A Study of Query-Based Dimensionality Reduction

Augustine S. Nsang
Computer Science Department
University of Cincinnati
Cincinnati, OH 45221-0030, USA
nsangas@mail.uc.edu

Anca Ralescu
Computer Science Department
University of Cincinnati
Cincinnati, OH 45221-0030, USA
Anca.Ralescu@uc.edu

Abstract

This paper considers two approaches to query-based dimensionality reduction. Given a data set, D, and a query, Q, the first approach performs a random projection on the dimensions of D that are not in Q to obtain the data set D_R. A new data set (D_{RQ}) is then formed comprising all the dimensions of D that are in the query Q together with the dimensions of D_R. The resulting data set ($Q(D_{RQ})$ is obtained by applying the query Q to D_{RQ}. A similar approach is taken in the second approach with the difference that the random projection method is replaced by Principal Component Analysis. Comparisons are made between these two approaches with respect to the inter-point distance preservation and computational complexity.

Introduction

Given a collection of n data points (vectors) in high dimensional space, it is often helpful to represent the data in a lower dimensional space without the data suffering great distortion (Achlioptas 2004). This operation is known as *dimensionality reduction*.

There are many known methods of dimensionality reduction, including *Random Projection (RP), Singular Value Decomposition (SVD), Principal Component Analysis (PCA), Kernel Principal Component Analysis (KPCA), Discrete Cosine Transform (DCT), Latent Semantic Analysis (LSA)* and many others(Nsang & Ralescu 2009b).

In random projection, the original d-dimensional data is projected to a k-dimensional ($k \ll d$) subspace through the origin, using a random $d \times k$ matrix R whose columns have unit lengths(Bingham & Mannila 2001). If $X_{n\times d}$ is the original set of n d-dimensional observations, then

$X^{RP}_{n\times k} = X_{n\times d} R_{d\times k}$

is the projection of the data in a lower k-dimensional subspace. The key idea of random projection arises from the Johnson Lindenstrauss lemma (Johnson & Lindenstrauss 1984) which states that 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.

Given n data points as an $n \times p$ matrix X of real numbers, to find the best q-dimensional approximation for the data ($q \ll p$) using the PCA approach, the SVD of X is first obtained. In other words, PCA finds matrices U, D and V such that

$$X = UDV^T$$

where:

- U is an $n \times n$ orthogonal matrix (i.e. $U^TU = I_n$) whose columns are the left singular vectors of X;
- V is a $p \times p$ orthogonal matrix (i.e. $V^TV = I_p$) whose columns are the right singular vectors of X;
- D is an $n \times p$ diagonal matrix with diagonal elements $d_1 \geq d_2 \geq d_3 \geq \ldots \geq d_p \geq 0$ which are the singular values of X. Note that the bottom rows of D are zero rows.
- Define V_q to be the matrix whose columns are unit vectors corresponding to the q largest right singular values of X. V_q is a $p \times q$ matrix.

The transformed data matrix is given by $X^{PCA} = X^TV_q$ (Bingham & Mannila 2001).

Dimensionality reduction has several applications in information retrieval, image data processing, nearest neighbor search, similarity search in a time series data set, clustering and signal processing(Nsang & Ralescu 2009b).

Bingham and Mannila (Bingham & Mannila 2001) suggest the use of random projections for query matching in a situation where a set of documents, instead of one particular one, were searched for. This suggests another application of dimensionality reduction, namely to reduce the complexity of the query process. Suppose, for instance, that we want to query a text document data set with say 5000 dimensions. It would be helpful if we can reduce it to 400 dimensions, say, before applying the query, provided that the dimensions represented in the query are not eliminated by the dimensionality reduction process. The complexity of the query processing is reduced while the speed is significantly increased. Besides, given the distance preserving properties of the random projection and other methods, we retain as much as possible the inter-point distances between data items in the original and reduced data sets. This means that algorithms based on

such distances (e.g. clustering, classification) will perform similarly on the original and reduced data sets.

In the first section of this paper, we discuss the original approach to query based dimensionality reduction (suggested by Bingham and Mannila) and explain why this approach fails. In the second section, we present the first alternative approach using random projections, and determine the values of g_1 and g_2 such that, if u and v are two rows of a data set D, and $f(u)$ and $f(v)$ are the corresponding rows of the data set D_{RQ} derived from D, then:

$g_1||u-v||^2 \leq ||f(u)-f(v)||^2 \leq g_2||u-v||^2$

We also determine, in this section, the speed up in query processing due to this approach. In the third section, we outline the second alternative approach (based on PCA), and in the fourth and last section, we compare the two alternative approaches with respect to inter-point distance preservation and computational complexity.

Original Approach (Bingham & Mannila 2001)

Suppose D is a text document data set, and Q is a query. Following the idea suggested by Bingham and Mannila (Bingham & Mannila 2001), instead of applying the query Q to the data set D to obtain the query result $Q(D)$, we first apply random projection to D to obtain the reduced data set, D_R. Querying D_R with the query Q produces the set of documents $Q(D_R)$. Ideally, for this process to become successful, $Q(D_R)$ should be equal to $R(Q(D))$, where R denotes the operation of Random Projection. Fig 1 captures this relationship.

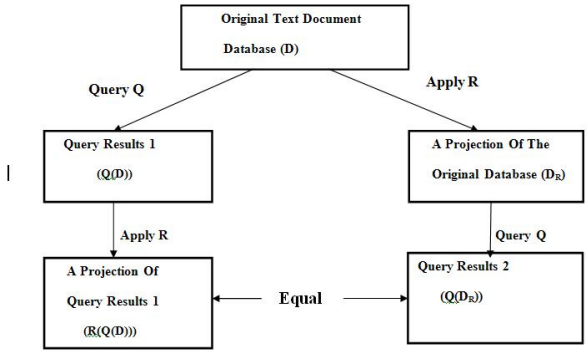

Figure 1: Original Dimensionality Reduction for Query Processing

Unfortunately, this approach fails. This is because when the random projection is applied to the original data set D, all or some of the attributes occurring in a query Q may have been eliminated, and therefore they do not occur in the reduced data set D_R. This can be illustrated by an explicit example (Nsang & Ralescu 2009a). Thus, an alternative approach to reducing the complexity of the query process while not eliminating possibly relevant records from the data set, is needed.

Query-based Dimensionality Reduction Using Random Projections

In this approach, we first perform a random projection on the dimensions of D that are NOT in Q to obtain the data set D_R. A new data set (D_{RQ}) is then formed comprising all the dimensions of D that are in the query Q together with the dimensions got by performing a random projection on the dimensions NOT in Q. The resulting data set ($Q(D_{RQ})$ is obtained by applying the query Q to D_{RQ} (see Fig 2). Thus $Q(D_{RQ})$ is the dimensionality reduced form of $Q(D)$, which is the result of applying the query Q to the text document data set D.

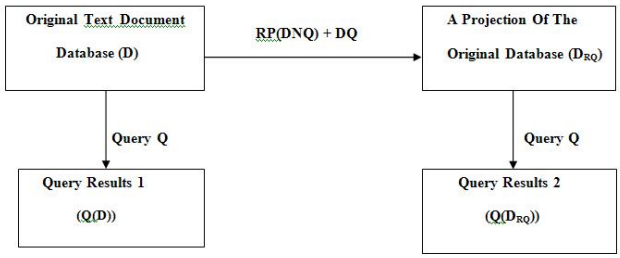

Figure 2: First Alternative Approach

More formally, we have the following. Given A_D, the set of attributes of the data set D, and A_Q, the set of attributes corresponding to query Q, $A_D \backslash A_Q$, the set of attributes of the data set D which are NOT in query Q, then

$$A_{D'} = A_Q \cup RP(A_D \backslash A_Q)$$

where D' corresponds to the data set D_{RQ} in Fig 2. For example, consider the text document data set, D, in Table 1 and the query

$Q =$ *List all documents which have more than two occurrences of each of the terms* **Augustine***(A4) and* **Ill***(A7)*

In this case

$$A_D = \{\text{My, Name, Is, Augustine, He, Was, Ill}\}$$

and

$$A_Q = \{\text{ Augustine, Ill}\}.$$

Thus

$$A_D \backslash A_Q = \{\text{My, Name, Is, He, Was}\}.$$

Now if

$$RP(A_D \backslash A_Q) = \{A_1, A_2, A_3\},$$

then

$$A_{D'} = A_Q \cup RP(A_D \backslash A_Q) = \{Augustine, Ill, A_1, A_2, A_3\}.$$

A natural question which arises is why we need to keep all the attributes NOT in the query (in some reduced form) instead of just discarding them. The answer is that in this way, given the distance preserving properties of the random projection, we retain as much as possible the inter-point distances between data items in the original and reduced data

Table 1: Original data set

Id	My (A1)	Name (A2)	Is (A3)	Augustine (A4)	He (A5)	Was (A6)	Ill (A7)
1	10	0	100	5	1	3	5
2	25	1	150	9	7	9	11
3	35	0	200	15	13	15	17
4	0	25	0	10	19	21	23
5	10	0	95	70	25	40	85
6	10	16	25	14	13	15	17

Table 2: Euclidean distances between the records in original data set

	1	2	3	4	5	6
1	0	53.4	105.6	108.3	**112.2**	80
2	53.4	0	52.4	155.4	117.2	127.3
3	105.6	52.4	0	204.9	141.7	**177.5**
4	108.3	155.4	204.9	0	132.6	30.5
5	**112.2**	117.2	141.7	132.6	0	117
6	80	127.3	**177.5**	30.5	117	0

sets. This means that algorithms based on such distances (e.g. clustering, classification) will perform similarly on the original and reduced data sets.

Consider again the data set represented by Table 1. Suppose that the query is given by: *List all documents which have more than five occurrences of each of the terms* **My, Is** *and at least one occurrence of term* **Name**. In this case, discarding all the attributes not in the query would make the **first** and the **fifth** records much more similar in the reduced set than they were in the original set and the Euclidean distance between the first and fifth records would be much less in the reduced set than in the original set (see Tables 2 and 3).

Table 3: Euclidean distances: data set reduced only to the query attributes

	1	2	3	4	5	6
1	0	52.2	103.1	103.6	**5**	76.7
2	52.2	0	51	154	57	126.8
3	103.1	51	0	204.6	108	**177.5**
4	103.6	154	204.6	0	98.7	28.4
5	**5**	57.0	107.9	98.7	0	71.8
6	76.7	126.8	**177.5**	28.4	71.8	0

At the same time, it would make the **third** and the **sixth** records much more dissimilar in the reduced set than they were in the original set even though the Euclidean distance between the **third** and **sixth** records would be about the same in the reduced set as in the original set.

On the other hand, reducing the original data set by reducing the non-query attributes using RP (and appending the query attributes to the result) significantly preserves the Euclidean distances between the **first** and **fifth** records, and between the **third** and **sixth** records (see Tables 2 and 4). Table 4 was generated from the matrix obtained by multiplying the matrix representing the non-query attributes of the original data set by a 4×3 random matrix R defined by:

$$r_{ij} = \begin{cases} +1 & \text{with probability } \frac{7}{24}; \\ 0 & \text{with probability } \frac{5}{12}; \\ -1 & \text{with probability } \frac{7}{24}. \end{cases} \quad (1)$$

Table 4: Euclidean Distances in the Data Set with Non-query Attributes Reduced by RP

	1	2	3	4	5	6
1	0	52.4	103.6	104.5	**125**	77.3
2	52.4	0	51.4	154.3	134.2	126.9
3	103.6	51.4	0	205	158.2	**177.5**
4	104.5	154.3	204.9	0	158.3	30.4
5	**125**	134.2	158.2	158.3	0	137.2
6	77.3	126.9	**177.5**	30.4	137.2	0

We next determine g_1 and g_2 such that for all $u, v \in D$, $g_1(||u-v||^2) \leq ||f(u)-f(v)||^2 \leq g_2(||u-v||^2)$ where $f(u)$ and $f(v)$ are the corresponding values to u and v in D_{RQ}, where D_{RQ} is obtained from D using the first alternative approach. Recall that for the regular random projection method, $g_1(x) = (1-\varepsilon)x$ and $g_2(x) = (1+\varepsilon)x$ for some value of ε.

Experiment

An experiment was carried out (in MATLAB) on the data set given by the matrix

$$D = \begin{bmatrix} 5 & 6 & 7 & 9 & \mathbf{0} & 9 & 8 & 7 & 6 & 11 & 6 & 74 \\ 3 & 2 & 10 & 6 & \mathbf{3} & 5 & 9 & 4 & 10 & 5 & 0 & 57 \\ 10 & 0 & 10 & 3 & \mathbf{4} & 6 & 2 & 8 & 12 & 0 & 9 & 64 \\ 6 & 3 & 10 & 3 & \mathbf{4} & 0 & 2 & 7 & 0 & 1 & 5 & 0 \\ 6 & 0 & 8 & 6 & \mathbf{1} & 5 & 5 & 7 & 11 & 0 & 2 & 51 \\ 1 & 3 & 4 & 8 & \mathbf{8} & 8 & 5 & 6 & 7 & 9 & 0 & 9 \\ 2 & 2 & 9 & 5 & \mathbf{0} & 5 & 10 & 6 & 3 & 5 & 9 & 4 \\ 2 & 0 & 5 & 2 & \mathbf{7} & 7 & 4 & 6 & 2 & 8 & 12 & 0 \\ 6 & 7 & 4 & 7 & \mathbf{4} & 4 & 0 & 10 & 8 & 4 & 9 & 5 \\ 5 & 2 & 10 & 3 & \mathbf{1} & 8 & 10 & 6 & 8 & 8 & 0 & 9 \end{bmatrix}$$

The columns in this data set represent values of 12 attributes, $A_1 - A_{12}$. The query Q for this experiment is: *Find all data points having an even number of occurrences of the attribute value for* A_5. $Q(D)$ is computed and the result obtained if we were to apply Q to D without first performing random projection is obtained.

To compute D_{RQ}, we generate D_{NQ}, the data set consisting only of the dimensions of D NOT in the query, and D_Q, the data set consisting only of the dimensions in the query. $Q(D), D_{NQ}$ and D_Q are given by:

$$Q(D) = \begin{bmatrix} 5 & 6 & 7 & 9 & \mathbf{0} & 9 & 8 & 7 & 6 & 11 & 6 & 74 \\ 10 & 0 & 10 & 3 & \mathbf{4} & 6 & 2 & 8 & 12 & 0 & 9 & 64 \\ 6 & 3 & 10 & 3 & \mathbf{4} & 0 & 2 & 7 & 0 & 1 & 5 & 0 \\ 1 & 3 & 4 & 8 & \mathbf{8} & 8 & 5 & 6 & 7 & 9 & 0 & 9 \\ 2 & 2 & 9 & 5 & \mathbf{0} & 5 & 10 & 6 & 3 & 5 & 9 & 4 \\ 6 & 7 & 4 & 7 & \mathbf{4} & 4 & 0 & 10 & 8 & 4 & 9 & 5 \end{bmatrix}$$

$$D_{NQ} = \begin{bmatrix} 5 & 6 & 7 & 9 & 9 & 8 & 7 & 6 & 11 & 6 & 74 \\ 3 & 2 & 10 & 6 & 5 & 9 & 4 & 10 & 5 & 0 & 57 \\ 10 & 0 & 10 & 3 & 6 & 2 & 8 & 12 & 0 & 9 & 64 \\ 6 & 3 & 10 & 3 & 0 & 2 & 7 & 0 & 1 & 5 & 0 \\ 6 & 0 & 8 & 6 & 5 & 5 & 7 & 11 & 0 & 2 & 51 \\ 1 & 3 & 4 & 8 & 8 & 5 & 6 & 7 & 9 & 0 & 9 \\ 2 & 2 & 9 & 5 & 5 & 10 & 6 & 3 & 5 & 9 & 4 \\ 2 & 0 & 5 & 2 & 7 & 4 & 6 & 2 & 8 & 12 & 0 \\ 6 & 7 & 4 & 7 & 4 & 0 & 10 & 8 & 4 & 9 & 5 \\ 5 & 2 & 10 & 3 & 8 & 10 & 6 & 8 & 8 & 0 & 9 \end{bmatrix}$$

$$D_Q = \begin{bmatrix} 0 \\ 3 \\ 4 \\ 4 \\ 1 \\ 8 \\ 0 \\ 7 \\ 4 \\ 1 \end{bmatrix}$$

Next we generate the random projection matrix, R, multiply it by D_{NQ} and append D_Q to the result to obtain D_{RQ}. Define the random projection matrix, $R = (r_{ij})$, as:

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

If we wanted D_{RQ} to have a dimensionality of 7, say, R will have to be a 11×6 matrix.

Thus $D_R = D_{NQ} * R$ and $D_{RQ} = DQ \cup D_R$ where $\cup$ denotes the operation of adding to D_R the columns of DQ. The result, $Q(D_{RQ})$, of applying the query Q to D_{RQ} is now the collection of data records in D_{RQ} that satisfy the query Q.

Figures 3, 4 and 5 show the results obtained from a run of a MATLAB implementation of this procedure to obtain R, D_{RQ} and $Q(D_{RQ})$.

We now investigate the relation between the pairwise distances in the original and reduced data sets. For any two records $u, v \in D$, let $f(u)$ and $f(v)$ denote the corresponding records in D_{RQ}. The pairwise distances obtained from our sample run are shown in Table 5 below.

Because the projection matrix R is random, each run generates a different value of $||f(u) - f(v)||^2$ for any pair of rows $u, v \in D$ (and of course the same value of $||u - v||^2$). As we know, the actual value of $||f(u) - f(v)||^2$ corresponding to a specific value of $||u - v||^2$ lies somewhere between $max(||f(u) - f(v)||^2)$ and $min(||f(u) - f(v)||^2)$. Thus, we shall use the midpoint between these two extremes as an estimate of the value of $||f(u) - f(v)||^2$ which corresponds to the value of $||u - v||^2$.

Sixteen runs of the program were made, and for each value of $||u - v||^2$, the maximum and minimum values of

R =
$$\begin{bmatrix} 1.7321 & 0 & 0 & -1.7321 & 0 & 0 \\ 0 & -1.7321 & 0 & 0 & 0 & 0 \\ 0 & 1.7321 & 0 & 0 & -1.7321 & 0 \\ -1.7321 & 0 & -1.7321 & 1.7321 & 0 & -1.7321 \\ 1.7321 & 0 & 0 & 0 & 0 & -1.7321 \\ 0 & 0 & 0 & 0 & 1.7321 & 0 \\ -1.7321 & 1.7321 & 1.7321 & 0 & 0 & -1.7321 \\ -1.7321 & 0 & 0 & -1.7321 & 0 & 1.7321 \\ -1.7321 & 0 & 1.7321 & -1.7321 & -1.7321 & 0 \\ -1.7321 & 0 & 0 & 0 & 0 & 1.7321 \\ 1.7321 & 0 & -1.7321 & 1.7321 & 0 & -1.7321 \end{bmatrix}$$

Figure 3: The R Matrix

D_{RQ} =
$$\begin{bmatrix} 84.8705 & 13.8564 & -112.5833 & 105.6551 & -17.3205 & -150.6884 & 0 \\ 69.2820 & 20.7846 & -93.5307 & 77.9423 & -10.3923 & -107.3872 & 3.0000 \\ 83.1384 & 31.1769 & -102.1910 & 77.9423 & -13.8564 & -103.9230 & 4.0000 \\ -17.3205 & 24.2487 & 8.6603 & -6.9282 & -15.5885 & -8.6603 & 4.0000 \\ 62.3538 & 25.9808 & -86.6025 & 69.2820 & -5.1962 & -96.9948 & 1.0000 \\ -20.7846 & 12.1244 & -3.4641 & 0 & -13.8564 & -41.5692 & 8.0000 \\ -29.4449 & 22.5167 & 3.4641 & -1.7321 & -6.9282 & -13.8564 & 0 \\ -36.3731 & 19.0526 & 20.7846 & -17.3205 & -15.5885 & -1.7321 & 7.0000 \\ -39.8372 & 12.1244 & 3.4641 & -10.3923 & -13.8564 & -15.5885 & 4.0000 \\ -5.1962 & 24.2487 & 3.4641 & -15.5885 & -13.8564 & -31.1769 & 1.0000 \end{bmatrix}$$

Figure 4: The D_{RQ} Matrix

$||f(u) - f(v)||^2$ were obtained. These were further reduced to midpoints between these two extremes (as explained above). More precisely,

$$M_d = max\{||f(u) - f(v)||_i^2 / ||u - v||^2 = d, i = 1..16\}$$

$$m_d = min\{||f(u) - f(v)||_i^2 / ||u - v||^2 = d, i = 1..16\}$$

$$mid_d = \frac{M_d + m_d}{2}$$

where $||f(u) - f(v)||_i^2$ is the distance between $f(u)$ and $f(v)$ in the i^{th} run. The results obtained are summarized in Table 6, and Figure 6 which shows the values of M_d, m_d and mid_d for each value of $d = ||u - v||^2, u, v \in D$ (in this table) and their linear regression lines.

Consider the value X on the $||u - v||^2$ axis (labeled on Figure 6). The corresponding values of $||f(u) - f(v)||_{min}^2$, $||f(u) - f(v)||_{estimate}^2$ and $||f(u) - f(v)||_{max}^2$ are $Y1, Y$ and $Y2$ respectively. Clearly

$$Y1 \leq Y \leq Y2.$$

$$Q(D_{RQ}) = \begin{bmatrix} 84.8705 & 13.8564 & -112.5833 & 105.6551 & -17.3205 & -150.6884 & 0 \\ 83.1384 & 31.1769 & -102.1910 & 77.9423 & -13.8564 & -103.9230 & 4.0000 \\ -17.3205 & 24.2487 & 8.6603 & -6.9282 & -15.5885 & -8.6603 & 4.0000 \\ -20.7846 & 12.1244 & -3.4641 & 0 & -13.8564 & -41.5692 & 8.0000 \\ -29.4449 & 22.5167 & 3.4641 & -1.7321 & -6.9282 & -13.8564 & 0 \\ -39.8372 & 12.1244 & 3.4641 & -10.3923 & -13.8564 & -15.5885 & 4.0000 \end{bmatrix}$$

Figure 5: The $Q(D_{RQ})$ Matrix

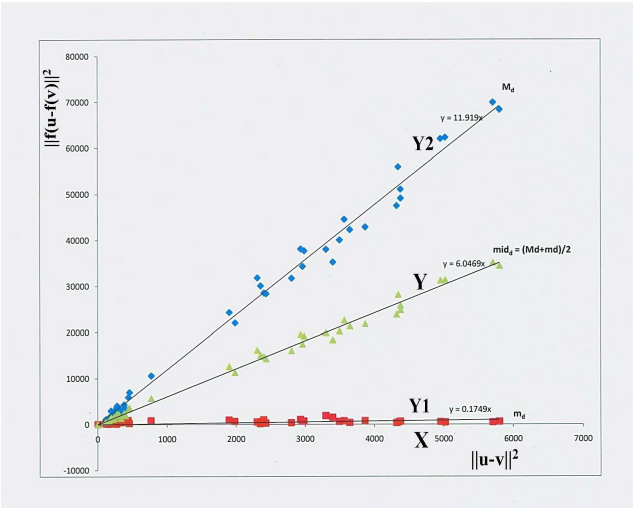

Figure 6: XY chart showing maximum, minimum and estimated values of $||f(u) - f(v)||^2$ for each value of $||u - v||^2$ (RP Approach)

But $Y1 = m_1X$ and $Y2 = m_2X$ where m_1 and m_2 are the slopes of the regression lines corresponding to $||f(u) - f(v)||^2_{min}$ and $||f(u) - f(v)||^2_{max}$ respectively. Thus,

$$m_1X \leq Y \leq m_2X$$

Generalizing, for any value of $||u - v||^2$, we obtain

$$m_1||u - v||^2 \leq ||f(u) - f(v)||^2 \leq m_2||u - v||^2$$

or, letting $g_1(x) = m_1x$ and $g_2(x) = m_2x$, we obtain

$$g_1(||u - v||^2) \leq ||f(u) - f(v)||^2 \leq g_2(||u - v||^2) \quad (2)$$

Since both $g1$ and $g2$ are nondecreasing functions we can prove the following result.

Proposition 1 *Suppose u and v are data points in D, and $f(u)$ and $f(v)$ are their mappings in D_{RQ} obtained from D using the query-based dimensionality reduction procedure. If v is in the neighborhood of u of radius r, then $f(v)$ is in the neighborhood of $f(u)$ of radius $g_2(r)$. Conversely, if $f(v)$ belongs to the neighborhood of radius $g_1(r)$ of $f(u)$, then v belongs to the neighborhood of radius r of u.*

u	v	$\|\|u - v\|\|^2$	$\|\|f(u) - f(v)\|\|^2$
1	2	450	3354
1	3	434	3394
1	4	5801	58117
1	5	764	5683
1	6	4376	46219
1	7	5020	56973
1	8	5705	69877
1	9	4952	60769
1	10	4342	50683
2	3	288	400
2	4	3493	34933
2	5	112	337
2	6	2428	26743
2	7	2956	34275
2	8	3561	44515
2	9	2980	37633
2	10	2348	29539
3	4	4319	38709
3	5	268	909
3	6	3396	30889
3	7	3864	38437
3	8	4377	49083
3	9	3642	42255
3	10	3296	33063
4	5	2797	29154
4	6	395	1456
4	7	185	322
4	8	216	702
4	9	211	744
..	..	..	..
..	..	..	..
8	9	227	612
8	10	373	2208
9	10	332	1626

Table 5: The values of $||u - v||^2$ and $||f(u) - f(v)||^2$ for all u, v in D (RP Approach)

Proof: The proof follows trivially. Define the neighborhood of u of radius r as:

$$\eta(u, r) = \{v \in D| \ ||u - v||^2 \leq r\} \quad (3)$$

From equation (3) it follows that if $v \in \eta(u, r)$, then $g_2(||u - v||^2) \leq g_2(r)$, and therefore $||f(u) - f(v)||^2 \leq g_2(r)$, that is, $f(v) \in \eta(f(u), g2(r))$. Conversely, if $f(v) \in \eta(f(u), g_1(r))$ then $g_1(||u - v||^2) \leq g_1(r)$ and therefore $||u - v||^2 \leq r$, that is $v \in \eta(u, r)$.

Determining the Speed-up of the Query-based Dimensionality Reduction

We investigate now the computational aspects of the query-based dimensionality reduction.

Complexity of the Original Approach Suppose that the query Q with q attributes is applied to the data set $D_{n\times p}$ resulting in a query result $Q(D)$ with m rows and p attributes.

Table 6: Maximum, minimum and estimated values of $||f(u) - f(v)||^2$ for each value of $||u - v||^2$ (RP Approach)

$d = \|\|u - v\|\|^2$	M_d	m_d	$mid_d = \frac{M_d + m_d}{2}$
112	1141	151	646
153	1420	211	815.5
154	1303	121	712
164	1543	142	842.5
185	2974	301	1637.5
211	2244	201	1222.5
216	2073	321	1197
227	2406	615	1510.5
230	2917	430	1673.5
238	2377	361	1369
268	3786	123	1954.5
272	4069	694	2381.5
288	3682	406	2044
301	3298	1177	2237.5
332	2322	441	1381.5
359	3333	546	1939.5
373	4305	699	2502
395	3361	589	1975
434	5908	895	3401.5
450	6996	222	3609
764	10609	838	5723.5
1894	24288	939	12613.5
1978	22012	625	11318.5
2300	31806	528	16167
2348	30043	142	15092.5
2396	28510	1003	14756.5
2428	28336	247	14291.5
2797	31710	339	16024.5
..	..	..	..
..	..	..	..
5020	62256	441	31348.5
5705	69877	448	35162.5
5801	68296	583	34439.5

To compute the query result $Q(D)$ from the data set, D, we must compare the value of each attribute in the query with the value of the corresponding attribute in D for **each row** of D, a total of nq operations. After this, to get the query result $Q(D)$ from D we have to generate an $m \times p$ matrix, which is of complexity $O(mp)$. Thus the original query process has complexity $O(nq + mp)$.

Complexity of the Query-based Reduction We recall that D_{RQ} is computed from D by performing a random projection of the dimensions of D that are NOT in Q, and then simply copying all the columns of D corresponding to attributes in Q into the result. Again, if there are q attributes in Q, then there are $(p - q)$ attributes NOT in Q. Also, if D_{RQ} has k attributes, then D_R has k - no of dimensions in Q = $k - q$ attributes. Thus the random projection reduces an $n \times (p - q)$ matrix into an $n \times (k - q)$ matrix.

Thus, according to the result in (Fradkin & Madigan 2003), the complexity of the random projection step is given by $O((p - q)(k - q)) + O(n(p - q)(k - q))$.

Generating the rest of the matrix D_{RQ} takes $O(nq)$, according to the result in (Fradkin & Madigan 2003) again (since we are generating data having n rows and Q columns). Thus the complexity of the process of generating D_{RQ} from D is $O((n + 1)(p - q)(k - q)) + O(nq)$.

Now, after having already generated D_{RQ}, using the result in the last section, to compute the query result $Q(D_{RQ})$ from the data set D_{RQ} takes $O(nq + mk)$. Thus, the speed up is approximately

$$\frac{C_1}{C_2} = \frac{nq + mp}{nq + mk}$$

where C_1 is the complexity of the original query process and C_2 is the complexity of the process of generating $Q(D_{RQ})$ from D_{RQ}.

Query-based Dimensionality Reduction Using PCA

Another possible approach to query-based dimensionality reduction would be to use PCA in the last section instead of RP. In this case, therefore, we first apply the PCA method on the dimensions of D that are NOT in the query Q to obtain D_P. A new data set (D_{PQ}) is then formed comprising all the dimensions of D_P together with the dimensions of D that are in Q. Applying the query Q to D_{PQ} yields the result $Q(D_{PQ})$.

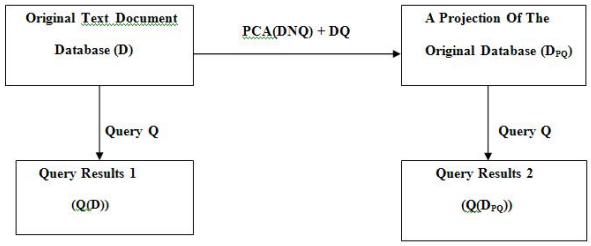

Figure 7: Query-based Dimensionality Reduction Using PCA

Implementation

To compute the projected data set, D_{PQ}, we need to first generate DNQ, the data set consisting only of the dimensions of D **not** in the query Q. We also need to compute DQ, the data set consisting only of the dimensions of D in the query Q. When applied to the data set D and query Q in the last section, the PCA reduction approach results in $DNQ = USV^T$ where U, S and V are shown in Figs 8, 9 and 10.

To compute D_{PQ} (with $k = 7$ columns) we multiply DNQ by the first $k-q$ columns of V (where q is the number of attributes in the query, 1 in this case), and append DQ to the result. $Q(D_{PQ})$ is obtained by applying the query Q to D_{PQ}. The matrices D_{PQ} and $Q(D_{PQ})$ obtained are shown in Figs 11 and 12 respectively.

Comparison of the RP and PCA Query-based Dimensionality Reductions

We now compare the performances of the first and second alternative approaches. To start with, we generate the values

$$U = \begin{bmatrix}
-0.580 & -0.084 & -0.261 & 0.591 & -0.041 & 0.327 & -0.214 & -0.065 & 0.282 & -0.021 \\
-0.451 & -0.097 & -0.288 & -0.249 & -0.105 & 0.109 & 0.258 & -0.200 & -0.703 & -0.142 \\
-0.506 & -0.109 & 0.543 & -0.121 & -0.060 & -0.354 & -0.216 & -0.043 & -0.052 & 0.494 \\
-0.031 & 0.320 & 0.329 & -0.224 & -0.192 & 0.677 & -0.296 & 0.364 & -0.152 & -0.029 \\
-0.406 & -0.079 & 0.118 & -0.339 & 0.142 & -0.116 & 0.312 & 0.394 & 0.407 & -0.498 \\
-0.107 & 0.332 & -0.405 & 0.009 & 0.468 & -0.126 & 0.011 & 0.537 & -0.137 & 0.415 \\
-0.075 & 0.455 & -0.033 & -0.005 & -0.448 & 0.077 & 0.611 & -0.122 & 0.269 & 0.344 \\
-0.038 & 0.443 & 0.115 & 0.411 & -0.325 & -0.468 & -0.115 & 0.214 & -0.275 & -0.402 \\
-0.082 & 0.423 & 0.367 & 0.155 & 0.631 & 0.132 & 0.174 & -0.426 & -0.069 & -0.154 \\
-0.117 & 0.412 & -0.346 & -0.465 & -0.053 & -0.164 & -0.489 & -0.369 & 0.254 & -0.104
\end{bmatrix}$$

Figure 8: The U Matrix

$$S = \begin{bmatrix}
132.97 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 35.98 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 17.24 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 13.43 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 11.41 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 8.58 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 5.50 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 3.38 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.35 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.72 & 0
\end{bmatrix}$$

Figure 9: The S Matrix

$$V = \begin{bmatrix}
-0.10 & 0.18 & 0.36 & -0.22 & 0.09 & 0.10 & -0.50 & -0.14 & 0.43 & 0.13 & -0.54 \\
-0.04 & 0.17 & -0.03 & 0.19 & 0.33 & 0.53 & -0.03 & -0.61 & -0.01 & 0.21 & 0.35 \\
-0.15 & 0.37 & 0.09 & -0.48 & -0.40 & 0.28 & -0.14 & 0.19 & -0.46 & 0.26 & 0.17 \\
-0.11 & 0.25 & -0.14 & 0.10 & 0.37 & 0.32 & 0.48 & 0.46 & 0.05 & 0.29 & -0.36 \\
-0.12 & 0.30 & -0.22 & 0.11 & 0.07 & -0.45 & -0.23 & 0.19 & 0.35 & 0.49 & 0.42 \\
-0.11 & 0.30 & -0.45 & -0.22 & -0.44 & 0.03 & 0.36 & -0.32 & 0.43 & -0.19 & -0.11 \\
-0.12 & 0.39 & 0.24 & -0.04 & 0.23 & 0.12 & -0.04 & 0.30 & 0.22 & -0.67 & 0.36 \\
-0.16 & 0.21 & 0.05 & -0.43 & 0.48 & -0.50 & 0.24 & -0.33 & -0.30 & -0.05 & -0.09 \\
-0.09 & 0.35 & -0.47 & 0.39 & 0.03 & -0.05 & -0.43 & -0.02 & -0.38 & -0.25 & -0.31 \\
-0.08 & 0.37 & 0.56 & 0.52 & -0.33 & -0.21 & 0.29 & -0.16 & -0.10 & 0.08 & -0.08 \\
-0.94 & -0.34 & -0.00 & 0.09 & -0.05 & 0.03 & -0.01 & 0.01 & -0.003 & -0.03 & 0.02
\end{bmatrix}$$

Figure 10: The V Matrix

$$D_{PQ} = \begin{bmatrix}
-77.1476 & -3.0072 & -4.5016 & 7.9347 & -0.4665 & 2.8078 & 0 \\
-59.9135 & -3.4798 & -4.9566 & -3.3439 & -1.2025 & 0.9329 & 3.0000 \\
-67.2098 & -3.9166 & 9.3613 & -1.6209 & -0.6805 & -3.0337 & 4.0000 \\
-4.0986 & 11.5020 & 5.6657 & -3.0036 & -2.1950 & 5.8030 & 4.0000 \\
-54.0249 & -2.8511 & 2.0324 & -4.5479 & 1.6193 & -0.9928 & 1.0000 \\
-14.1784 & 11.9594 & -6.9816 & 0.1167 & 5.3415 & -1.0821 & 8.0000 \\
-9.9353 & 16.3832 & -0.5621 & -0.0651 & -5.1173 & 0.6564 & 0 \\
-5.1125 & 15.9464 & 1.9743 & 5.5184 & -3.7046 & -4.0096 & 7.0000 \\
-10.9375 & 15.2266 & 6.3227 & 2.0865 & 7.2036 & 1.1301 & 4.0000 \\
-15.5171 & 14.8357 & -5.9684 & -6.2444 & -0.6095 & -1.4054 & 1.0000
\end{bmatrix}$$

Figure 11: The D_{PQ} Matrix

of $u, v, ||u-v||^2$ and $||f(u)-f(v)||^2$ for each pair of tuples $u, v \in D_Q$, and corresponding pair of tuples $f(u), f(v) \in D_{PQ}$. For the same example data set, the results obtained using the PCA approach are shown in Table 7.

A graph of $||u-v||^2$ (on the x-axis) against $||f(u)-f(v)||^2$ (on the y-axis) is a straight line through the origin which makes an angle of 45^o with the horizontal, as shown in Figure 13 below. Thus, it is clear that the PCA approach preserves the inter-point distances much much better than our first alternative approach (with RP).

Given a p-dimensional data set with n rows, and assuming we want to have q rows in the reduced data set, the computational complexity of PCA is $(O(p^2n) + O(p^3)$ (Fradkin & Madigan 2003; Bingham & Mannila 2001), while that of RP is $O(pq)+O(npq)$ (as mentioned above). Thus the PCA approach is much more expensive computationally than the RP method.

Conclusion

In this paper, we have examined different approaches to query-based dimensionality reduction. As we observed, the original approach (suggested by Bingham and Mannila (Bingham & Mannila 2001)) which reduces the dimensionality of the entire text document data set by random projection before applying the query will not work when the original and dimensionality reduced data sets have no common attributes, making it impossible in general to query the dimensionality reduced data set using the query that was meant for the original data set.

We then looked at an approach which overcomes this problem by performing random projection only on dimensions not found in the query, Q, and simply adding all the dimensions found in the query to the result. We saw that this approach, like the regular random projection method, preserves inter-point distances to a reasonable extent.

Next, we looked at an approach which simply replaces RP in the approach just described with PCA. We realized this new approach preserves inter-point distances much much better (in fact, perfectly) than the RP approach.

However, the PCA approach is also much more expensive computationally than the RP method.

It would be worth applying the two query-based dimensionality reduction approaches discussed in this paper to image data, and comparing their performances with that of *Discrete Cosine Transform* (Bingham & Mannila 2001).

$$Q(D_{PQ}) = \begin{bmatrix} -77.1476 & -3.0072 & -4.5016 & 7.9347 & -0.4665 & 2.8078 & 0 \\ -67.2098 & -3.9166 & 9.3613 & -1.6209 & -0.6805 & -3.0337 & 4.0000 \\ -4.0986 & 11.5020 & 5.6657 & -3.0036 & -2.1950 & 5.8030 & 4.0000 \\ -14.1784 & 11.9594 & -6.9816 & 0.1167 & 5.3415 & -1.0821 & 8.0000 \\ -9.9353 & 16.3832 & -0.5621 & -0.0651 & -5.1173 & 0.6564 & 0 \\ -10.9375 & 15.2266 & 6.3227 & 2.0865 & 7.2036 & 1.1301 & 4.0000 \end{bmatrix}$$

Figure 12: The $Q(D_{PQ})$ Matrix

Table 7: The Values of $||u-v||^2$ and $||f(u)-f(v)||^2$ for all u, v in D_Q (Approach With PCA)

u	**v**	$\|\|u-v\|\|^2$	$\|\|f(u)-f(v)\|\|^2$	$\frac{\|\|f(u)-f(v)\|\|^2}{\|\|u-v\|\|^2}$
1	2	434	433	0.998
1	3	5801	5798	0.999
1	4	4376	4369	0.998
1	5	5020	4999	0.996
1	6	4952	4945	0.999
2	3	4319	4317	1.000
2	4	3396	3391	0.999
2	5	3864	3843	0.995
2	6	3642	3636	0.998
3	4	395	392	0.992
3	5	185	173	0.935
3	6	211	197	0.934
4	5	272	255	0.938
4	6	238	226	0.950
5	6	230	222	0.965

Acknowledgments

Augustine Nsang's work was partially supported by the Department of the Navy, Grant ONR N000140710438.

References

Achlioptas, D. 2004. Random matrices in data analysis. In *Lecture Notes In Computer Science; Vol. 3202, Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases*, 1 – 7.

Bingham, E., and Mannila, H. 2001. Random projections in dimensionality reduction: Applications to image and text data. In *Conference on Knowledge Discovery in Data, Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and data mining*, 245–250.

Fradkin, D., and Madigan, D. 2003. Experiments with random projections for machine learning. In *Conference on Knowledge Discovery in Data, Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 517–522.

Johnson, W. B., and Lindenstrauss, J. 1984. Extensions of lipshitz mapping into hilbert space. *Contemporary Mathematics* 26:189–206.

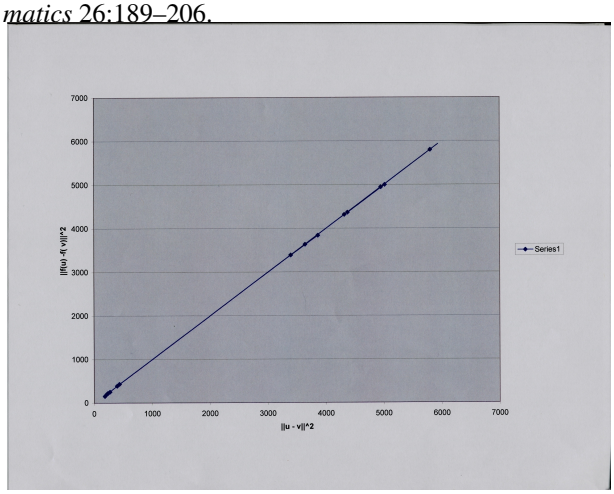

Figure 13: Graph of $||u-v||^2$ (on the x-axis) against $||f(u)-f(v)||^2$ (on the y-axis) for our alternative approach using PCA

Nsang, A., and Ralescu, A. 2009a. Query-based dimensionality reduction applied to text and web data. In *Proceedings of the Twentieth Midwest Artificial Intelligence and Cognitive Science Conference (MAICS 2009)*, 129–136.

Nsang, A., and Ralescu, A. 2009b. A review of dimensionality reduction methods and their applications. In *Proceedings of the Twentieth Midwest Artificial Intelligence and Cognitive Science Conference (MAICS 2009)*, 118–123.