

Dimensionality Reduction and Similarity Computation by Inner Product Approximations*

Ömer Egecioğlu and Hakan Ferhatosmanoğlu

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

Abstract

We develop dynamic dimensionality reduction based on the approximation of the standard inner-product. The inner-product, by itself, is used as a distance measure in a wide area of applications such as document databases, e.g. latent semantic indexing (LSI). A first order approximation to the inner-product is usually obtained from the Cauchy-Schwarz inequality. The method proposed in this paper refines such an approximation by using higher order power symmetric functions of the components of the vectors, which are powers of the p -norms of the vectors for $p = 1, 2, \dots, m$. 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. Our experiments on synthetic and document data show that with this technique, the similarity between two objects in high dimensional space for certain applications can be accurately approximated by a significantly lower dimensional representation.

Keywords: *Distance approximation, dimensionality reduction, similarity search, inner-product, document databases, p -norm.*

1 Introduction

Modern databases and applications use multiple types of digital data, such as documents, images, audio, video, etc. Some examples of such applications are document databases [7], medical imaging [17], and multimedia information systems [11, 21]. The general approach is to represent the data objects as multi-dimensional points and to measure the similarity between objects by the distance between the corresponding multi-dimensional points. It is assumed that the closer the points, the more similar the data objects. Since the dimensionality and the amount of data that need to be processed increases very rapidly, it becomes important to support efficient high dimensional similarity searching in large-scale sys-

tems. To this end, a number of index structures for retrieval of multi-dimensional data along with associated algorithms for similarity search have been developed [14, 2, 4]. However, it has also been noted that as dimensionality increases, query performance degrades significantly [3]. This anomaly 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 for scalable query performance [20, 18] in which the dimensions of the feature vectors are reduced to enhance the performance of the underlying indexing technique. Evidently there is a trade-off between the accuracy obtained from the information stored in the index structure and the efficiency obtained by the reduction. The most common approaches found in the literature for dimensionality reduction are linear-algebraic methods such as the Singular Value Decomposition (SVD), or applications of mathematical transforms such as the Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), or Discrete Wavelet Transform (DWT). In these methods, lower dimensional vectors are created by taking the first few leading coefficients of the transformed vectors [1].

Approximation methods in similarity queries have also attracted attention [18, 13]. It can be argued that approximate methods achieve efficiency at the expense of exact results. However exact results are difficult to obtain in several applications to begin with. One reason is that the generation of feature vectors from the original objects itself may be based on heuristics. Besides, the semantics expected from most application domains are not as strict as the exact queries used in relational databases. For example, the QBIC project at IBM provides the ability to run queries based on colors, shapes, and sketches [20, 11]. Similarly, Alexandria Project at UC Santa Barbara provides similarity queries for texture data [19]. As mentioned in the Asilomar Report on Database Research [5], imprecise information will not only appear as the output of queries, it already appears in data sources as well. For several applications, it is much more reasonable to define approximate queries; consider a user submitting a query such as “Are there any good Italian restaurants close to where I live?”. 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 is useful to provide an approximate answer to the given query.

In this paper, we develop dynamic dimensionality reduction techniques for efficient and accurate approximation of similarity evaluations between high dimensional vectors. More specifically, we focus on approximating the inner-product and consequently approximating the cosine of the angle be-

*This work was partially supported by the NSF under grants EIA98-18320, IIS-9817432, CCR-9821038, and IIS99-70700.

tween two vectors. To the best of our knowledge, there is no other technique for approximation of similarity computation based on inner-products. In some sense, the techniques presented here are the multi-dimensional analogues of the Cauchy-Schwarz inequality, which can be thought of as a first order approximation to the inner-product.

Approximating the inner-product, by itself, has a number of important applications. It is used extensively in the document database world, for example. Documents are compared in the semantic space by comparing their multi-dimensional representations created by statistical analysis, and their similarity are measured by the cosine of the angle between these vectors [23]. Latent Semantic Indexing (LSI) is a well-known example to applications that use the inner-product [16, 7, 8]. The dimensionality reduction and the inner-product approximation techniques proposed in this paper can effectively be used to approximate the original similarity in a reduced dimensional space in these and similar applications. Besides efficiency gains in indexing, we also want to have an overall gain on the computation time for similarity checking. In typical applications, the amount of data is huge, therefore efficient processing of similarity computation becomes more important. If the current trends continue, large organizations will have petabytes of data that need to be processed [5].

The outline of this paper is as follows. In section 2 we describe the main tools used in our reduction. Section 3 describes the calculation of the optimal coefficients for the uniform distribution. The first set of experiments appear in section 4. Optimal coefficients for distributions other than the uniform distribution, as well a dynamic update rule for the non-parametric case of an unknown distribution are given in section 5. Section 6 presents comparisons with well-known methods such as SVD, DFT, and DCT. Conclusions and future work appear in section 7.

The theoretical results we use are from [9] 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.

2 Reduction with power symmetric functions

We first summarize how we represent the high dimensional data of dimension n with reduced number of dimensions m with $m \ll n$. Then we develop techniques for these representations so that the similarity measure between high dimensional vectors are approximated closely in the lower dimensional space.

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

$$\psi_p(z) = z_1^p + z_2^p + \dots + z_n^p. \quad (1)$$

This is the p -th power symmetric function in the variables $z = (z_1, z_2, \dots, z_n)$. Equivalently, $\psi_p(z)$ is the p -th power of the p -norm $\|z\|_p$. In particular $\|z\|_2$ is the ordinary length of the vector z , and $\|x - y\|_2$ is the Euclidean distance between x and y . Note that the ordinary Euclidean distance between x and y and the power symmetric functions are related by

$$\|x - y\|_2 = \sqrt{\psi_2(x) + \psi_2(y) - 2 \langle x, y \rangle}, \quad (2)$$

where $\langle x, y \rangle$ is the standard inner-product given by $\langle x, y \rangle = x_1 y_1 + x_2 y_2 + \dots + x_n y_n$. The Cauchy-Schwarz inequality itself can be written in the form

$$\langle x, y \rangle^2 \leq \psi_2(x) \psi_2(y).$$

Using the quantities for $\psi_p(z)$ computed for each data vector z in the database, we look for an approximation for

$\langle x, y \rangle$ by approximating its m -th power in the form

$$\langle x, y \rangle^m \approx b_1 \psi_1(x) \psi_1(y) + b_2 \psi_2(x) \psi_2(y) + \dots + b_m \psi_m(x) \psi_m(y) \quad (3)$$

for large n , where the b_i are constants chosen independently of x and y . In our method for each high dimensional vector x , we calculate $\psi_1(x), \psi_2(x), \dots, \psi_m(x)$, and keep these m real numbers as a representative of the original vector x .

Our assumption on the structure of the data vectors is as follows: we have a table of a large number of n -dimensional vectors $x = (x_1, x_2, \dots, x_n)$ whose components are independently drawn from a common (but possibly unknown) distribution $F(t)$ with density $f(t)$. 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. Given an arbitrary input vector $y = (y_1, y_2, \dots, y_n)$, the main problem is to find the vectors x in the table minimizing (with high probability) the inner-product $\langle x, y \rangle$ without actually calculating all inner-products. This is done by computing $\psi_1(y), \psi_2(y), \dots, \psi_m(y)$ and then using the m stored quantities $\psi_1(x), \psi_2(x), \dots, \psi_m(x)$ via (3).

We consider approximations of the form (3) by finding the best set of constants b_1, b_2, \dots, b_m for the approximation in the sense of least-squares. 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 of large data sets. Note that just as the ordinary 2-norm used in the Cauchy-Schwarz inequality, the quantities $\psi_p(z)$ used in (3) are also symmetric functions of the coordinates. A more general class of algorithms is obtained by taking instead $\psi_p(qz)$ in (3) where $qz = (q_1 z_1, q_2 z_2, \dots, q_n z_n)$ with $q_j \geq 0$ and $q_1 + q_2 + \dots + q_n = 1$. This has the effect of giving a degree of importance (weight) to individual features of x and y . For computational simplicity we look at the symmetric case in this paper, in which $\psi_p(z)$ is as given in (1) and $z \in I^n$, the n -dimensional unit cube. 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 is a particular instance.

3 Determination of the optimal parameters

The best approximation in the least-squares sense minimizes

$$\int \left[\langle x, y \rangle^m - \sum_{j=1}^m b_j \psi_j(x) \psi_j(y) \right]^2 dx dy \quad (4)$$

where $dx = dx_1 dx_2 \dots dx_n$, $dy = dy_1 dy_2 \dots dy_n$, and the integral is over the $2n$ -dimensional unit cube I^{2n} . The so-called *normal equations* that b_1, b_2, \dots, b_m must satisfy are found by differentiating (4) with respect to each b_i , and setting the resulting expressions to zero.

This results in an $m \times m$ linear system that b_1, \dots, b_m must satisfy, obtained from

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

$$\int \langle x, y \rangle^m \psi_i(x) \psi_i(y) dx dy$$

for $1 \leq i \leq m$. Putting

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

$$c_i = \int \langle x, y \rangle^m \psi_i(x) \psi_i(y) dx dy,$$

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: $\langle x, y \rangle^m \approx b_1 \psi_1(x)\psi_1(y) + \dots + b_m \psi_m(x)\psi_m(y)$: asymptotic expansion coefficients b_1, b_2, \dots, b_m for the uniform distribution.

we find that b_1, \dots, b_m satisfy the $m \times m$ linear system $\mathbf{A}\mathbf{b} = \mathbf{c}$.

We present the mathematical treatment for the case of the 2×2 system that arises for $m = 2$, and work out in detail the derivation of the asymptotic expansion coefficients b_1, b_2 in (3). The details of the proof of the general case can be found in [9]. For $m = 2$,

$$\begin{aligned}
a_{1,1} &= \int_{I^{2n}} \psi_1(x)\psi_1(y)\psi_1(x)\psi_1(y) dx dy \\
a_{2,2} &= \int_{I^{2n}} \psi_2(x)\psi_2(y)\psi_2(x)\psi_2(y) dx dy \\
a_{1,2} = a_{2,1} &= \int_{I^{2n}} \psi_1(x)\psi_1(y)\psi_2(x)\psi_2(y) dx dy \\
c_1 &= \int_{I^{2n}} \langle x, y \rangle^2 \psi_1(x)\psi_1(y) dx dy \\
c_2 &= \int_{I^{2n}} \langle x, y \rangle^2 \psi_2(x)\psi_2(y) dx dy.
\end{aligned}$$

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

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

Similarly,

$$\begin{aligned}
\int_{I^n} \psi_1(x)\psi_2(x) dx &= n \left(\frac{n-1}{6} + \frac{1}{4} \right), \\
\int_{I^n} \psi_2(x)\psi_2(x) dx &= n \left(\frac{n-1}{9} + \frac{1}{5} \right).
\end{aligned}$$

Therefore

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

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

matrix of coefficients is

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

Next we compute the quantities c_1 and c_2 in terms of n . We have

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

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

$$n \int x_1^2 y_1^2 \psi_1(x)\psi_1(y) dx dy = n \left(\int x_1^2 \psi_1(x) dx \right)^2 = n \left(\frac{2n+1}{12} \right)^2.$$

It can be shown that off-diagonal terms contribute

$$n(n-1) \int x_1 y_1 x_2 y_2 \psi_1(x)\psi_1(y) dx dy =$$

$$n(n-1) \left(\int x_1 x_2 \psi_1(x) dx \right)^2 = n(n-1) \left(\frac{3n+2}{24} \right)^2.$$

Therefore

$$c_1 = n \left(\frac{2n+1}{12} \right)^2 + n(n-1) \left(\frac{3n+2}{24} \right)^2. \quad (5)$$

By a similar calculation, we find

$$c_2 = n \left(\frac{5n+4}{45} \right)^2 + n(n-1) \left(\frac{n+1}{12} \right)^2. \quad (6)$$

The resulting system satisfied by b_1, b_2 is

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

$$n\left(\frac{5n+4}{45}\right)^2 + n(n-1)\left(\frac{n+1}{12}\right)^2$$

Since we are interested in these approximations for large n , it is tempting to let $n \rightarrow \infty$ in the resulting linear system and then solve for b_1, b_2 directly to obtain an asymptotic formula. Attempting to do this results in a singular system, however. To circumvent this problem, we include not only the highest order term in n , but the second highest as well. This gives in the (asymptotic) system

$$\begin{aligned} \left(\frac{n}{16} + \frac{1}{24}\right)b_1 + \left(\frac{n}{36} + \frac{1}{36}\right)b_2 &= \frac{n}{64} + \frac{19}{576} \\ \left(\frac{n}{36} + \frac{1}{36}\right)b_1 + \left(\frac{n}{81} + \frac{8}{405}\right)b_2 &= \frac{n}{144} + \frac{25}{1296} \end{aligned} \quad (7)$$

which is nonsingular for every n . Solving (7) symbolically for b_1 and b_2 and taking limits, we find

$$b_1 = \frac{9-n}{4(4n+1)} \rightarrow -\frac{1}{16}, \quad b_2 = \frac{5(9n-7)}{16(4n+1)} \rightarrow \frac{45}{64}.$$

This means that for $m=2$, we approximate $\langle x, y \rangle$ by the expression

$$\sqrt{\left| -\frac{1}{16}\psi_1(x)\psi_1(y) + \frac{45}{64}\psi_2(x)\psi_2(y) \right|}. \quad (8)$$

For general m it can be shown [9] that

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

This matrix again has rank 1, but the inclusion of the second highest term works as before [9]. We omit the details of the derivation of the optimal coefficients b_1, b_2, \dots, b_m for $m > 2$.

Values of b_1, \dots, b_m we have computed for various values of m for the uniform distribution appear in Figure 1. For the uniform distribution coefficients with $m=2$, the approximation (8) we obtained does not involve the dimension n . This is not the case for $m > 2$. For instance for $m=3$ the optimal least-squares approximation to $\langle x, y \rangle^3$ is

$$-\frac{5}{16}n\psi_1(x)\psi_1(x) + \frac{3}{2}n\psi_2(x)\psi_2(y) - \frac{7}{6}n\psi_3(x)\psi_3(y).$$

4 Experiments: part 1

In the first set of experiments, we analyze the accuracy of the approximation technique introduced here by checking the error made in inner-product calculations, keeping in mind that the inner-product is directly used as distance measure in several applications, e.g. LSI.

First consider the case $m=2$ and the approximation given by (8). 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 given by

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

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

For 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 . Figure 3 is the plot of the error versus the original dimension for the approximations corresponding to reduced dimension m for $m=2, 4, 6, 8$, and dimension n ranging from 16 to 256. Note that as m increases, the corresponding approximation method produces larger error for small n , but eventually dips below the error curves for smaller m . The reason for this is the asymptotic nature of the constants b_1, b_2, \dots, b_m .

5 Optimal b_1, b_2 for various 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

$$F(t) = \int_{-\infty}^t f(x)dx \quad \text{with} \quad \int_{-\infty}^{\infty} f(x)dx = 1,$$

and $Pr\{a < x < b\} = \int_a^b f(x)dx$. The i -th moment μ_i of f (about the origin) is defined by

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

We have the following general result, whose proof can be found in [9].

Theorem 1 *The constants b_1, b_2 which minimize*

$$\int_{\mathbb{R}^{2n}} [\langle x, y \rangle^2 - b_1\psi_1(x)\psi_1(y) - b_2\psi_2(x)\psi_2(y)]^2 dF(x)dF(y)$$

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

$$\begin{aligned} b_1 &= \mu_1^2 \cdot \frac{2\mu_2^3 + \mu_1^2\mu_4 - 3\mu_1\mu_2\mu_3}{\mu_2^3 - \mu_1^2\mu_4 - 2\mu_1\mu_2\mu_3}, \\ b_2 &= \frac{\mu_1^4}{\mu_2} \cdot \frac{\mu_1\mu_3 - \mu_2^2}{\mu_2^3 - \mu_1^2\mu_4 - 2\mu_1\mu_2\mu_3}. \end{aligned}$$

In view of Theorem 1, explicit formulas for the approximation coefficients b_1, b_2 in the expansion

$$\langle x, y \rangle^2 \approx b_1\psi_1(x)\psi_1(y) + b_2\psi_2(x)\psi_2(y)$$

can be found using Theorem 1 as soon as the first four moments of the density are known. For most common distributions, these moments can be calculated explicitly as functions of the parameters of the distribution (see, for example [15]).

A summary of these calculations for power, exponential, binomial, normal, Poisson, and Beta distributions appears in Figure 4. The last two columns are the optimal values of b_1 and b_2 expressed in terms of the parameters of the corresponding distribution.

When the components are drawn from a distribution with an unknown density $f(t)$ (i.e. in the non-parametric case) we can estimate and incrementally update estimates for the moments μ_i . If we know the empirical moments $\bar{\mu}_i = \bar{\mu}_i(N)$ of density $f(t)$, $0 \leq t \leq 1$, based on samples t_1, t_2, \dots, t_N , it can be shown that given t_{N+1} , we can obtain the estimate $\bar{\mu}_i(N+1)$ by

$$\bar{\mu}_i(N+1) = \frac{1}{N+1} (N\bar{\mu}_i(N) + t_{N+1}^i). \quad (9)$$

and adjust the b_i accordingly, using Theorem 1.

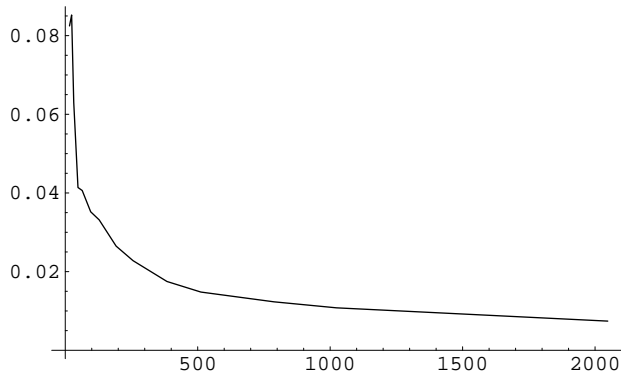


Figure 2: Average relative error versus dimension n , $16 \leq n \leq 2048$ for vectors from the uniform distribution with $m = 2$.

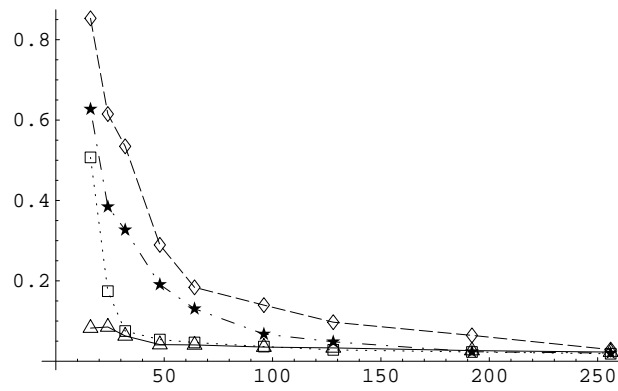


Figure 3: Average relative error versus dimension n : $16 \leq n \leq 256$. (Legend: $\triangle m = 2$, $\square m = 4$, $\star m = 6$, $\diamond m = 8$)

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

Figure 4: $\langle x, y \rangle^2 \approx b_1 \psi_1(x) \psi_1(y) + b_2 \psi_2(x) \psi_2(y)$: optimal asymptotic expansion coefficients b_1, b_2 for various parametric distributions.

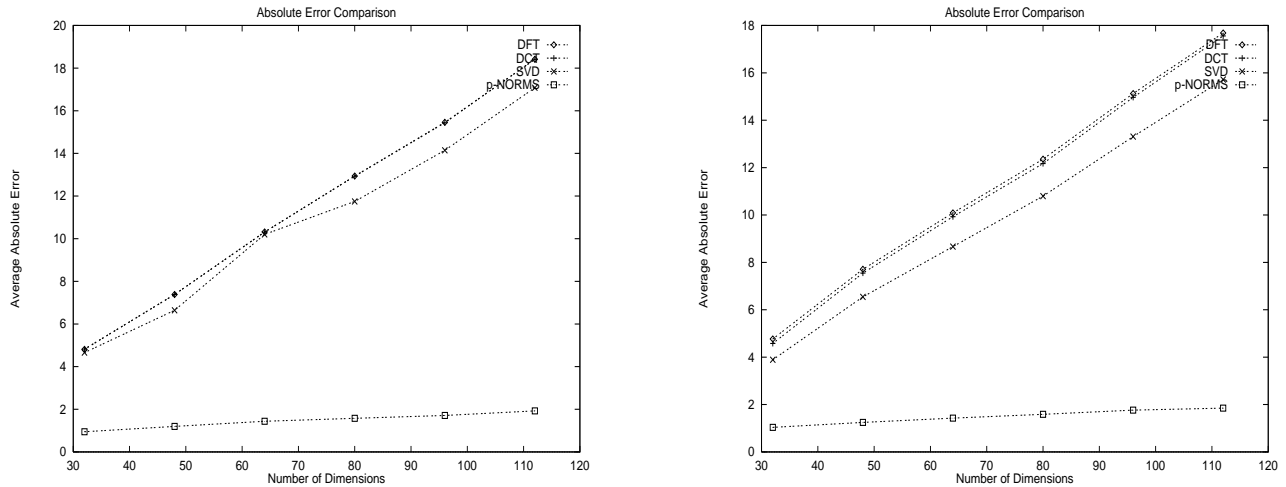


Figure 5: Error comparisons of dimensionality reduction techniques (for $m = 2$ and $m = 4$)

6 Experiments: part 2

The techniques presented in this paper can be readily used for approximation of the similarity also with respect to Euclidean distance metric. Suppose $x, y \in I^n$ are two n -dimensional real vectors. We use the expression (2) for the Euclidean distance between x and y . Since we already have $\psi_2(x)$ and $\psi_2(y)$ stored as a part of our dimensionality reduction, it is enough to compute $\langle x, y \rangle$ to find the distance between two feature vectors. By using the stored m values we approximate $\langle x, y \rangle$, and hence approximate the original distance.

Next, we compare the performance of our technique that we refer to as p -NORMS, with current approaches on real and synthetic data sets. 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. We implemented SVD, DFT, and DCT, and our new algorithm, and analyzed their approximation quality for distance measurements. We first compute the distance for each pair of data vectors in the data set. A motivation for this is similarity joins, in which in the worst case the distance between each pair is computed and is compared to a given threshold criteria of similarity. For similarity queries, instead of computing the distance between each pair of vectors, the distances between the query point and all of the points in the data set are computed. The query point may be chosen from the data set or can be specified by the user.

In the experiments, pairwise distances of the data vectors are computed. We use SVD, DFT, DCT, and p -NORMS to reduce the dimensionality of high dimensional vectors. Reduced dimensional vectors are representatives of original high dimensional vectors. We compute the distance between each pair of vectors of smaller dimensions. The real distance is approximated in reduced dimensional space. For each technique, we compute the *absolute error*, i.e. difference between approximated distance to real distance, for each pair of vectors. First the summation of the errors for all pairs is computed, then this value is divided by the number of pairs, i.e. the number of distance calculations (Note that, in the first part of the experiments, error metric was the average of relative errors which is a different metric).

In the first setup, we generated 500 32-dimensional random points from the uniform distribution on I^{32} . Pairwise distances are calculated both for original data and reduced dimensional data. For each technique, absolute approximation error of each distance calculation is summed and divided by the total number of pairs (25,000). This calculated average error gives the quality of the approximations achieved by each technique. First, we reduce the number of dimensions to $m = 2$. For other techniques, we reduce the dimensionality to 3 because the DFT technique produces complex numbers therefore the second component has actually two floating numbers. Even when the other techniques use 3 coefficients, in this case their 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 lowest error (4.6) among the implemented transform methods is made by SVD. On the other hand, p -NORMS has an average absolute error of only 0.9.

We repeated the experiments by increasing the number of dimensions n and analyzing the resulting approximations. The left figure in Figure 5 illustrates the measurements for each of SVD, DFT, DCT, and p -NORMS. Since we use average of absolute errors, the error naturally increases as dimensionality increases. However, it can be seen that as n increases, the quality difference between p -NORMS and the other three also 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 absolute error of p -NORMS is 2.1 and the average absolute error of the SVD technique, the best of the three is 18.8. Similar experiments with $m = 4$ for all techniques were also performed. The right figure in figure 5 illustrates the results of these. The error in p -NORMS is about 8 times less than that of SVD for 128 dimensions.

We also compute the approximation quality ratio of our technique with SVD, on the same data set, as dimensionality increases in order to illustrate the scalability of our approach. Figure 6 illustrates the superiority of p -NORMS over SVD as a function of dimensionality.

We 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. We note that since the data is not restricted to be within the range $[0..1]$ as

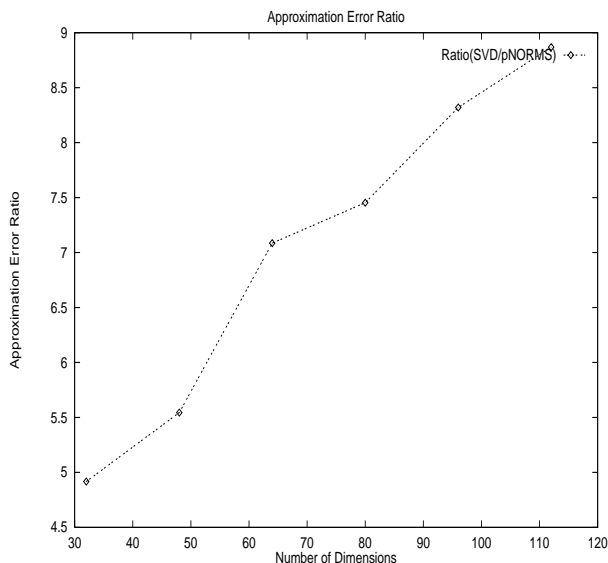


Figure 6: Scalability Comparison of SVD and p-NORMS

before, there are dimensions that are much greater than 1 in the data set. Therefore the absolute errors of the experiments are greater than the previous cases. Approximations based on p -NORMS gave an error 1.6 times lower than the best of the three other techniques, in this case SVD. Figure 7 illustrates the results of these experiments.

The techniques were also compared on a *real document database* of substrings from a large set of documents consisting of normalized data in I^{16} . We reduce the number of dimensions to $m = 2$ which can be indexed efficiently [1] by spatial indexing techniques. Increasing m will increase the accuracy of the approximations, but also will increase the index-search time. Similar to the synthetic data case, we computed the pairwise distances and took the average of absolute errors made by low-dimensional distance computations. Approximations based on p -NORMS performs twice as well as SVD and 2.2 times better than DCT. We note that SVD performs better than DCT on real data as well. Further experimental results with graphs on the performance of p -NORMS can be found in [10].

7 Conclusions and future work

We developed dynamic dimensionality reduction techniques for efficient and accurate approximation of similarity measures between high dimensional vectors. The method is based on the approximation of the standard inner-product as a certain function of the p -norms of the vectors. A high dimensional real vector x of dimension n is represented as the sequence of values $(\psi_1(x), \psi_2(x), \dots, \psi_m(x))$ where $\psi_p(x)$ is the p -th power of the p -norm of x . The magnitude of m controls the magnitude of the 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, \dots, b_m so that the approximation

$$\langle x, y \rangle^m \approx b_1 \psi_1(x) \psi_1(y) + b_2 \psi_2(x) \psi_2(y) + \dots + b_m \psi_m(x) \psi_m(y)$$

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

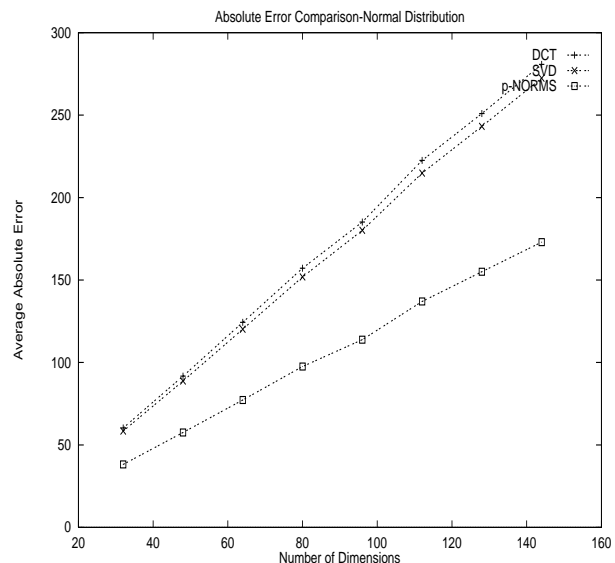


Figure 7: Error comparisons for normal distribution

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. Further details of our technique can be found in [9, 10].

There are a number of issues and extensions we are currently pursuing. Among these are 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, \dots, q_n) , properties of the non-parametric update rule given in (9), and hybrid approaches which can take advantage of various methods currently available for dynamic dimensionality reduction and similarity distance computation.

References

- [1] R. Agrawal, C. Faloutsos, and A. Swami. Efficient similarity search in sequence databases. In *4th Int. Conference on Foundations of Data Organization and Algorithms*, pages 69–84, 1993.
- [2] 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.
- [3] 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*, Tucson, Arizona, 1997.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] S. Deerwester, S.T. Dumais, G.W.Furnas, T.K. Launder, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407, 1990.
- [8] S. T. Dumais. Improving the retrieval of information from external sources. *Behavior Research Methods, Instruments and Computers*, 23:229–236, 1991.
- [9] Ö. Egecioğlu. 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.
- [10] Ö. Egecioğlu and H. Ferhatosmanoğlu, Dynamic Dimensionality Reduction and Similarity Distance Computation by Inner Product Approximations, *Technical Report TRCS99-20*, Department of Computer Science, University of California at Santa Barbara, June 1999.
- [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. Gionis, P. Indyk, and R. Motwani. Similarity searching in high dimensions via hashing. In *Proceedings of the Int. Conf. on Very Large Data Bases*, pages 518–529, Edinburgh, Scotland, UK, September 1999.
- [14] 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.
- [15] N.A.J. Hastings and J.B. Peacock. *Statistical Distributions*, Halsted Press, New York, 1975.
- [16] 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.
- [17] F. Korn, N. Sidiropoulos, C. Faloutsos, E. Siegel, and Z. Protopapas. 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.
- [18] 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.
- [19] 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.
- [20] 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. 1998 on Storage and Retrieval for Image and Video Databases*, volume 1908, pages 173–187, February 1993.
- [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] V.S. Subrahmanian. *Principles of Multimedia Database Systems*. Morgan Kaufmann Publishers, Inc., San Francisco, California, 1999.