dimensions dramatically. Such reduction of dimensionality can be used in a number of different data mining applications. Text data mining is an example. In text mining it is not uncommon to see as many as 100,000 features derived for each document in the corpus. The ability to reduce the number of features for a given data record down to a more manageable number is very useful particularly as we move toward real-time applications.

This paper presents our preliminary experiments with the randomized projection algorithm (in short the LLR algorithm) developed by Linial, London and Rabinovich [LINI94]. We report experiments performed on a diverse set of data and compare its performance with the popular Principal Component Analysis (PCA) [HOTE33].

Section 2 presents the background and the related work. It also describes the data that we use for our experiments. Section 3 describes the methodology and setup of our experiments. Section 4 presents the results. Finally, section 5 concludes this paper and identifies the future work.

2 Background and Related Work

A large collection of related work on this problem can be found in the Linial paper [LINI94]. We present a brief synopsis. Numerous efforts in the field of feature (dimensionality) reduction for finite metric spaces have been documented in [ARIA92], [BALL90] and [BOUR85] as well as others. Hjaltason and Samet [HJAL00] describe work into a variant of Lipschitz embeddings, which is aimed at reducing the high cost of computing the embeddings and eliminating the large number of coordinate values. Another topic that fits with-in this realm of graph metrics is low-dimensional graph coloring [AWER90] and [LINI93].

Several groups are using randomized projection techniques in various ways including Mannila and his group [MANN01], which are using random projection techniques to map sequences of events. Arriaga and Vempala [ARRI99], in the field of concept learning, are using projections to learn concept classes while maintaining a desired level of robustness in half-spaces. Their implementation is based on a neural network, which they call a neuronal, allows for the robustness parameter not to be known in advance. The locality preserving hashing schemes proposed by Indyk et. al. [INDY97] has many applications including high-dimensional search and multimedia indexing. Hristescu and Farach-Colton [HRIS99] use the Smith-Waterman distance function in their projection approach to perform efficient similarity searches of protein databases. Cowen and Priebe [COWE97] are applying projections in a pattern recognition problem where they are, among other methods, clustering PET scan brain volumes.

Building on the previous work of those like Linial *et al.*, we are exploring and have begun to evaluate the effectiveness of a low-dimensional embedding algorithm as a possible aide to the problem of dimensionality in applications such as text data mining. Below is a description of the algorithm and datasets used in our experiments.

2.1 Algorithm

There are many problem domains where a Euclidian distance metric would not make sense, for example text mining. Our approach was to implement a solution that did not have the traditional restraints of relying on the Euclidian distance metric or metric space. The avenue that we chose was to define a method to efficiently construct an embedding that would represent a non-Euclidian space as a Euclidian space as efficiently as possible.

Given a domain (χ^n) that contains a dataset, let (ρ_x) be a metric that defines the distance between any two points in that dataset. An isometry is a mapping γ from one metric space (X^n,ρ_x) to another metric space (Y^m,ρ_y) such that $\rho_x(x_1, x_2) = \rho_y(\gamma(x_1), \gamma(x_2))$. In other words, γ preserves the distance among x_1, x_2. We say the mapping γ to be $\in$–nearly isometric, if and only if $\frac{\rho_x(x_1, x_2)}{\rho_y(\gamma(x_1), \gamma(x_2))} \leq \in$. In this case we may say that the mapping has an $\in$ distortion [KARG00].

The algorithm that we used in our experiments is a variation of Linial's [LINI94], which is an extension of the Johnson-Lindenstrauss [JOHN84] and Bourgain [BOUR85] algorithms.

Theorem 2.1: (**Johnson-Lindenstrauss [JOHN84]**) *Any set of n points in a Euclidean space can be mapped to* $\Re^t$ *where* $t = O(\frac{\log n}{\varepsilon^2})$ *with distortion* $\leq 1 + \varepsilon$ *in the distances. Such a mapping may be found in random polynomial time.*

Theorem 2.2: (**Bourgain [BOUR85]**) *Every n-point metric space (X,d) can be embedded in an O(log n)-dimensional Euclidean space with an O(log n) distortion.*

Lemma 3.3: (**Linial [LINI94]**) *In random polynomial-time (X,d) may be embedded in* $l_p^{O(\log^2 n)}$ *(for any p > 2), with distortion O(log n).*

The general organization of the algorithm is as follows. For each cardinality $k < n$ which is a power of 2, randomly pick O(log n) sets $A \subseteq V(G)$ of cardinality k. Map every vertex x to the vector (d(x,A)) (where $d(x,A) = \min\{d(x,y) | y \in A\}$) with one coordinate for each A selected [LINI94]. We show that this mapping has an O(log n) distortion.

Briefly stated, the algorithm chooses *O(log n)* number of subsets from the data and computes the minimum distance between the points and the subsets to create the projection. We extended the algorithm so that instead of producing (log n) features for a given dataset in the new space, the algorithm would produce any desired number of features for the dataset. This extension was extremely useful during our experiments, as the results will show.

2.2 Data

To demonstrate the effectiveness of the implementation and methodology of the LLR algorithm, we used three different datasets. Two were from the UCI Machine Learning Repository and the third was from a DARPA sponsored project.

2.2.1 Forest Cover Type

The Forest Cover Type dataset, obtained from UCI, is the actual forest cover type for a given observation (30 x 30 meter cell) which was determined from the US Forest Service (USFS) Region 2 Resource Information System (RIS) data. The original data was obtained from the US Geological Survey (USGS) and USFS data with derivation of independent variables added. The data was used in its raw form (not scaled) and contained binary (0 or 1) columns of data for qualitative independent variables

(wilderness areas and soil types). The dataset contained 54 features and a class variable to designate the cover type [BAY99].

2.2.2 The Insurance Company Benchmark

The Insurance Company Benchmark (COIL 2000) dataset has information about customers that consists of 86 variables and includes product usage data and socio-demographic data derived from zip and area codes. The data is based on a real world business problem and was supplied to UCI by the Dutch data mining company Sentient Machine Research. The training set contains over 5000 descriptions of customers, including the information of whether or not they have a caravan insurance policy. A test set contains 4000 customers of whom only the organizers know if they have a caravan insurance policy [BAY99]. For our experimentations we only used the first 1000 records of the training set.

2.2.3 Network Intrusion

The third dataset that was used in our experiments came from the 1998 off-line intrusion detection evaluation (IDEVAL), which was conducted by MIT Lincoln Laboratory under DARPA sponsorship. The contents of network traffic such as SMTP, HTTP, and FTP file transfers were either statistically similar to live traffic, or sampled from public-domain sources. These statistical profiles indicated the frequency of occurrence of different UNIX commands (e.g. mail, lynx, ls, cd, vi, cc, and man), typical login times and telnet session durations, typical source and destination machines, and other information [LIPP00].

The following attack families were included in the evaluation: *user to root, remote to local, denial of service,* and *probe/surveillance*. A *user to root* attack occurs when a local user on a machine tries to obtain privileges normally reserved for the UNIX root or super user. In *remote to local* attacks, an attacker who does not have an account on a victim machine sends packets to that machine in order to gain local access. *Denial of service* attacks are designed to disrupt a host or network service. *Probe/surveillance* attacks occur when an unauthorized user scans a network of computers to gather information or find known vulnerabilities [LIPP00], perhaps in order to then launch one of the other attacks. For a more detailed explanation and definition of these families of network attacks, see Kendall's thesis [KENN99].

The IDEVAL data was gathered by running tcpdump, a network-sniffing tool, on a local network and saving the packets to a file. We received a copy of this file as our dataset. This file was still in raw packet form when we received it; therefore, our first task was to re-assemble it into individual sessions, where each session represented a complete user interaction. These sessions were then preprocessed to produce features. In our case, the features were n-grams. N-grams are n character sequences computed by sliding window size of n across the entire session one character at a time. For our dataset

of 1000 sessions and using 3 as a value on n, 6500 total features were created for the session corpus.

3 Experiments

Our implementation of the low-dimensional, embedding algorithm was written in the C programming language, and all experiments with the algorithm were performed on a Linux machine. Multiple experiments were developed and executed to test the accuracy of our implementation. In addition to the algorithm, multiple auxiliary programs were written in C to evaluate the algorithm's performance. The performance of our low-dimensional, randomized projection embedding was measured against the original dataset as well as against Principal Component Analysis (PCA) [HOTE33]. For our PCA algorithm experiments we used the R Statistical Package (www.r-project.org).

The thread of our experimental procedure was as follows: input a given dataset to our algorithm which produced as output a low-dimensional embedded dataset with a given number of features. Next, compute the pairwise distance of all the points in our new low-dimensional space as well as compute the pairwise distance of the points in our original space. Both of these pairwise distance matrices were then normalized and a cell-by-cell difference was calculated. This difference calculation was averaged over the entire matrix to produce the results graphed below. The same thread was also applied to the PCA algorithm.

For both of the UCI datasets, we created three subsets with varying number of points (10, 100 and 1000) in each. A subset of 1000 data points was obtained from the DARPA dataset. The experimental thread defined above was run against all of these datasets. Described below are the results of these experimental runs.

4 Results

For each dataset, our algorithm was run multiple times with varying number of features produced for each experimental run. The results for each dataset are gathered together in the figures below. Graphed together, for each of the difference dataset sizes, are our low-dimensional embedding algorithm results and the results produced using PCA. Each figure as a whole presents a comparison of our low-dimensional embedding algorithm and PCA for a given dataset. In all graphs, the x-axis is the number of features that were calculated in the new space and the y-axis is the average difference between the particular low-dimensional embedding method and the original dataset.

The explanations of both Figure 1 and Figure 2 are identical, therefore, only Figure 1 will be discussed here. As can be seen in the graphs, for each dataset size, the corresponding graphs for our algorithm and PCA are similar, with PCA having a sharper drop off as compared to our algorithm in some cases. The important item to notice is that in every instance our algorithm had a significantly lower average difference than the

corresponding PCA plot. This means that with the same number of low-dimensional features our algorithm preserved the characteristics of the actual data much better than did the PCA.

The performance of our algorithm on the DARPA dataset is comparable to that of the UCI datasets. This suggests that a huge win can be achieved when using our randomized projections instead of the original dataset in data mining/information retrieval algorithms. A reduction from the original 6500 features down to approximately 30 would allow algorithms that could not handle those numbers of features to run to completion successfully. It should be noted that because of limitations with the particular PCA implementation we were using it was not possible to get a comparison run against the DARPA intrusion detection dataset. We feel that these results are very promising and are planning further experimentation into the usability of this type of algorithm.

5 Conclusion and Future Work

We have shown through our experiments that our algorithm out-performs a standard linear algorithm on three datasets. For all experimental results presented, our algorithm has an overall smaller average difference from the original dataset than the comparison algorithm, PCA. In the DARPA dataset case, the linear algorithm would not even run to completion. These results are promising and show that our low-dimensional, randomized projection, embedding algorithm can be used successfully to reduce a large feature space to a more manageable size with very little distortion from the original dataset.

One of our next steps with this development process is to investigate how well the algorithm scales with respect to the number of features and the number of data points in a given dataset. Throughout this phase, multiple data structures will be developed and implemented to store the algorithm's internal information. As the process continues we will begin to increase the upper limits of data points, metric space and data features that the algorithm can handle.

Another avenue, as our development continues, is to derive a plan to use our algorithm as a pre-processor to a number of feature bound systems, for example network intrusion detection applications and text data mining applications.

6 Acknowledgments

The second author acknowledges support from the United States National Science Foundation CAREER award IIS-0093353.

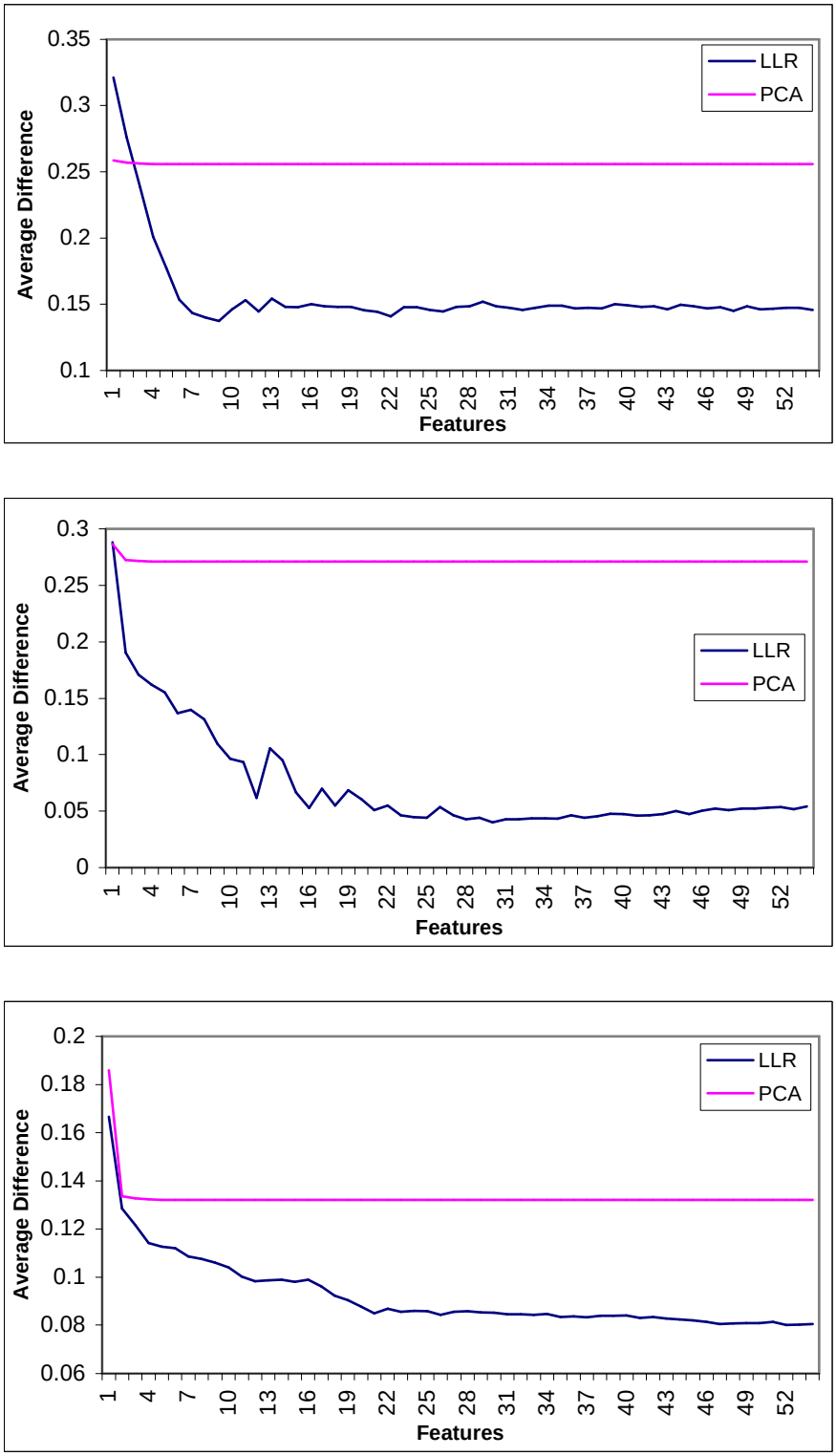

Figure 1. Performance comparison between our algorithm and PCA on the Forest Cover Type Dataset for (top) 10 data points, (center) 100 data points and (bottom) 1000 data points. The graphs show average difference between the original dataset and our algorithm's embedding along with the average difference between the original dataset and PCA's embedding for each of the calculated features.

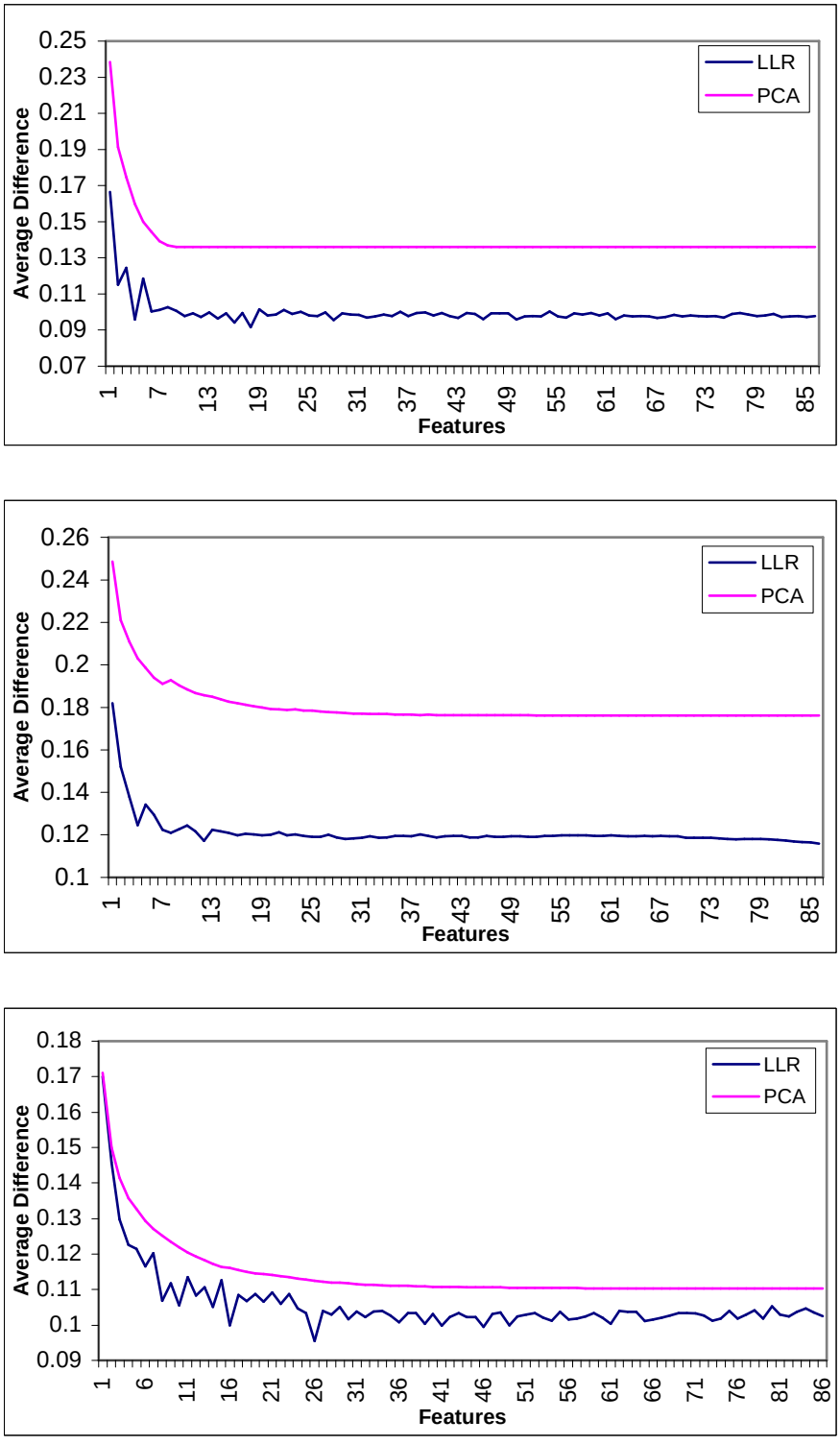

Figure 2. Performance comparison between our algorithm and PCA on The Insurance Company Benchmark (COIL 2000) Dataset for (top) 10 data points, (center) 100 data points and (bottom) 1000 data points. The graphs show average difference between the original dataset and our algorithm's embedding along with the average difference between the original dataset and PCA's embedding for each of the calculated features.

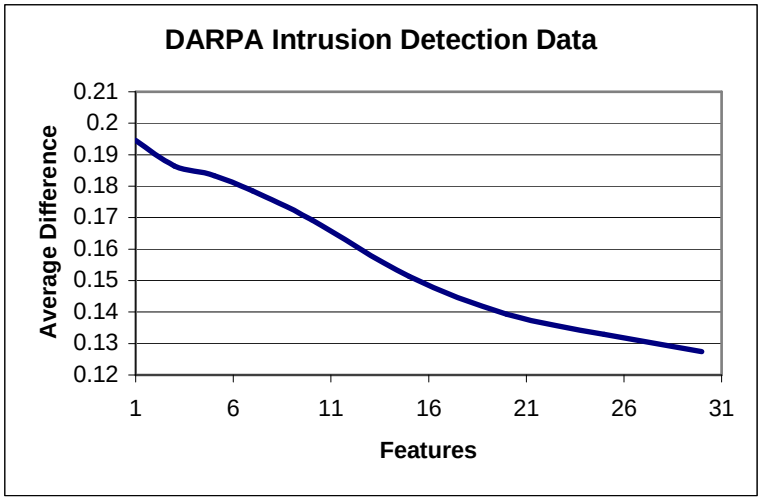

Figure 3. Performance results of our algorithm on the DARPA Intrusion Detection Dataset. The graph shows the average difference between the original dataset and our algorithm's embedding for each of the calculated features.

References

[ARIA92] J. Arias-de-Reyna and L. Rodrigues-Piazza, "Finite metric spaces needing high dimension for Lipschitz embeddings in Banach spaces," Israel J. Math. 79 (1992), 103-111.

[ARRI99] Rosa I. Arriaga and Santosh Vempala, "An algorithmic theory of learning: Robust Concepts and Random Projection," In Proceedings of the 40th Foundations of Computer Science (FOCS '99), New York, 1999.

[AWER90] B. Awerbuch and D. Peleg, "Sparse partitions," In Proceedings of the 40th Foundations of Computer Science (FOCS) 31 (1990), 503-513.

[BALL90] K. Ball, "Isometric embeddings in l_p-spaces," European Journal of Combinatorics 11 (1990), 305-311.

[BAY99] S. D. Bay, (1999). The UCI KDD Archive [http://kdd.ics.uci.edu]. Irvine, CA: University of California, Department of Information and Computer Science.

[BOUR85] J. Bourgain, "On Lipschitz embedding of finite metric spaces in Hillbert space," Israel J. Math. 52 (1985), 46-52.

[COWE97] L.J. Cowen and C.E. Priebe, "Randomized non-linear projections uncover high-dimensional structure," Advances in Applied Math 19:319-331, 1997.

[HJAL00] G. R. Hjaltason and H. Samet, “Contractive Embedding Methods for Similarity Searching in Metric Spaces,” University of Maryland Computer Science TR 4102, February 2000.

[HOTE33] H. Hotelling, “Analysis of a complex of statistical variables into principal components,” Journal of Educational Psychology, 24, 1933.

[HRIS99] Gabriela Hristescu and Martin Farach-Colton. “Cluster-preserving embedding of proteins,” Rutgers University Center for Discrete Mathematics and Computer Science (DIMACS) TR 99-50, 1999.

[INDY97] Piotr Indyk, Rajeev Motwani, Prabhakar Raghavan and Santosh Vempala, “Locality-Preserving Hashing in Multidimensional Spaces,” In Procedings of the 29th ACM Symposium on the Theory of Computing (STOC '97), El Paso, 1997.

[JOHN84] W. B. Johnson and J. Lindenstrauss, “Extensions of Lipschitz mappings into a Hilbert space,” Contemporary Mathematics 26 (1984), 189-206.

[KARG00] H. Kargupta, W. Huang, S. Krishnamoorthy and E. Johnson, “Distributed Clustering Using Collective Principal Component Analysis,” Accepted for publication in Knowledge and Information Systems Journal Special Issue on Distributed and Parallel Knowledge Discovery. (In press).

[KEND99] K.Kendall, “A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems”, S. M. Thesis, MIT Department of Electrical Engineering and Computer Science, June 1999.

[LINI93] N. Linial, D. Peleg, Yu. Rabinovich and M. Saks, “Sphere packing and local majorities in graphs,” The Second Israel Syposium on Theory and Computing Systems (1993), 141-149.

[LINI94] N. Linial, E. London and Y. Rabinovich, “The geometry of graphs and some of its algorithmic applications,” In Proceedings of the 35th Annual Symposium on Foundations of Computer Science, pages 577-591, Los Alamitos, California, USA, 1994. IEEE Computer Society Press.

[LIPP00] Richard P. Lippmann, David J. Fried, Isaac Graf, Joshua W. Haines, Kristopher R. Kendall, David McClung, Dan Weber, Seth E. Webster, Dan Wyschogrod, Robert K. Cunningham, and Marc A. Zissman, “Evaluating Intrusion Detection Systems: The 1998 DARPA Off-Line Intrusion Detection Evaluation,” in Proceedings of the 2000 DARPA Information Survivability Conference and Exposition, 2000, Vol 2.

[MANN01] Heikki Mannila and Jouni K. Seppãnen, “Finding similar situations in sequences of events via random projections,” in Proceedings of the First SIAM International Conference on Data Mining, 2001