Dynamic Dimensionality Reduction and Similarity Distance Computation by Inner Product Approximations

ÖMER EĞECIOĞLU* and HAKAN FERHATOSMANOĞLU

Department of Computer Science
University of California at Santa Barbara
{omer,hakan}@cs.ucsb.edu

Abstract

Developing efficient ways for dimensionality reduction is crucial for the query performance in multimedia databases. For approximation queries a careful analysis must be performed on the approximation quality of the dimensionality reduction technique. Besides having the lower-bound property, we expect the techniques to have good quality of distance measures when the similarity distance between two feature vectors is approximated by some notion of distance between two lower dimensional transformed vectors. Thus it is desirable to develop techniques which have accurate approximations to the original similarity distance when we eschew the lower-bound property.

In this paper, we develop dynamic dimensionality reduction based on the approximation of the standard inner-product. The method uses the power symmetric functions of the components of the vectors, which are powers of the p-norms of the vectors for $p = 1, 2, \ldots, m$. The number m of such norms used is a parameter of the algorithm whose simplest instance gives a first-order approximation implied by the Cauchy-Schwarz inequality. We show how to compute fixed coefficients that work as universal weights based on the moments of the probability density function assumed for the distribution of the components of the input vectors in the data set. If the distribution of the components of the vectors is not known we show how the method can be adapted to work dynamically by incremental adjustment of the parameters.

The experiments conducted on various distributions show that with this technique the similarity between two objects in high dimensional space can be accurately approximated by a significantly lower dimensional representation.

Keywords: *Distance approximation, dimensionality reduction, similarity search, inner-product, least-squares, symmetric function, p-norm.*

1 Introduction

Modern databases and applications use various new kinds of digital data, such as images, audio, video, maps, etc. Example of these applications include Geographical Information Systems (GIS) [9], Multimedia Information Systems [11, 21], CAD/CAM [7], medical imaging [10]. General approach is to represent the data objects as multidimensional points and to define the similarity between objects by the distance between the corresponding multidimensional points. It is assumed that the closer the points, the more similar the data objects. Both the dimensionality and the amount of data that need to be processed increases very rapidly.

*Supported in part by NSF Grant no. IIS-9817432.

Therefore, it becomes important to support high dimensional similarity searching in large-scale systems. This support depends on the development of efficient techniques to support approximate search techniques for high dimensional data sets. There have been several index structures for retrieval of multidimensional data. Examples of these include kdb-trees [20], hB-tree [17], R-tree [13], R*-tree [1], SS-tree [23], TV-tree [16], X-tree [6], and the Pyramid Technique [5]. Various algorithms for similarity searching have been developed associated to these indexing mechanisms. However, as dimensionality increases, query performance in these techniques degrades significantly [6, 4]. This problem is referred as the dimensionality curse [12] and has attracted the attention of several researchers.

A popular solution to the problem of dimensionality curse is dimensionality reduction. Several researchers use this approach for scalable query performance [14, 16, 19, 24, 15]. For example, in image databases, the dimensions of the feature vectors can be reduced to a desired value such that the underlying indexing technique performs optimally. There is a trade-off between the accuracy obtainable by the amount of information we store in the index structure and the efficiency we expect. It is well-known that if each image is represented by smaller dimensions significant performance speedups can be achieved. The most common approach in the literature for dimensionality reduction is to use a mathematical formulation such as the Discrete Fourier Transform (DFT), Discrete Wavelet Transform (DWT), and Singular Value Decomposition (SVD). These transformations are known to be distance preserving. The high dimensional original feature vectors are transformed into new vectors by using one of these transformations, and then lower dimensional vectors are created by taking the first few coefficients of the transformed vectors. These new vectors are the representatives of the the original data used in an indexing mechanism for efficient retrieval and for distance approximation methods. When the distance between the transformed vectors is less than the distance between the original feature vectors the lower-bound filtering property is said to hold [22]. By this property, if a point is in ϵ neighborhood of the query point, it still remains to be in ϵ neighborhood in the transformed domain, but since the query result is underestimated, the returned result set may contain extra data. These false hits are eliminated by checking the original distance. Therefore at the end the query result has only the data that is in the actual answer.

Exact results is difficult to obtain in several applications however. One obvious reason is the semantics expected from new applications. Most of the new applications need some notion of similarity to define reasonable queries. For instance, image databases may organize their data based on the content of the images. The user may want to search the database by describing some desired general properties, or by providing a sample image, and request a search for similar images to the described properties or similar to the image. For example, the QBIC project at IBM provides the ability to run queries based on colors, shapes, and sketches [19, 11]. Similarly, Alexandria Project at UC Santa Barbara provides similarity queries for texture data [18]. Another reason results from the time consumed to run queries on databases having huge amount of data. Exact queries in many data-intensive applications require a long period of time to execute. Moreover, as mentioned in Asilomar Report [8], imprecise information will not only appear as the output of queries, it already appears in data sources as well. For several applications, it is not unreasonable to define the queries with approximations. Consider a user submitting a query such as “Are there any really good Italian restaurants close to where I live?” [8]. There is no exact answer to this query since it is difficult to give a perfect definition of goodness and even closeness. In such instances it may be much more useful to provide an approximate answer to the given query.

For approximation queries a careful analysis must be given to the approximation quality of the dimensionality reduction technique. Besides having the lower-bound property, we expect the techniques to have good quality of distance measures. It is essential to develop techniques which have accurate approximations to the original similarity distance so that the similar objects in the original domain remain similar in the transformed domain.

In this paper, we develop dynamic dimensionality reduction techniques for efficient and accurate approximation of similarity evaluations between high dimensional vectors. By using these techniques, the similarity between two objects in high dimensional space can be accurately approximated by the lower dimensional transformed space. In some sense the techniques presented here are multidimensional analogues of the Cauchy-Schwarz inequality, which which can be thought of as a first order approximation to the inner product. The theoretical results we use are from [3] where fast dynamic methods for similarity by means of approximations to the inner-product using p-norms and minimization through least-squares methods were introduced.

The outline of this paper is as follows. In section 2 we introduce the power symmetric functions $\psi_p(x)$ and recast the lower-bound property satisfied by the Cauchy-Schwarz inequality in terms of these. Section 3 introduces the parameterized version of the proposed inner product approximation using the functions $\psi_p(x)$. The determination of the universal coefficients optimal in the sense of least-squares when the components of the vectors in the data set are drawn from the uniform distribution is covered in section 4. We give the proof of a special case ($m = 2$) in detail, and state the results for higher values. The experiments that show the quality of approximation of the method as a function of m are presented in section 5. Calculation of the optimal parameters for distributions other than uniform and exact formulas for various parametric distributions appear in sections 6 and 7, respectively. Theorem 1 is the major result of section 7, which characterizes the optimal parameters in terms of the moments of the assumed density function. Section 8 covers the nonparametric case. We show that it is possible to estimate the moments incrementally even when the distribution is not parametric. Section 9 presents the results of our experiments and comparisons with well-known methods such as SVD, DFT, and DCT for the uniform and normal distributions. Conclusions and future work appear in section 10.

2 Dimensionality reduction satisfying the lower-bound property

Developing efficient ways for dimensionality reduction is crucial for the query performance in multimedia databases. We first summarize how we represent the high dimensional data with reduced number of dimensions. Then we develop techniques for these representatives so that the similarity measure between high dimensional vectors are approximated in the lower dimensional space. We first briefly discuss, how these representatives can be used to guarantee the lower-bound property. Then, we focus on developing techniques which satisfy accurate approximations for the similarity distance between high dimensional objects, which is important for similarity searching. We assume that the dimensionality of the original vectors are n and we reduce the dimensionality to m.

For a given pair of integers $n, p > 0$ let

$$\psi_p(z) = z_1^p + z_2^p + \cdots + z_n^p. \tag{1}$$

This is the p-th power symmetric function in the variables $z = (z_1, z_2, \ldots, z_n)$. Equivalently, $\psi_p(z)$ is the p-th power of the p-norm $\|z\|_p$ which is defined as

$$\|z\|_p = \sqrt[p]{z_1^p + z_2^p + \cdots + z_n^p}$$

Thus $\psi_p(z) = \|z\|_p^p$. In particular the case $p = 2$ is the ordinary length of the vector z, and $\|x - y\|_2$ is the Euclidean distance between x and y. For each high dimensional vector x, we calculate the corresponding quantities $\psi_1(x), \psi_2(x), \ldots, \psi_m(x)$, and keep these m real numbers as a representative of the original vector x.

Let I^n denote the n-dimensional unit cube defined by $0 \leq z_j \leq 1$ for $j = 1, 2, \ldots, n$. Suppose $x, y \in I^n$ are two n-dimensional real vectors. The Euclidean distance between these two vectors is given by

$$distance(x, y) = \|x - y\|_2 = \sqrt{\psi_2(x) + \psi_2(y) - 2 < x, y >} \tag{2}$$

where $< x, y >$ is the standard inner product of the vectors x and y and defined as

$$< x, y >= x_1 y_1 + x_2 y_2 + \ldots + x_n y_n.$$

Since we already have $\psi_2(x)$ and $\psi_2(y)$ stored as a part of our dimensionality reduction, it is enough to compute $< x, y >$ to find the distance between two feature vectors. n dimensional vectors were reduced to m dimensional space as described above. It is desirable to compute an approximation for the inner product by using the summary information we already stored. By using these stored m values we will approximate $< x, y >$, and hence approximate the original distance. The exact values of $\psi_2(x)$ and $\psi_2(y)$, the approximation of the inner product and equation (2) are used for overall approximation.

In order to have lower-bound property the original distance needs to be estimated from below. If we find an upper bound for $< x, y >$ and use this value in equation (2), the approximated distance value computed in this way always becomes smaller than the original distance value because of the negative sign for $< x, y >$. The Cauchy-Schwarz inequality below gives an upper bound for the inner product:

$$< x, y > \;\; \leq \;\; \|x\|_2 \|y\|_2. \tag{3}$$

Since $\psi_2(z)$ is already defined as $\|z\|_2^2$, we have $< x, y > \;\; \leq \sqrt{\psi_2(x)\psi_2(y)}$. We approximate $< x, y >$ by $\sqrt{\psi_2(x)\psi_2(y)}$ which is always an upper bound to inner product. Therefore, by only storing the $\psi_2(z)$ values for each z in the database, it is possible to approximate the distance between them. Furthermore this particular first order approximation is guaranteed to have the lower-bound property.

3 Accurate approximations for similarity distance

The general approach in the current techniques, e.g. SVD, DFT, is to approximate the distance by satisfying the lower-bound property. In the previous section, we established the way that the new dimensionality reduction technique can be used to satisfy this property. In order to have lower-bound property the distance values must be always estimated from below, but because of the nature of the similarity queries, the quality of approximations is important. Therefore, in this paper, we also focus on the quality of the approximations, i.e. minimize the error made on the distance computations.

Since we already stored the exact values for $\psi_p(z)$ for each data vector z in the database, we look for an approximation for $< x, y >$ by using these values. The general form of the approximation is

$$< x, y >^m \;\; \approx \;\; b_1 \psi_1(x)\psi_1(y) + b_2 \psi_2(x)\psi_2(y) + \cdots + b_m \psi_m(x)\psi_m(y) \tag{4}$$

for large n, where the b_i are constants independent of x and y. Formally, we have table of large number of n-dimensional vectors $x = (x_1, , x_2, \ldots, x_n)$ whose components are independently drawn from a common distribution $F(t)$ with (possibly unknown) density $f(t)$. In other words each x_i is drawn independently of other coordinates from distribution $F(t)$. Given an arbitrary input vector $y = (y_1, y_2, \ldots, y_n)$, the main problem is to find the vectors x in the table minimizing (with high probability) the inner product $< x, y >$ without actually calculating all inner products. The approximation of the inner product of two vectors is used instead of the actual value. In the general case, the components do not need to satisfy $0 \leq z_j \leq 1$, nor do they have to be distributed identically.

A secondary problem is dynamic in nature. When the contents of the database changes, for instance by adding new data vectors, how can the parameters used for the approximation problem to the inner product calculation be recomputed/adjusted efficiently? We consider approximations of the form (4) by means of finding *the best set of constants* $b_1, b_2, \ldots, b_m$ for the approximation. We calculate and keep m real numbers for each vector x, alongside the fixed b_j's that are used for each approximation. For the input vector y, we calculate the corresponding quantities $\psi_1(y), \psi_2(y), \ldots, \psi_m(y)$ and use (4). If m can be taken much smaller than the dimension n with reasonable approximation to the inner product, besides efficiency gains in indexing, we also have an overall gain on the computation time for similarity checking. Note that in this formulation the quantities $\psi_p(z) = z_1^p + z_2^p + \cdots + z_n^p$ used in (4) are symmetric functions of the coordinates. A more general class of algorithms is obtained by taking instead $\psi_p(qz)$ in (4) where $qz = (q_1 z_1, q_2 z_2, \ldots, q_n z_n)$ with $q_j \geq 0$ and $q_1 + q_2 + \cdots + q_n = 1$. This has the effect of giving a degree of importance (weight) to individual features (components) of the vectors x and y. For computational simplicity we first look at the symmetric case in which $\psi_p(z)$ as given in (1) and $z \in I^n$. By taking each $q_j = 1/n$, we can write $\psi_p(z) = n^p \psi_p(qz)$, so the calculation of the symmetric case a particular instance.

4 Determination of the optimal parameters

The best approximation in the least-squares sense minimizes

$$\int \left[< x, y >^m - \sum_{j=1}^{m} b_j \psi_j(x) \psi_j(y) \right]^2 dx dy \tag{5}$$

where $dx = dx_1 dx_2 \cdots dx_n$, $dy = dy_1 dy_2 \cdots dy_n$, and the integral is over the $2n$-dimensional unit cube I^{2n}. The normal equations that $b_1, b_2, \ldots, b_m$ must satisfy are found by differentiating (5) with respect to each b_i, and setting the resulting expressions to zero. The resulting equations are that system $b_1, \ldots, b_m$ must satisfy are found to be

$$\sum_{j=1}^{m} \left[\int \psi_j(x) \psi_j(y) \psi_i(x) \psi_i(y) \, dx dy \right] b_j = \int < x, y >^m \psi_i(x) \psi_i(y) \, dx dy$$

where $1 \leq i \leq m$. This linear system which is satisfied by $b_1, \ldots, b_m$ can be written in the form $\mathbf{A}\mathbf{b} = \mathbf{c}$ where $\mathbf{A} = (a_{i,j})$ is an $m \times m$ matrix of real numbers $a_{i,j}$

$$a_{i,j} = \int \psi_j(x) \psi_j(y) \psi_i(x) \psi_i(y) \, dx dy \tag{6}$$

for $1 \leq i, j \leq m$, $\mathbf{c} = [c_1, \ldots, c_m]^t$ with

$$c_i = \int < x, y >^m \psi_i(x) \psi_i(y) \, dx dy \tag{7}$$

for $1 \leq i \leq m$, and $\mathbf{b} = [b_1, \ldots, b_m]^t$ is the column vector of the unknown coefficients.

4.1 The $m = 2$ case

Before we present the general case, we consider the 2×2 system that arises for $m = 2$ and work out the derivation of the asymptotic expansion coefficients b_1, b_2 in (4) in detail. The details of the proof of the general case will not be given here (see [3]). The linear system satisfied by b_1, b_2 is

$$\begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

with

$$\begin{aligned}
a_{1,1} &= \int_{I^{2n}} \psi_1(x)\psi_1(y)\psi_1(x)\psi_1(y)dxdy\ , \quad a_{2,2} = \int_{I^{2n}} \psi_2(x)\psi_2(y)\psi_2(x)\psi_2(y)dxdy \\
a_{1,2} = a_{2,1} &= \int_{I^{2n}} \psi_1(x)\psi_1(y)\psi_2(x)\psi_2(y)dxdy \\
c_1 &= \int_{I^{2n}} < x, y >^2 \psi_1(x)\psi_1(y)dxdy\ , \quad c_2 = \int_{I^{2n}} < x, y >^2 \psi_2(x)\psi_2(y)dxdy.
\end{aligned}$$

These quantities can be computed exactly as functions of n. First of all

$$\int_{I^n} \psi_1(x)\psi_1(x)dx \ = \ \sum_{k=1}^{n} \int_{I^n} x_k\psi_1(x)dx \ = \ n(\frac{n-1}{4} + \frac{1}{3}).$$

Similarly,

$$\int_{I^n} \psi_1(x)\psi_2(x)dx \ = \ n(\frac{n-1}{6} + \frac{1}{4}), \quad \int_{I^n} \psi_2(x)\psi_2(x)dx \ = \ n(\frac{n-1}{9} + \frac{1}{5}).$$

Therefore

$$\begin{aligned}
a_{1,1} &= \int_{I^n} \psi_1(x)\psi_1(x)dx \int_{I^n} \psi_1(x)\psi_1(y)dy \\
&= \left(\int_{I^n} \psi_1(x)\psi_1(x)dx\right)^2 \ = \ n^2(\frac{3n+1}{12})^2.
\end{aligned}$$

By a similar computation for $a_{2,2}$ and $a_{1,2}$, we find that the matrix of coefficients is

$$\begin{bmatrix} n^2(\frac{3n+1}{12})^2 & n^2(\frac{2n+1}{12})^2 \\ \\ n^2(\frac{2n+1}{12})^2 & n^2(\frac{5n+4}{45})^2 \end{bmatrix}$$

Next we calculate c_1 and c_2.

$$c_1 = \int_{I^{2n}} (\sum_{k=1}^{n} x_k y_k)^2 \psi_1(x)\psi_1(y)dxdy$$

There are two kinds of terms arising from the expansion of $(\sum x_k y_k)^2$. Diagonal terms of the form $x_r^2 y_r^2$, and off-diagonal terms of the form $x_r y_r x_s y_s$ for $r \neq s$. The contribution of the first kind of terms to c_1 is

$$n \int x_1^2 y_1^2 \psi_i(x)\psi_i(y)dxdy = n \left(\int x_1^2 \psi_i(x)dx\right)^2 = n \left(\frac{2n+1}{12}\right)^2.$$

It can be shown that off-diagonal terms contribute $n(n-1)\left(\frac{3n+2}{24}\right)^2$. Therefore

$$c_1 = n(\frac{2n+1}{12})^2 + n(n-1)(\frac{3n+2}{24})^2. \tag{8}$$

By a similar calculation, we find

$$c_2 = n(\frac{5n+4}{45})^2 + n(n-1)(\frac{n+1}{12})^2. \tag{9}$$

Therefore we have the following system for b_1, b_2:

$$\begin{aligned} n^2(\frac{3n+1}{12})^2 b_1 \ + \ n^2(\frac{2n+1}{12})^2 b_2 &= n(\frac{2n+1}{12})^2 + n(n-1)(\frac{3n+2}{24})^2 \\ n^2(\frac{2n+1}{12})^2 b_1 \ + \ n^2(\frac{5n+4}{45})^2 b_2 &= n(\frac{5n+4}{45})^2 + n(n-1)(\frac{n+1}{12})^2 \end{aligned} \quad (10)$$

Since we are looking for an asymptotic formula, it is tempting to let $n \to \infty$ in (10) and then solve the resulting numerical system for b_1, b_2 directly. Attempting to do this and simplifying the resulting equations gives the system

$$\begin{aligned} \frac{1}{4^2} b_1 + \frac{1}{6^2} b_2 &= \frac{1}{8^2} \\ \frac{1}{6^2} b_1 + \frac{1}{9^2} b_2 &= \frac{1}{12^2} \end{aligned}$$

which has determinant $6^2 12^2 - 8^2 9^2 = 0$ and is therefore singular. To circumvent this problem, we include not only the highest order term in n, but the second highest as well. This results in the (asymptotic) system

$$\begin{aligned} (\frac{n}{16} + \frac{1}{24}) b_1 \ + \ (\frac{n}{36} + \frac{1}{36}) b_2 &= \frac{n}{64} + \frac{19}{576} \\ (\frac{n}{36} + \frac{1}{36}) b_1 \ + \ (\frac{n}{81} + \frac{8}{405}) b_2 &= \frac{n}{144} + \frac{25}{1296} \end{aligned} \quad (11)$$

which is nonsingular for every n. Solving (11) symbolically for b_1 and b_2, we find

$$b_1 = \frac{9-n}{4(4n+1)} \ , \quad b_2 = \frac{5(9n-7)}{16(4n+1)} \ .$$

Therefore the limiting optimal values are

$$b_1 = -\frac{1}{16} \ , \quad b_2 = \frac{45}{64}. \quad (12)$$

As a result, this means that for $m = 2$, we approximate $< x, y >$ by the expression

$$\sqrt{\left| -\frac{1}{16}\psi_1(x)\psi_1(y) + \frac{45}{64}\psi_2(x)\psi_2(y) \right|} \ . \quad (13)$$

4.2 Uniform distribution: arbitrary m

For general m it can be shown [3] that

$$a_{i,j} = \int_{I^{2n}} \psi_i(x)\psi_i(y)\psi_j(x)\psi_j(y)dxdy \ = \ \frac{n^2(ij + n(i+j+1))^2}{(i+1)^2(j+1)^2(i+j+1)^2}$$

Therefore in the resulting matrix for general m, we see from the above expression that the (i, j)-th entry $a_{i,j}$ satisfies

$$a_{i,j} \sim \frac{n^4}{(i+1)^2(j+1)^2}.$$

But the matrix

$$(\frac{1}{(i+1)^2(j+1)^2})$$

again has rank 1, and therefore the system obtained by ignoring all but the highest degree of n that appears in the system we are required to solve is singular for $m > 1$. Fortunately, the inclusion of the second highest term works as before, albeit with more complicated calculations [3]. We omit the details of the derivation of the optimal coefficients $b_1, b_2, \ldots, b_m$ for $m > 2$.

For uniform distribution with $m = 2$ we obtained the approximation

$$< x, y >^2 \;\approx\; -\frac{1}{16}\psi_1(x)\psi_2(y) + \frac{45}{64}\psi_2(x)\psi_2(y),$$

in which the dimension n does not appear in the asymptotic expansion coefficients. For $m = 3$ the optimal least-squares approximation is

$$< x, y >^3 \;\approx\; -\frac{5}{16}n\psi_1(x)\psi_1(x) + \frac{3}{2}n\psi_2(x)\psi_2(y) - \frac{7}{6}n\psi_3(x)\psi_3(y),$$

and for $m = 4$

$$< x, y >^4 \;\approx\; -\frac{59}{256}n^2\psi_1(x)\psi_1(x) + \frac{1575}{1024}n^2\psi_2(x)\psi_2(y) - \frac{175}{64}n^2\psi_3(x)\psi_3(y) + \frac{1575}{1024}n^2\psi_4(x)\psi_4(y).$$

Values of $b_1, \ldots, b_m$ for various values of m for the uniform distribution appear in Figure 1.

m	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8
2	$-\frac{1}{16}$	$\frac{45}{64}$						
3	$-\frac{5}{16}n$	$\frac{3}{2}n$	$-\frac{7}{6}n$					
4	$-\frac{59}{256}n^2$	$\frac{1575}{1024}n^2$	$-\frac{175}{64}n^2$	$\frac{1575}{1024}n^2$				
5	$-\frac{31}{256}n^3$	$\frac{9}{8}n^3$	$-\frac{27}{8}n^3$	$\frac{135}{32}n^3$	$-\frac{297}{160}n^3$			
6	$-\frac{221}{4096}n^4$	$\frac{11025}{16384}n^4$	$-\frac{6125}{2048}n^4$	$\frac{202125}{32768}n^4$	$-\frac{24255}{4096}n^4$	$\frac{35035}{16384}n^4$		
7	$-\frac{89}{4096}n^5$	$\frac{45}{128}n^5$	$-\frac{275}{128}n^5$	$\frac{825}{128}n^5$	$-\frac{1287}{128}n^5$	$\frac{1001}{128}n^5$	$-\frac{2145}{896}n^5$	
8	$-\frac{535}{65536}n^6$	$\frac{43659}{262144}n^6$	$-\frac{43659}{32768}n^6$	$\frac{2837835}{524288}n^6$	$-\frac{3972969}{327680}n^6$	$\frac{3972969}{262144}n^6$	$-\frac{81081}{8192}n^6$	$\frac{1378377}{524288}n^6$

Figure 1. $< x, y >^m \;\approx\; b_1\psi_1(x)\psi_1(y) + \cdots + b_m\psi_m(x)\psi_m(y)$ **: asymptotic expansion coefficients** $b_1, b_2, \ldots, b_m$ **for the uniform distribution.**

5 Experiments: part 1

In the first set of experiments, we analyze the accuracy of the new approximation techniques by checking the error made in inner product calculations. As mentioned before, the accuracy of this approximation directly effects the quality of the similarity distance approximation. First consider the case $m = 2$ and the approximation given by (13). The graph of the average relative error made appears in Figure 2. The dimension n ranged from 2^4 to 2^{11}. For each dimension n, 100 pairs of vectors $x, y \in I^n$ were independently generated by drawing each coordinate from the uniform distribution on the unit interval I. The error calculated for n is the average relative error of these 100 experiments where the relative error of a single experiment is

$$\left| < x, y > - \sqrt{\left| -\frac{1}{16}\psi_1(x)\psi_1(y) + \frac{45}{64}\psi_2(x)\psi_2(y) \right|} \right| / < x, y > .$$

These are then accumulated and divided by the number of experiments.

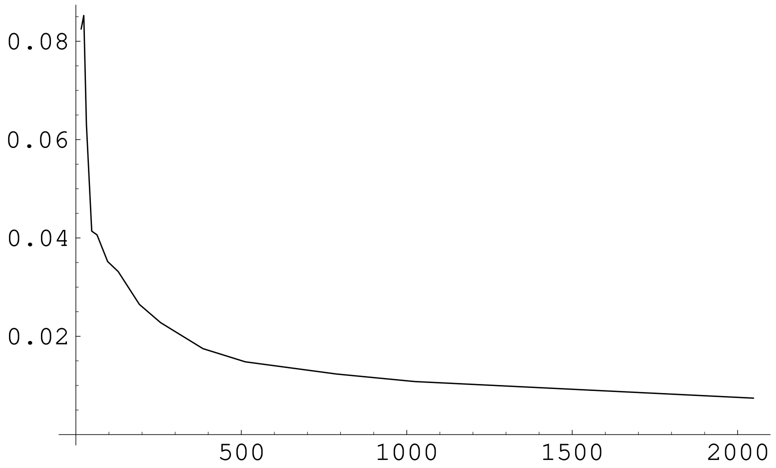

Figure 2. Average relative error versus dimension n, $16 \leq n \leq 2048$ for vectors from the uniform distribution with $m = 2$.

For the rest of the experiments of this type with larger values of m, again 100 pairs of vectors $x, y \in I^n$ were independently generated from the uniform distribution on I^n (each coordinate was drawn from the uniform distribution on the unit interval I). The dimension n ranged from 16 to 2048. The error calculated is the average relative error of these 100 experiments where the relative error of a single experiment is given by

$$\left| < x, y > - \left| \sum_{j=1}^{m} b_j \psi_j(x) \psi_j(y) \right|^{1/m} \right| / < x, y >$$

These relative errors are then accumulated and divided by the number of experiments performed. Figures 3 and 4 are the plots of the errors versus the original dimension for the approximations corresponding to reduced dimension m for $m = 2, 4, 6, 8$.

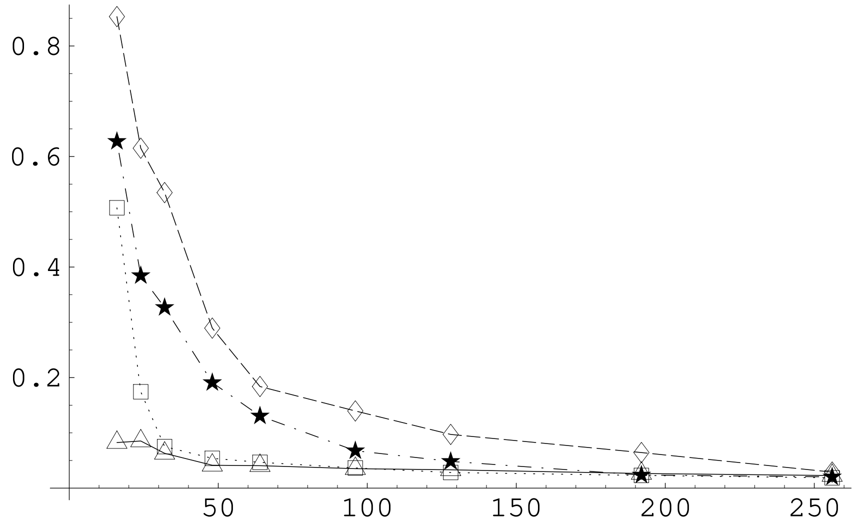

Figure 3. Average relative error versus dimension n**:** $16 \leq n \leq 256$**. (Legend:** $\triangle$ $m = 2$**,** $\square$ $m = 4$**,** $\star$ $m = 6$**,** $\diamond$ $m = 8$**)**

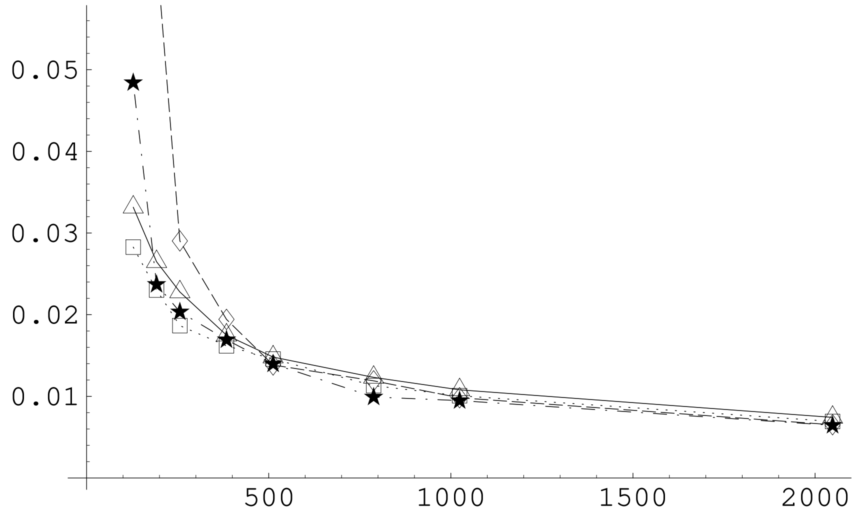

Figure 4. Average relative error versus dimension n**:** $128 \leq n \leq 2048$**. (Legend:** $\triangle$ $m = 2$**,** $\square$ $m = 4$**,** $\star$ $m = 6$**,** $\diamond$ $m = 8$**)**

6 Calculation of the optimal coefficients b_1, b_2 for other distributions

Suppose now that the coordinates of the vectors x and y are drawn from not the uniform distribution on the unit interval I, but some other distribution F on the real line. We assume that F has density f. Thus

$$F(t) = \int_{-\infty}^{t} f(x)dx \quad \text{with} \quad \int_{-\infty}^{\infty} f(x)dx \ = \ 1, \quad \text{and} \quad Pr\{a < x < b\} = \int_{a}^{b} f(x)dx.$$

The i-th moment μ_i of f (about the origin) is defined by

$$\mu_i = \int_{-\infty}^{\infty} x^i f(x)dx \ .$$

In minimizing the least squared error between $< x, y >^m$ and $\sum_{j=1}^{m} b_j \psi_j(x)\psi_j(y)$, the coefficients $b_1, \ldots, b_m$ to be determined satisfy a linear system $Ab = c$ where

$$a_{i,j} \ = \ \int \psi_j(x)\psi_j(y)\psi_i(x)\psi_i(y)dF(x)dF(y) \tag{14}$$

$$c_i \ = \ \int < x, y >^m \ \psi_i(x)\psi_i(y)dF(x)dF(y). \tag{15}$$

Lemma 1 *Suppose $a_{i,j}$ is as given in (14). Then*

$$a_{i,j} = n^2(\mu_{i+j} + (n-1)\mu_i\mu_j)^2.$$

Proof As before,

$$a_{i,j} = \left(\int_{\mathbb{R}^n} \psi_i(x)\psi_j(x)dF(x) \right)^2$$

and

$$\begin{aligned}
\int \psi_i(x)\psi_j(x)dF(x) \ &= \ n \int x_1^i \psi_j(x)dF(x) \\
&= \ n \int x_1^{i+j} f(x_1)dx_1 + n(n-1) \int x_1^i x_2^j f(x_1)f(x_2)dx_1dx_2 \\
&= \ n \int t^{i+j} f(t)dt + n(n-1) \left(\int t^i f(t)dt \right) \left(\int t^j f(t)dt \right) \\
&= \ n\mu_{i+j} + n(n-1)\mu_i\mu_j.
\end{aligned}$$

□

The quantities c_i for the $m = 2$ case are given in the following lemma.

Lemma 2 *Suppose c_i is as given in (15). Then*

$$\begin{aligned}
c_1 \ &= \ n\left[(n-1)\mu_1\mu_2 + \mu_3\right]^2 + n(n-1)\left[(n-2)\mu_1^3 + 2\mu_1\mu_2\right]^2 \\
c_2 \ &= \ n\left[(n-1)\mu_2^2 + \mu_4\right]^2 + n(n-1)\left[(n-2)\mu_1^2\mu_2 + 2\mu_1\mu_3\right]^2 .
\end{aligned} \tag{16}$$

Proof Omitted. □

These expressions for c_1, c_2 are a special case of a more general theorem that appears in [3]. For $m = 2$, using the expressions in (16) for c_1, c_2 and the 2×2 matrix of coefficients from lemma 1 for $m = 2$, we have

$$\begin{bmatrix} n^2[\mu_2 + (n-1)\mu_1^2]^2 & n^2[\mu_3 + (n-1)\mu_1\mu_2]^2 \\ n^2[\mu_3 + (n-1)\mu_1\mu_2]^2 & n^2[\mu_4 + (n-1)\mu_2^2]^2 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

Inverting this system symbolically and letting $n \to \infty$ we obtain the following result:

Theorem 1 *The constants b_1, b_2 which minimize*

$$\int_{\mathbb{R}^{2n}} \left[< x, y >^2 - b_1\psi_1(x)\psi_1(y) - b_2\psi_2(x)\psi_2(y) \right]^2 dF(x)dF(y)$$

are functions of the first four moments of the density $f(x)$. They are given by the formulas

$$b_1 = \mu_1^2 \cdot \frac{2\mu_2^3 + \mu_1^2\mu_4 - 3\mu_1\mu_2\mu_3}{\mu_2^3 - \mu_1^2\mu_4 - 2\mu_1\mu_2\mu_3}\,, \quad b_2 = \frac{\mu_1^4}{\mu_2} \cdot \frac{\mu_1\mu_3 - \mu_2^2}{\mu_2^3 - \mu_1^2\mu_4 - 2\mu_1\mu_2\mu_3}\,. \tag{17}$$

7 Explicit formulas for b_1, b_2 for various parametric distributions

Suppose the coordinates of $x, y \in \mathbb{R}^n$ are drawn identically and independently from a probability distribution with density function $f(x)$. In view of Theorem 1 and the explicit formulas for the approximation coefficients b_1, b_2 in the expansion

$$< x, y >^2 \;\approx\; b_1\psi_1(x)\psi_1(y) + b_2\psi_2(x)\psi_2(y)$$

can be found using (17) as soon as the first four moments of the density are known. For most common distributions, these moments are well known as explicit functions of the parameters of the distribution (see, for example [2]). Below, we work out a number of examples.

7.1 Power distribution

For a shape parameter c, the distribution function on $0 \le x \le 1$ is given by $F(x) = x^c$, with density function $f(x) = cx^{c-1}$. The i-th moment of $f(x)$ (around the origin) is given by $\mu_i = c/(c+i)$. From Theorem 1 we get

$$b_1 = -\frac{2c^3}{(c+1)^2(c^2+3c+4)}\,, \quad b_2 = \frac{c^2(c+2)^2(c+4)}{(c+1)^2(c^2+3c+4)}\,. \tag{18}$$

For $c = 1$, $f(x) = 1$ on $0 \le x \le 1$ and the distribution is uniform. In this case the formulas in (18) specialize to

$$b_1 = -\frac{1}{16}\,, \quad b_2 = \frac{45}{64}$$

which are the previously computed values for the uniform distribution given in (12).

7.2 Exponential distribution

For a scale parameter b, the distribution function on $0 \le x \le \infty$ is given by $F(x) = 1 - \exp(-x/b)$, with density function $f(x) = (1/b)\exp(-x/b)$. The i-th moment of $f(x)$ (around the origin) is $\mu_i = i!\, b^i$. From Theorem 1 we get

$$b_1 = \frac{b^2}{2}\,, \quad b_2 = \frac{1}{8}\,.$$

7.3 Binomial distribution

Let $0 \leq x \leq N$ be the number of successes in N independent Bernoulli trials where the probability of success at each trial is p and the probability of failure is $q = 1 - p$. The distribution function is $\sum_{i=0}^{x} \binom{N}{i} p^i q^{N-x}$ and the probability density function is $f(x) = \binom{N}{x} p^x q^{N-x}$. The first four moments of of $f(x)$ (around the origin) are given by

$$\begin{aligned} \mu_1 &= Np \\ \mu_2 &= Np((N-1)p+1) \\ \mu_3 &= Np((N-1)(N-2)p^2 + 3(N-1)p + 1) \\ \mu_4 &= Np((N-1)(N-2)(N-3)p^3 + 6(N-1)(N-2)p^2 + 7(N-1)p + 1). \end{aligned}$$

From Theorem 1 we get

$$b_1 = \frac{N^2p^2(1-2p)}{np - 3p + 2}, \quad b_2 = \frac{N^2p^2}{(np - p + 1)(np - 3p + 2)}.$$

7.4 Normal distribution

Normal distribution with mean μ and standard deviation σ has the probability density function

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp(\frac{-(x-\mu)^2}{2\sigma^2})$$

for $-\infty \leq x \leq \infty$. The first four moments of of $f(x)$ (around the origin) are given by

$$\mu_1 = \mu,\ \ \mu_2 = \mu^2 + \sigma^2,\ \ \mu_3 = \mu^3 + 3\mu\sigma^2,\ \ \mu_4 = \mu^4 + 6\mu^2\sigma^2 + 3\sigma^4\ .$$

From Theorem 1 we get

$$b_1 = \frac{2\mu^2\sigma^4}{\mu^4 + \sigma^4}, \quad b_2 = \frac{\mu^4(\mu^2 - \sigma^2)}{(\mu^2 + \sigma^2)(\mu^4 + \sigma^4)}.$$

7.5 Poisson distribution

Poisson distribution with parameter $\lambda > 0$ (the mean) has density function $f(x) = \lambda^x \exp(-\lambda)/x!$ for integer x in the range $0 \leq x \leq \infty$. The first four moments of $f(x)$ (around the origin) are given by

$$\mu_1 = \lambda,\ \ \mu_2 = \lambda(\lambda+1),\ \ \mu_3 = \lambda(\lambda^2 + 3\lambda + 1),\ \ \mu_4 = \lambda(\lambda^3 + 6\lambda^2 + 7\lambda + 1)\ .$$

From Theorem 1 we get

$$b_1 = \frac{\lambda^2}{\lambda + 2}, \quad b_2 = \frac{\lambda^2}{(\lambda+2)(\lambda+1)}.$$

7.6 Beta distribution

Beta distribution on $0 \leq x \leq 1$ has two shape parameters $v, w > 0$, and density function

$$f(x) = \frac{(v+w-1)!x^{v-1}(1-x)^{w-1}}{(v-1)!(w-1)!}$$

(for integer v, w). The i-th moment about the origin is given by

$$\mu_i = \prod_{r=0}^{i-1} (v + r)/(v + w + r).$$

Using the first four of these moments in Theorem 1 we get

$$b_1 = \frac{2v^2(w - v - 1)}{(v + w)^2((v + 1)^2 + (v + 3)w)}\ , \quad b_2 = \frac{v^2(w + v + 1)^2(w + v + 3)}{(v + w)^2((v + 1)^3 + (v + 1)(v + 3)w)}.$$

A summary of these calculations for $m = 2$ appears in Figure 5.

Distribution	Density $f(x)$	Range	b_1	b_2
Uniform	1	$0 \le x \le 1$	$-\frac{1}{16}$	$\frac{45}{64}$
Power	cx^{c-1}	$0 \le x \le 1$	$-\frac{2c^3}{(c+1)^2(c^2+3c+4)}$	$\frac{c^2(c+2)^2(c+4)}{(c+1)^2(c^2+3c+4)}$
Exponential	$(1/b)\exp(-x/b)$	$0 \le x \le \infty$	$\frac{b^2}{2}$	$\frac{1}{8}$
Binomial	$\binom{N}{x} p^x q^{N-x}$	$0 \le x \le N$	$\frac{N^2p^2(1-2p)}{np-3p+2}$	$\frac{N^2p^2}{(np-p+1)(np-3p+2}$
Normal	$\frac{1}{\sigma\sqrt{2\pi}}\exp(\frac{-(x-\mu)^2}{2\sigma^2})$	$-\infty \le x \le \infty$	$\frac{2\mu^2\sigma^4}{\mu^4+\sigma^4}$	$\frac{\mu^4(\mu^2-\sigma^2)}{(\mu^2+\sigma^2)(\mu^4+\sigma^4)}$
Poisson	$\lambda^x \exp(-\lambda)/x!$	$0 \le x \le \infty$	$\frac{\lambda^2}{\lambda+2}$	$\frac{\lambda^2}{(\lambda+2)(\lambda+1)}$
Beta	$\frac{(v+w-1)!x^{v-1}(1-x)^{w-1}}{(v-1)!(w-1)!}$	$0 \le x \le 1$	$\frac{2v^2(w-v-1)}{(v+w)^2((v+1)^2+(v+3)w)}$	$\frac{v^2(w+v+1)^2(w+v+3)}{(v+w)^2((v+1)^3+(v+1)(v+3)w)}$

Figure 5. $< x, y >^2 \ \approx\ b_1\psi_1(x)\psi_1(y) + b_2\psi_2(x)\psi_2(y)$ **: asymptotic expansion coefficients** b_1, b_2 **for various distributions.**

8 Non-parametric case: estimating the moments for an unknown density

If the coordinates of each vector x are drawn from a known parametric distribution family, then the parameters can be estimated by various methods and the moments computed as we indicated. Here we describe a method to estimate (and also incrementally update) the moments μ_i when the components are drawn from a distribution with an unknown density $f(t)$. As before, we assume that each coordinate of x is drawn independently from the corresponding distribution. By a transformation of the real line, we may also assume that f is identically zero outside the interval $0 \le t \le 1$.

The problem we address is the following. Suppose we know the empirical moments $\bar{\mu_i} = \bar{\mu_i}(N)$ of density $f(t)$, $0 \le t \le 1$, based on samples $t_1, t_2, \ldots, t_N$. Given t_{N+1}, how do we update $\bar{\mu_i}(N)$ to obtain the estimate $\bar{\mu_i}(N + 1)$? The idea is based on the following lemma.

Lemma 3 *An estimate of the moment μ_i under the assumptions above is given by*

$$\bar{\mu_i}(N) = \frac{1}{N}\sum_{j=1}^{N} x_j^i \tag{19}$$

Proof An estimate $F_N(t)$ for the distribution given the samples $t_1, t_2, \ldots, t_N$ is the histogram

$$F_N(t) = \frac{1}{N}|\{t_j \mid t_j < t\}|$$

where the bars denote cardinality. Integrating by parts, we find that

$$\bar{\mu_i}(N) = \int_0^1 t^i dF_N(t) = 1 - i\int_0^1 t^{i-1}F_N(t)dt \ = 1 - \frac{1}{N}\sum_{j=1}^{N} j\left(t_{j+1}^i - t_j^i\right)$$

where $t_{N+1} = 1$. This sum simplifies to (19) as stated. □

Using lemma 3, we can write $\bar{\mu_i}(N+1)$ in terms of $\bar{\mu_i}(N)$ and the $(N+1)$-st sample t_{N+1} as

$$\bar{\mu_i}(N+1) = \frac{1}{N+1}\left(N\bar{\mu_i}(N) + t_{N+1}^i\right). \tag{20}$$

This update rule takes on a particularly nice form when we think of a table of vectors and run this update rule for every vector incrementally, instead of individual components. Suppose currently there are r vectors present in the table, each n-dimensional with entries in the unit interval, drawn from a distribution with unknown density. Let $\bar{\mu_i}[r]$ denote the estimate of the i-th moment of this density obtained from the $N = nr$ samples which are the components of these r vectors. If x is the $(r+1)$-st vector, then the new estimate is obtained by the update rule

$$\bar{\mu_i}[r+1] = \frac{1}{r+1}\left(r\bar{\mu_i}[r] + \frac{1}{n}\psi_i(x)\right). \tag{21}$$

Note that for the $m = 2$ approximation we need to compute $\psi_1(x), \psi_2(x)$ anyway. To estimate the moments up to $i = 4$ (which are needed for the calculation of b_1, b_2 by theorem 1), by (21) we also compute $\psi_3(x), \psi_4(x)$, and $\psi_5(x)$.

9 Experiments: part 2

In the second part of the experiments, we compare the performance of our technique with current approaches. Singular Value Decomposition (SVD) and Discrete Fourier Transform (DFT) are the best known and the most widely used approaches in the literature. Here we also consider the Discrete Cosine Transform (DCT) for dimensionality reduction, which we found to be quite effective in our experiments. Although it is simpler than DFT, it gives better performance in most cases. We implemented our new algorithm, SVD, DFT, and DCT, and analyzed their relative approximation quality. We first compute the distance for each pair of vectors in the data set. This gives a general error measure for the similarity checking. For similarity joins, in the worst case the distance between each pair is computed and is compared to the give a threshold criteria of similarity, i.e. a bound ϵ. For similarity queries, instead of computing the distance between each each pair of vectors, the distances between the query point and all of the points in data set are computed. The query point may be chosen from the data set or can be specified by the user.

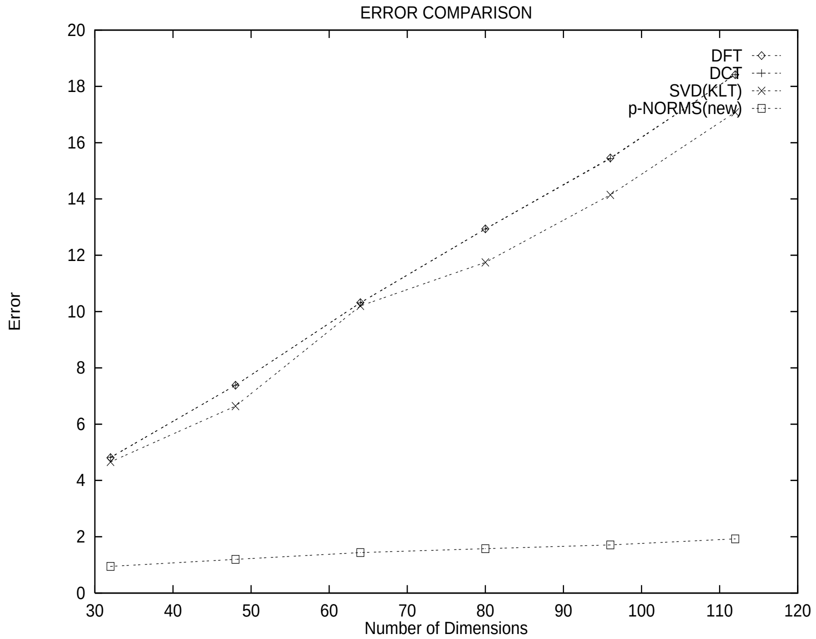

Figure 6. Error comparisons

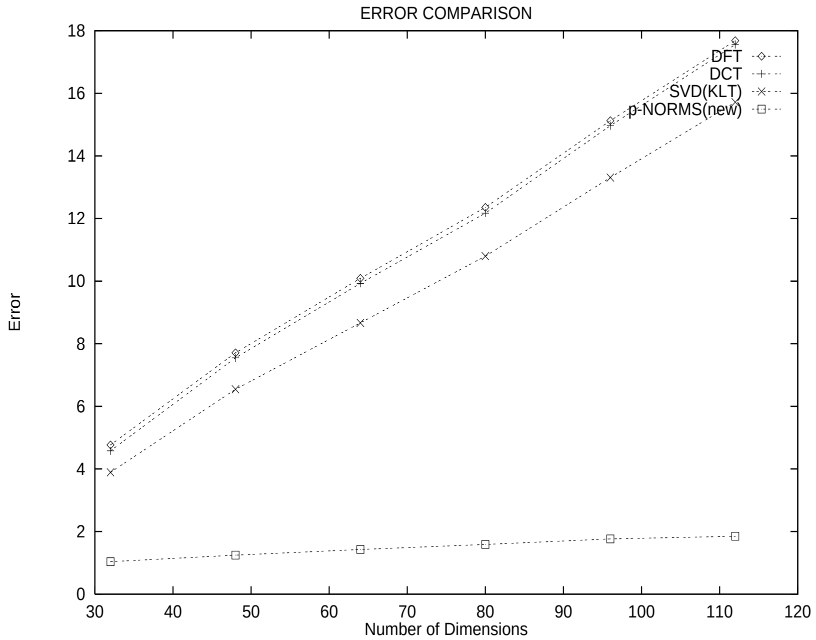

Figure 7. Error comparisons

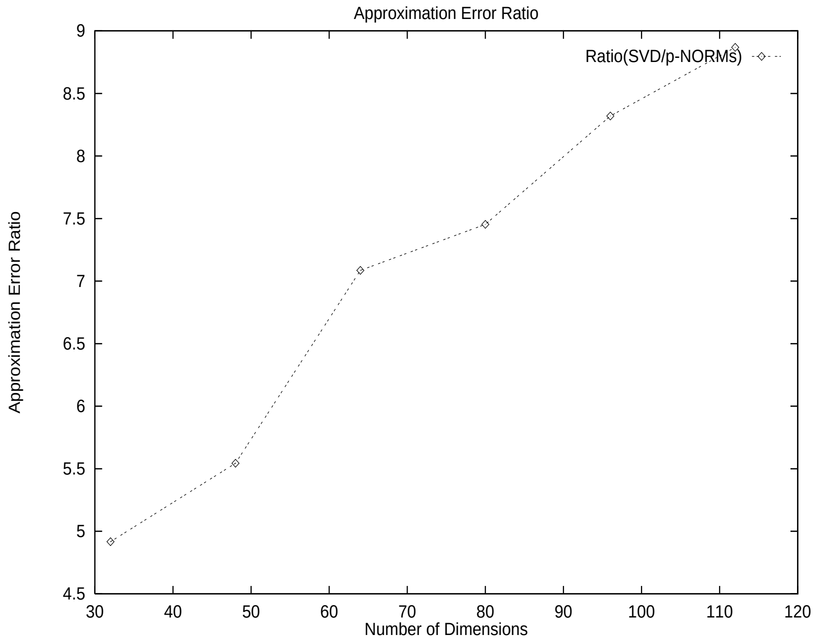

Figure 8. Scalability

Experiments on the uniform distribution

We generated 500 32-dimensional random points from the uniform distribution on I^{32}. Pairwise distances were calculated. In this way, all the possible distance computations in the database are done and the overall error of each method is measured. This value gives the quality of the approximations achieved by each technique. In the figures we refer to our technique as the *p-NORMS*. We used p-NORMS to reduce the dimensionality down to m and implemented the formulations that we developed in this paper to approximate the distance between objects. First we reduce the number of dimensions to $m = 2$ (for other techniques we reduce the dimensionality to 3, which is obviously more advantageous). Even when the other techniques use 3 coefficients, the approximation quality appears much worse than our technique. p NORMS gives an approximation error which is 5 times less than the current approaches for $n = 32$ dimensional vectors. The best average error for each distance computation is reached by SVD transformation and it is 4.66. On the other hand, the new approach gives an average error of 0.94, which implies a much better approximation.

We repeated the experiments for different dimensionalities by increasing the number of dimensions n and analyzing the resulting approximation quality. Figure 6 illustrates the measurements for each of SVD, DFT, DCT, and p-NORMS. Dimensionalities varied from $n = 32$ to $n = 128$. It can be seen that as n increases, the quality difference between p-NORMS and the other three approaches increases. For 80 dimensional data, for instance, the new technique's approximation is 7.45 times better than the current best approach. For 128 dimensional data, the average error of p-NORMS is 2.12 and the average error of the SVD technique, the best current, is 18.780. Thus the new approach has an error 8.86 times less than the best available. We repeated the experiments, this time taking $m = 4$ for all techniques. Figure 7 illustrates the results of these. Again the error in p-NORMS is 8.1 times less than SVD for 128 dimensions.

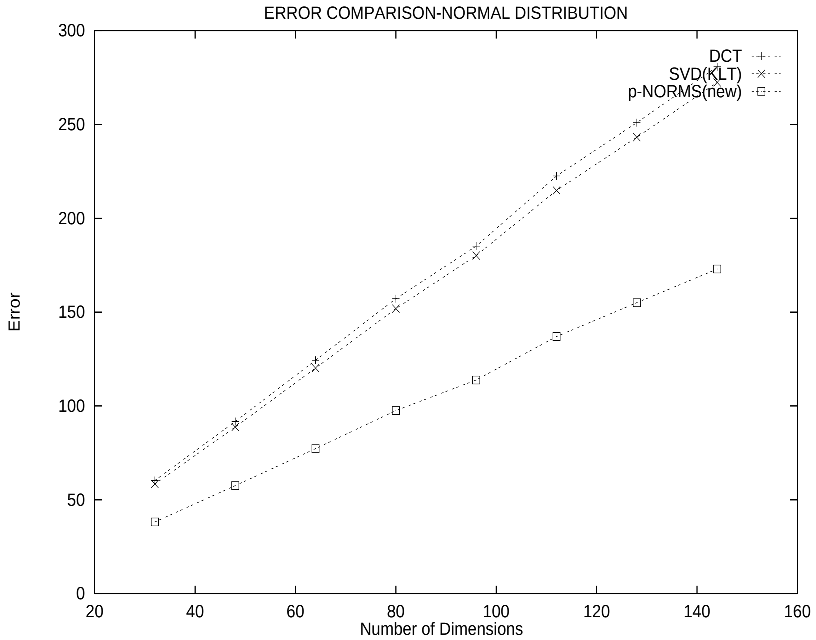

Figure 9. Error comparisons for normal distribution

Scalability experiments

In order to test the scalability of our approach, we analyzed the approximation quality ratio of our technique with SVD as dimensionality increases. Figure 8 illustrates the scalability of our approach as a function of dimensionality: As dimensionality increases, the quality achieved by p-NORMS becomes much better than the current approaches.

Experiments on the normal distribution

We also analyzed the quality of the approximations developed for data sets where the components are drawn from a normal distribution. We generated 500 random points from a normal distribution with mean 0.5 and variance 1. Approximations based on p-NORMS gave an error of 1.6 times lower than the best current technique, in this case SVD. Figure 9 illustrates the results of these experiments. For clarity, we did not show the results for DFT which has worse results than DCT. Our technique keeps its quality over the others as dimensionality increases.

10 Conclusions and future work

In this paper, we developed dynamic dimensionality reduction techniques for efficient and accurate approximation of similarity evaluations between high dimensional vectors. The method is based on the approximation of the standard inner product as a function of the p-norms of the vectors. A high dimensional real vector x of dimension n is represented as the sequence of values $(\psi_1(x), \psi_2(x), \ldots, \psi_m(x))$ where $\psi_p(x)$ is the p-th power of the p-norm of x. The magnitude of m controls the amount of dimension reduction made. Assuming that the components of the vectors in the data set are identically distributed, we find optimal universal constants $b_1, b_2, \ldots, b_m$ so that the approximation

$$< x, y >^m \;\approx\; b_1\psi_1(x)\psi_1(y) + b_2\psi_2(x)\psi_2(y) + \cdots + b_m\psi_m(x)\psi_m(y)$$

is the best possible for large n in the least-squares sense. This approximation is then used for estimating the inner product, and consequently for approximating the similarity distance between x and y. Even for $m = 2$, the performance is very good, and the approximation error is better than well-known methods such as the SVD, DFT, and DCT as verified by numerical simulations. We showed that if the components are from a distribution with a standard density, then the moments of the density directly determine the best constants. If the distribution of the components of the vectors is not known, then the method can be adapted to work dynamically by incremental adjustment of the parameters.

There are a number of issues and extensions we are currently pursuing. Among these are the implementation of the nonparametric estimation algorithm described in section 8 for large data sets; the analytic solution of the best constants when the distribution of the components of the vectors in the data set are described by some arbitrary probability vector $(q_1, q_2, \ldots, q_n)$, not necessarily weighted equally; and hybrid approaches which can take advantage of various methods currently available for dynamic dimensionality reduction and similarity distance computation.

References

[1] N. Beckmann, H. Kriegel, R. Schneider, and B. Seeger. The R* tree: An efficient and robust access method for points and rectangles. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 322–331, May 23-25 1990.

[2] N.A.J. Hastings and J.B. Peacock. *Statistical Distributions,* Halsted Press, New York, 1975.

[3] Ö. Egecioglu. How to approximate the inner-product: fast dynamic algorithms for similarity. *Technical Report TRCS98-37*, Department of Computer Science, University of California at Santa Barbara, December 1998.

[4] S. Berchtold, C. Bohm, D. Keim, and H. Kriegel. A cost model for nearest neighbor search in high-dimensional data space. In *Proc. ACM Symp. on Principles of Database Systems*, Tuscon, Arizona, 1997.

[5] S. Berchtold, C. Bohm, and H.-P. Kriegel. The Pyramid-Technique: Towards breaking the curse of dimensionality. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 142–153, Seattle, Washington, USA, June 1998.

[6] S. Berchtold, D. Keim, and H. Kriegel. The x-tree: An index structure for high-dimensional data. In *Proceedings of the Int. Conf. on Very Large Data Bases*, pages 28–39, Bombay, India, 1996.

[7] Kriegel H.-P. Berchtold S. S3: Similarity search in cad database systems. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 564–567, Tuscon, Arizona, 1997.

[8] P. Bernstein, M. Brodie, S. Ceri, D. DeWitt, M. Franklin, H. Garcia-Molina, J. Gray, J. Held, J. Hellerstein, H. Jagadish, M. Lesk, D. Maier, J. Naughton, H. Pirahesh, M. Stonebraker, and J. Ullman. The Asilomar report on database research, December 1998.

[9] X. Cheng, R. Dolin, M. Neary, S. Prabhakar, K. Ravikanth, D. Wu, D. Agrawal, A. El Abbadi, M. Freeston, A. Singh, T. Smith, and J. Su. Scalable access within the context of digital libraries. In *IEEE Proceedings of the International Conference on Advances in Digital Libraries, ADL*, pages 70–81, Washington, D.C., 1997.

[10] Korn F., Sidiropoulos N., Faloutsos C., Siegel E., and Protopapas Z. Fast nearest neighbor search in medical image databases. In *Proceedings of the Int. Conf. on Very Large Data Bases*, pages 215–226, Mumbai, India, 1996.

[11] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3:231–262, 1994.

[12] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. Fast subsequence matching in time-series databases. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 419–429, Minneapolis, May 1994.

[13] A. Guttman. R-trees: A dynamic index structure for spatial searching. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 47–57, 1984.

[14] D. Hull. Improving text retrieval for the routing problem using latent semantic indexing. In *Proc. of the 17th ACM-SIGIR Conference*, pages 282–291, 1994.

[15] K. V. Ravi Kanth, D. Agrawal, and A. Singh. Dimensionality reduction for similarity searching in dynamic databases. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, 1998.

[16] K. Lin, H. V. Jagadish, and C. Faloutsos. The TV-tree: An index structure for high-dimensional data. *VLDB Journal*, 3:517–542, 1995.

[17] D. B. Lomet and B. Salzberg. The hb-tree: A multi-attribute indexing method with good guaranteed performance. *ACM Transactions on Database Systems*, 15(4):625–658, December 1990.

[18] B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–42, August 1996.

[19] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, and P. Yanker. The QBIC project: Querying images by content using color, texture and shape. In *Proc. of the SPIE Conf. 1908 on Storage and Retrieval for Image and Video Databases*, volume 1908, pages 173–187, February 1993.

[20] J. T. Robinson. The kdb-tree: A search structure for large multi-dimensional dynamic indexes. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 10–18, 1981.

[21] T. Seidl and Kriegel H.-P. Efficient user-adaptable similarity search in large multimedia databases. In *Proceedings of the Int. Conf. on Very Large Data Bases*, pages 506–515, Athens, Greece, 1997.

[22] T. Seidl and H.P. Kriegel. Optimal multi-step k-nearest neighbor search. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, Chicago, Illinois, U.S.A., June 1998. ACM.

[23] D. White and R. Jain. Similarity indexing with the SS-tree. In *Proc. Int. Conf. Data Engineering*, pages 516–523, 1996.

[24] D. Wu, D. Agrawal, A. El Abbadi, and T. R. Smith. Efficient retrieval for browsing large image databases. In *Proc. Conf. on Information and Knowledge Management*, pages 11–18, November 1996.