Dimensionality Reduction and Similarity Computation by Inner-Product Approximations

Ömer Egecioglu, Hakan Ferhatosmanoglu, *Member*, *IEEE*, and Umit Ogras, *Student Member*, *IEEE*

Abstract—As databases increasingly integrate different types of information such as multimedia, spatial, time-series, and scientific data, it becomes necessary to support efficient retrieval of multidimensional data. Both the dimensionality and the amount of data that needs to be processed are increasing rapidly. Reducing the dimension of the feature vectors to enhance the performance of the underlying technique is a popular solution to the infamous curse of dimensionality. 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 resulting in accurate approximations to the original similarity distance. In this paper, we investigate dimensionality reduction techniques that *directly target minimizing the errors* made in the approximations. In particular, we develop dynamic techniques for efficient and accurate approximation of similarity evaluations between high-dimensional vectors based on *inner-product approximations*. Inner-product, by itself, is used as a distance measure in a wide area of applications such as document databases. A first order approximation to the inner-product is obtained from the Cauchy-Schwarz inequality. We extend this idea to higher order power symmetric functions of the multidimensional points. We show how to compute fixed coefficients that work as universal weights based on the moments of the probability density function of the data set. We also develop a *dynamic model* to compute the universal coefficients for data sets whose *distribution is not known*. Our experiments on synthetic and real data sets show that the similarity between two objects in high-dimensional space can be accurately approximated by a significantly lower-dimensional representation.

Index Terms—Inner-product approximation, dimensionality reduction, p-NORMS, similarity search, high-dimensional data.

1 Introduction

Size of the data utilized in modern applications grows at an increasing rate. For example, the number of documents that can be reached through the Internet is increasing rapidly and satellite data repositories will soon add one to two terabytes of data every day [1]. The general approach is to represent the data objects as multidimensional points and to measure the similarity between objects by the distance between the corresponding multidimensional points [12], [14], [29], [19], [37]. Many large data sets in scientific domains contain a large number of attributes that may be queried and analyzed and, therefore, considered as high-dimensional data. For example, High Energy Physics data typically contains more than 500 attributes that describe the properties of the objects in experiment data [36]. Since the dimensionality and the amount of data is large, it becomes important to support efficient high-dimensional searching in large-scale systems. To this end, a number of index structures for retrieval of multidimensional data along with associated algorithms for similarity searching have been developed [35], [31], [24], [3], [41], [30], [6], [5], [21], [22]. However, it is well-known that as dimensionality increases, query performance of these techniques degrades significantly [4]. This anomaly is referred as the dimensionality curse [20] and has attracted the attention of several researchers.

Reducing the dimensionality to enhance the performance of the underlying technique is a popular solution to the curse of dimensionality [15], [33], [28]. Evidently, there is a trade off between the accuracy obtained from the information stored and the efficiency obtained by the reduction. It is well-known that, if each data is represented by a smaller number of dimensions, significant performance speed-ups can be achieved, while part of the information is lost. The most common approaches found in the literature for dimensionality reduction are linear-algebraic methods such as the Karhunen-Loeve Transformation (KLT) [26], or applications of mathematical transforms such as the Discrete Fourier Transform (DFT) [34], Discrete Cosine Transform (DCT) [27], or Wavelet Transform (DWT) [9]. As the transformations are known to be distance preserving, the general approach is to transform the high-dimensional feature vectors and obtain lower-dimensional vectors by taking a small subset of dimensions which restore the highest energy [2], [42]. Several reduction techniques were proposed for time-series [2], [10], image [42], [33], [19], [28], and document data [15], [14], [16]. We and Chakrabarti and Mehrotra recently proposed an integration of dimensionality reduction with clustering [10], [22]. Random projections have been used recently for dimensionality reduction in image and text data [8], [13]. Theoretical results and experiments on noisy image data demonstrate the ability of this method in preserving the distances. And, finally, a nonlinear dimensionality reduction was proposed in [40]. If the distance between the transformed vectors is a lower

- *Ö. Egecioglu is with the Department of Computer Science, University of California, Santa Barbara, CA 93106. E-mail: omer@cs.ucsb.edu.*
- *H. Ferhatosmanoglu and U. Ogras are with the Department of Computer and Information Science, 395 Dreese Lab, 2015 Neil Ave., Ohio State University, Columbus, OH 43210.*
  E-mail: {hakan, ogras}@cis.ohio-state.edu.

Manuscript received 15 Mar. 2003; revised 30 Sept. 2003; accepted 9 Dec. 2003.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-0007-0303.

bound to the distance between the original feature vectors, then the lower-bound filtering property is said to hold [38]. Most of the current approaches focus on the lower-bound property even in the expense of approximation quality. However, in many applications, approximation quality is practically more important than the lower-bound property. Approximate query processing is such an example [28], [23], [22]. Approximate methods achieve efficiency at the expense of exact results especially for large-scale data sets. 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 [32]. Moreover, imprecise information will not only appear as the output of queries, it already appears in data sources as well [7]. In this paper, we first present a reduction technique that has the lower-bound filtering property. We then focus more on the approximation quality and improve the approximations significantly compared to the state-of-the-art techniques.

In particular, 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 high-dimensional objects can be accurately approximated by the lower-dimensional representations. More specifically, we focus on approximating the inner-product and, consequently, approximating the cosine of the angle between the two vectors in high dimensions. In some sense, the techniques presented here are the multidimensional analogues of the Cauchy-Schwarz inequality, which can be thought of as a first order approximation to the inner-product. In a recent work [11], Charikar discusses a sketching scheme for estimating the cosine similarity measure between two vectors. Apart from this, to the best of our knowledge, there is no other technique for approximation of similarity computation based on inner-products. 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 multidimensional representations created by statistical analysis, and their similarity are measured by the cosine of the angle between these vectors [39], [15], [14], [16].

The proposed techniques, unlike many others, can be efficiently adapted also for streaming data. In many recent applications, data is more conveniently modeled as streams rather than finite, stored databases. Examples include network monitoring, security, sensor networks, manufacturing, and financial analysis. Data streams are rapid, continuous, unbounded, and dynamic in nature. Furthermore, in data stream applications, both storage and computation resources are limited and random access to the data is not possible. Due to these challenges, most existing data mining algorithms cannot be utilized for data stream applications. However, our method is dynamic in nature and can be efficiently applied in data stream applications.

This paper is a significantly extended version of the earlier work which appeared in [18]. In [18], Egecioglu and Ferhatosmanoglu introduced inner-product approximation based on symmetric p-NORMS. Then, we used this result to reduce the dimensionality of data sets whose elements are drawn from a known probability distribution. Our main contributions in this paper can be summarized as follows:

1. We develop a dynamic technique to compute the best set of coefficients for unknown distributions and apply it to real data sets.
2. We evaluate the query performance of the technique using various real data sets.
3. We extend the discussion of computing the best set of coefficients to poisson, power, beta, exponential, and binomial distributions, and provide corresponding experimental results.
4. We discuss how to utilize our technique in data stream applications.

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 other distributions are given in Section 5. Theorem 1 is the major result of Section 5, which characterizes the optimal parameters in terms of the moments of the assumed density function. This result is then used to compute the optimal parameters. Section 6 covers the dynamic case. We show that it is possible to estimate the moments incrementally when the distribution is nonparametric. Section 7 presents comparisons with well-known methods such as SVD, DFT, and DCT. Conclusions and future work appear in Section 8.

2 Reduction with Power Symmetric Functions

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 of dimension n with reduced number of dimensions m with $m \ll n$. Then, we develop techniques for these representatives so that the similarity measure between high-dimensional vectors are approximated closely in the lower-dimensional space. We specifically focus on developing techniques which provide accurate approximations for the similarity distance between high-dimensional objects, which is important for similarity searching.

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

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

This is the pth power symmetric function in the variables $z = (z_1, z_2, \ldots, z_n)$. Equivalently, $\psi_p(z)$ is the pth 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, $\|z\|_2$ 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

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

where $<x,y>$ is the standard inner-product given by $<x,y>= x_1y_1 + x_2y_2 + \ldots + x_ny_n$.

In our method, we calculate $\psi_1(x), \psi_2(x), \ldots, \psi_m(x)$, and keep these m real numbers as a representative of each original high-dimensional vector x. In order to have the 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 (2), the approximated distance value computed in this way always becomes smaller than the original distance value due to the negative sign of $<x,y>$. The Cauchy-Schwarz inequality below provides an upper bound for the inner-product:

$$<x,y> \leq \|x\|_2\|y\|_2. \qquad (3)$$

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 the 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. However, this approximation does not minimize the error. As stated before, it is important to support approximate answers to get fast queries. Besides the time factor, we also want the answer to be as accurate as possible. We focus on the quality of the approximations, i.e., we aim to minimize the error made on the distance computations. We can try a correction with lower order terms with the hope of obtaining a better approximation to $<x,y>$ "most of the time"; in fact, we can try for a linear combination of the form $<x,y>^2 \approx b_1\psi_1(x)\psi_1(y) + b_2\psi_2(x)\psi_2(y)$ for some b_1, b_2. Now, we have lost the actual inequality (unless $b_1 = 0$, $b_2 = 1$), but hopefully the approximation is now better on the average if the b_1 and b_2 are chosen well. What "on the average" means in our treatment is best in the sense of least squares. In general, using the quantities for $\psi_p(z)$ computed for each data vector z in the database, we look for an approximation for $<x,y>$ by approximating its mth power in the form

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

for large n, where b_i is a constant chosen independently of x and y. Our assumption on the structure of the data vectors is as follows: We have a table of a large number of n-dimensional vectors $x = (x_1, x_2, \ldots, x_n)$ whose components are independently drawn from a common (possibly unknown) distribution $F(t)$ with density $f(t)$. In other words, each x_i is drawn independently of other coordinates from a probability 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. This is done by computing $\psi_1(y), \psi_2(y), \ldots, \psi_m(y)$ and then using the m stored quantities $\psi_1(x), \psi_2(x), \ldots, \psi_m(x)$ via (4). The coefficients $b_1, \ldots, b_m$ are fixed for a chosen m and do not depend on x or y.

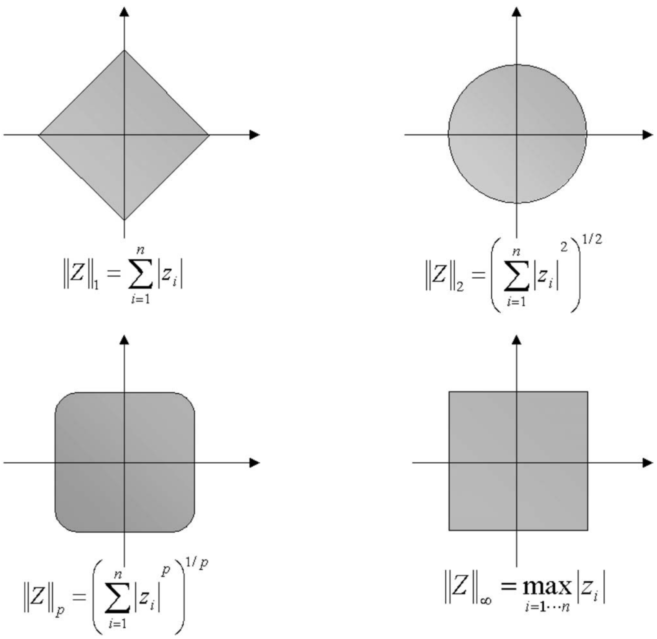

$\|Z\|_1 = \sum_{i=1}^{n} |z_i|$
$\|Z\|_2 = \left(\sum_{i=1}^{n} |z_i|^2\right)^{1/2}$
$\|Z\|_p = \left(\sum_{i=1}^{n} |z_i|^p\right)^{1/p}$
$\|Z\|_\infty = \max_{i=1\cdots n} |z_i|$

Fig. 1. The geometry of the unit disk in the plane for various norms: (a) $\|z\|_1 \leq 1$, (b) the Euclidean disk $\|z\|_2 \leq 1$, (c) $\|z\|_p \leq 1$ for $2 < p < \infty$, and (d) $\|z\|_\infty \leq 1$.

We consider approximations of the form (4) by finding *the best set of constants* $b_1, b_2, \ldots, b_m$ 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, we also have an overall gain on the computation time for similarity checking of large data sets besides efficiency gains in indexing. Note that, just as the ordinary 2-norm used in the Cauchy-Schwarz inequality, the quantities $\psi_p(z)$ used in (4) are also symmetric functions of the coordinates. A more general class of algorithms is obtained by taking instead $\psi_p(qz)$ in (4), where $qz = (q_1z_1, q_2z_2, \ldots, q_nz_n)$ 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 of x and y. For computational simplicity, we look at the symmetric case, in which $\psi_p(z)$ is as given in (1) and $z \in I^n$, where I^n is 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.

A secondary problem we address is dynamic in nature. We show that when the contents of the database change by adding new data vectors, for example, the parameters used for the approximation problem to the inner-product calculation can be adjusted efficiently.

Note that various distances defined from the p-norm in (1) result in different geometric interpretations and, consequently, different notions of distance. For example, the geometry of the unit disk is shown in Fig. 1 in the plane for various values of p. The case $p = 2$ is the usual Euclidean metric as defined in (2) in terms of the power symmetric functions. For $2 < p < \infty$, the resulting disk is squeezed between the ordinary unit circle and the unit square in the plane. Methods such as the SVD also have intuitive geometric interpretations. For example, the singular values of A (6) are the lengths of the principal axes of the ellipsoid which is the image of the unit disk $\|z\|_2 \leq 1$ under A. Our method is more algebraic in nature. We essentially compute the projection of the form $< x, y >^m$ onto the space defined by all linear combinations of $\psi_1(x)\psi_1(y), \ldots, \psi_m(x)\psi_m(y)$. We compute the best set of parameters $b_1, b_2, \ldots, b_m$ for an expansion of the form (4).

3 Determination of the Optimal Parameters

The best approximation in the least-squares sense minimizes

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

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 so-called *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. This results in an $m \times m$ linear system that $b_1, \ldots, b_m$ must satisfy

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

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

$$\begin{aligned} a_{i,j} &= \int \psi_j(x)\psi_j(y)\psi_i(x)\psi_i(y) dxdy, \\ c_i &= \int < x, y >^m \psi_i(x)\psi_i(y) dxdy, \end{aligned} \tag{6}$$

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

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 (4). The details of the proof of the general case can be found in [17]. For $m = 2$,

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

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

$$\int_{I^n} \psi_1(x)\psi_1(x) dx = \int_{I^n} \sum_{k=1}^{n} x_k \psi_1(x) dx = \sum_{k=1}^{n} \int_{I^n} x_k \psi_1(x) dx.$$

By symmetry, this last expression can be written as

$$\begin{aligned} n \int_{I^n} x_1 \psi_1(x) dx &= n \int_{I^n} x_1^2 dx + n(n-1) \int_{I^n} x_1 x_2 dx \\ &= n\left(\frac{1}{3}\right) + n(n-1)\left(\frac{1}{4}\right). \end{aligned}$$

Therefore,

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

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

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

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

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

There are two kinds of terms arising from the expansion of $\left(\sum x_k y_k\right)^2$. 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

$$\begin{aligned} n(n-1) &\int x_1 y_1 x_2 y_2 \psi_i(x)\psi_i(y) dxdy \\ &= n(n-1)\left(\int x_1 x_2 \psi_i(x) dx \right)^2 \\ &= n(n-1)\left(\frac{n-2}{8} + \frac{1}{6} + \frac{1}{6} \right)^2 \\ &= n(n-1)\left(\frac{3n+2}{24} \right)^2. \end{aligned}$$

$n(n-1)\left(\frac{3n+2}{24}\right)^2$. Therefore,

$$c_1 = n\left(\frac{2n+1}{12} \right)^2 + n(n-1)\left(\frac{3n+2}{24} \right)^2. \tag{7}$$

By a similar calculation, we find

$$c_2 = n\left(\frac{5n+4}{45}\right)^2 + n(n-1)\left(\frac{n+1}{12}\right)^2. \tag{8}$$

Therefore,

$$\begin{aligned} c_1 &= n\left(\frac{2n+1}{12}\right)^2 + n(n-1)\left(\frac{3n+2}{24}\right)^2 \\ c_2 &= n\left(\frac{5n+4}{45}\right)^2 + n(n-1)\left(\frac{n+1}{12}\right)^2. \end{aligned}$$

The resulting system satisfied by b_1, b_2 is

$$\begin{aligned} &n^2\left(\frac{3n+1}{12}\right)^2 b_1 + n^2\left(\frac{2n+1}{12}\right)^2 b_2 \\ &\quad = n\left(\frac{2n+1}{12}\right)^2 + n(n-1)\left(\frac{3n+2}{24}\right)^2 \\ &n^2\left(\frac{2n+1}{12}\right)^2 b_1 + n^2\left(\frac{5n+4}{45}\right)^2 b_2 \\ &\quad = n\left(\frac{5n+4}{45}\right)^2 + n(n-1)\left(\frac{n+1}{12}\right)^2. \end{aligned} \tag{9}$$

Since we are interested in these approximations for large n, it is tempting to let $n \to \infty$ in the resulting linear system and then solve for b_1, b_2 directly to obtain an asymptotic formula. 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

$$\begin{aligned} \left(\frac{n}{16} + \frac{1}{24}\right) b_1 + \left(\frac{n}{36} + \frac{1}{36}\right) b_2 &= \frac{n}{64} + \frac{19}{576} \\ \left(\frac{n}{36} + \frac{1}{36}\right) b_1 + \left(\frac{n}{81} + \frac{8}{405}\right) b_2 &= \frac{n}{144} + \frac{25}{1,296}, \end{aligned} \tag{10}$$

which is nonsingular for every n. Solving (10) symbolically for b_1 and b_2 and taking limits, we find

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

Therefore, the limiting optimal values are

$$b_1 = -\frac{1}{16}, \quad b_2 = \frac{45}{64}. \tag{11}$$

This means that for $m = 2$, we approximate $< x, y >$ by the expression

$$\sqrt{\left| -\frac{1}{16}\psi_1(x)\psi_1(y) + \frac{45}{64}\psi_2(x)\psi_2(y) \right|}. \tag{12}$$

3.1 Uniform Distribution: Arbitrary m

For general m, it can be shown [17] that

$$\begin{aligned} a_{i,j} &= \int_{I^{2n}} \psi_i(x)\psi_i(y)\psi_j(x)\psi_j(y)\,dx\,dy \\ &= \frac{n^2(ij + n(i+j+1))^2}{(i+1)^2(j+1)^2(i+j+1)^2}. \end{aligned}$$

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

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

This matrix

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

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 [17]. We omit the details of the derivation of the optimal coefficients $b_1, b_2, \ldots, b_m$ for $m > 2$. For the uniform distribution coefficients with $m = 2$, the approximation (12) we obtained

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

does not involve the dimension n. This is not the case for $m > 2$. 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 $m = 4$

$$\begin{aligned} < x, y >^4 \approx &-\frac{59}{256} n^2\psi_1(x)\psi_1(x) + \frac{1,575}{1,024} n^2\psi_2(x)\psi_2(y) \\ &- \frac{175}{64} n^2\psi_3(x)\psi_3(y) + \frac{1,575}{1,024} n^2\psi_4(x)\psi_4(y). \end{aligned}$$

Values of $b_1, \ldots, b_m$ we have computed for various values of m for the uniform distribution appear in Fig. 2.

4 Accuracy of Inner-Product Approximation

In the first set of experiments, we analyze the accuracy of our approximation techniques 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. Besides this, the accuracy of this approximation directly affects the quality of the similarity distance approximation in Euclidean spaces as mentioned before.

First, consider the case $m = 2$ and the approximation given by (12). The graph of the average absolute error made appears in Fig. 3. 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

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	$-\frac{535}{65536}n^6$	$\frac{43659}{262144}n^6$	$-\frac{43659}{32768}n^6$	$\frac{2837835}{524288}n^6$	$-\frac{3972969}{327680}n^6$	$\frac{3972969}{262144}n^6$	$-\frac{81081}{8192}n^6$	$\frac{1378377}{524288}n^6$

Fig. 2. $< x, y >^m \approx b_1\psi_1(x)\psi_1(y) + \cdots + b_m\psi_m(x)\psi_m(y)$: asymptotic expansion coefficients $b_1, b_2, \ldots, b_m$ for the uniform distribution.

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

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

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.

Fig. 4 shows the average absolute error versus dimension for the reduced dimension $m = 2, 4, 6, 8$, and original dimension n ranging from 2 to 2,048.

5 Optimal b_1, b_2 for Various Parametric Distributions

Suppose now that the coordinates of the vectors x and y are not drawn from 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,

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

The ith 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

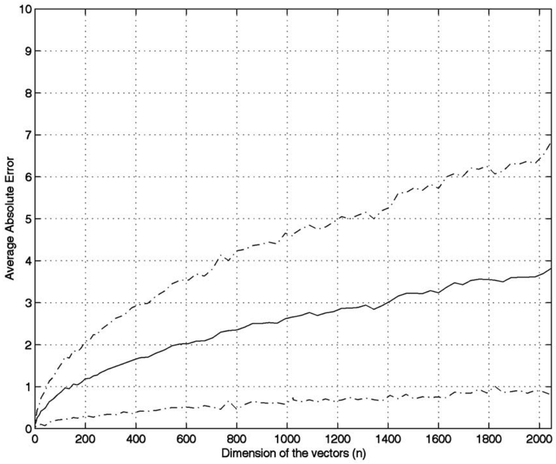


Fig. 3. Average absolute error versus dimension n, $2 \leq n \leq 2{,}048$ for vectors from the uniform distribution with $m = 2$. Average error plus minus standard deviation is also shown by dotted lines.

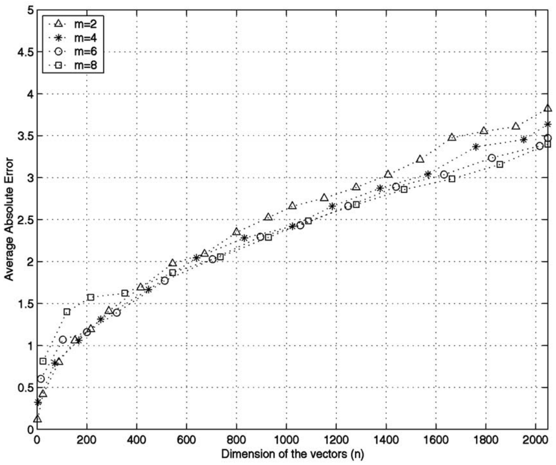


Fig. 4. Average absolute error versus dimension n, $2 \leq n \leq 2{,}048$ for vectors from the uniform distribution with $m = 2, 4, 6, 8$. It can be observed that increasing m decreases average relative error.

$$a_{i,j} = \int \psi_j(x)\psi_j(y)\psi_i(x)\psi_i(y)dF(x)dF(y), \tag{13}$$

$$c_i = \int < x, y >^m \psi_i(x)\psi_i(y)dF(x)dF(y). \tag{14}$$

Lemma 1. *Suppose $a_{i,j}$ is as given in (13). 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 (14). Then,*

$$\begin{aligned}c_1 &= [(n-1)\mu_1\mu_2 + \mu_3]^2 + n(n-1)\left[(n-2)\mu_1^3 + 2\mu_1\mu_2\right]^2\\ c_2 &= n\left[(n-1)\mu_2^2 + \mu_4\right]^2 + n(n-1)\left[(n-2)\mu_1^2\mu_2 + 2\mu_1\mu_3\right]^2.\end{aligned} \tag{15}$$

Proof. Omitted. □

These expressions for c_1, c_2 are a special case of a more general theorem that appears in [17]. For $m = 2$, using the expressions in (15) 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, whose proof can be found in [17].

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 formulae

$$\begin{aligned}b_1 &= \mu_1^2 \cdot \frac{2\mu_2^3 + \mu_1^2\mu_4 - 3\mu_1\mu_2\mu_3}{\mu_2^3 - \mu_1^2\mu_4 - 2\mu_1\mu_2\mu_3},\\ b_2 &= \frac{\mu_1^4}{\mu_2} \cdot \frac{\mu_1\mu_3 - \mu_2^2}{\mu_2^3 - \mu_1^2\mu_4 - 2\mu_1\mu_2\mu_3}.\end{aligned} \tag{16}$$

Suppose now that 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, 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 (16) 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, [25]). Below, we work out a number of examples. It is interesting to note that the expressions for the optimal constants b_1 and b_2 both have μ_1 as a multiplicative factor. Therefore, if the mean of the distribution of the coordinates is zero, then, for the approximation with $m = 2$, the algorithm gives $< x, y > \approx 0$. In this case, the distance between x and y is approximated by

$$\|x - y\|_2 \approx \sqrt{\psi_2(x) + \psi_2(y)}$$

in accordance with the identity (2).

5.1 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 ith moment of $f(x)$ (around the origin) is given by $\mu_i = c/(c+i)$. From Theorem 1, we get

$$\begin{aligned}b_1 &= -\frac{2c^3}{(c+1)^2(c^2+3c+4)},\\ b_2 &= \frac{c^2(c+2)^2(c+4)}{(c+1)^2(c^2+3c+4)}.\end{aligned} \tag{17}$$

For $c = 1$, $f(x) = 1$ on $0 \le x \le 1$ and the distribution is uniform. In this case, the formulas in (17) 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 (11).

5.2 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 ith moment of $f(x)$ (around the origin) is $\mu_i = i!\, b^i$. From Theorem 1, we get

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

5.3 Binomial Distribution

Let $0 \le x \le N$ 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	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	$\binom{N}{x}p^x q^{N-x}$	$0 \leq x \leq N$	$\frac{N^2p^2(1-2p)}{np-3p+2}$	$\frac{N^2p^2}{(np-p+1)(np-3p+2}$
Normal	$\frac{1}{\sigma\sqrt{2\pi}}\exp(\frac{-(x-\mu)^2}{2\sigma^2})$	$-\infty \leq x \leq \infty$	$\frac{2\mu^2\sigma^4}{\mu^4+\sigma^4}$	$\frac{\mu^4(\mu^2-\sigma^2)}{(\mu^2+\sigma^2)(\mu^4+\sigma^4)}$
Poisson	$\lambda^x \exp(-\lambda)/x!$	$0 \leq x \leq \infty$	$\frac{\lambda^2}{\lambda+2}$	$\frac{\lambda^2}{(\lambda+2)(\lambda+1)}$
Beta	$\frac{(v+w-1)!x^{v-1}(1-x)^{w-1}}{(v-1)!(w-1)!}$	$0 \leq x \leq 1$	$\frac{2v^2(w-v-1)}{(v+w)^2((v+1)^2+(v+3)w)}$	$\frac{v^2(w+v+1)^2(w+v+3)}{(v+w)^2((v+1)^3+(v+1)(v+3)w)}$

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

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 $f(x)$ (around the origin) are given by

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

From Theorem 1, we get

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

5.4 Normal Distribution

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

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

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

$$\begin{aligned}
&\mu_1 = \mu, \quad \mu_2 = \mu^2 + \sigma^2, \\
&\mu_3 = \mu^3 + 3\mu\sigma^2, \quad \mu_4 = \mu^4 + 6\mu^2\sigma^2 + 3\sigma^4.
\end{aligned}$$

From Theorem 1, we get

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

5.5 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

$$\begin{aligned}
&\mu_1 = \lambda, \quad \mu_2 = \lambda(\lambda+1), \\
&\mu_3 = \lambda(\lambda^2 + 3\lambda + 1), \quad \mu_4 = \lambda(\lambda^3 + 6\lambda^2 + 7\lambda + 1).
\end{aligned}$$

From Theorem 1, we get

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

5.6 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 ith 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

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

A summary of these calculations for power, exponential, binomial, normal, Poisson, and Beta distributions appears in Fig. 5. The last two columns are the optimal values of b_1 and b_2 expressed in terms of the parameters of the corresponding distribution.

6 Nonparametric Case: Estimating the Moments for an Unknown Density

Data streams are rapid, continuous, unbounded, and dynamic in nature. Hence, the best set of coefficients $b_1, b_2, \ldots, b_m$ may vary with time. Consequently, we need a technique to dynamically compute best coefficients for nonparametric distributions. In the following section, we show how to estimate the moments of an unknown density and use them to compute $b_1, b_2, \ldots, b_m$. Note that this result can be utilized both for static databases with unknown densities and data streams. If the psi values grow indefinitely for the data stream case, one can reset them to zero and restart the approximation process.

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. Now, 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 $0 \leq t \leq 1$.

Suppose we know the empirical moments $\bar{\mu_i} = \bar{\mu_i}(N)$ of density $f(t)$, $0 \leq t \leq 1$, based on samples $t_1, t_2, \ldots, t_N$. 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*

$$\bar{\mu_i}(N) = \frac{1}{N(i+1)}\left(N - (t_1^{i+1} + t_2^{i+1} + \cdots + t_N^{i+1})\right).$$

Proof. An estimate $f_N(t)$ for the density given the samples $t_1, t_2, \ldots, t_N$ is the histogram

$$f_N(t) = \frac{1}{N}|\{t_j | t_j < t\}|,$$

where the bars denote cardinality. Therefore,

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

where $t_{N+1} = 1$. This sum simplifies to

$$\bar{\mu_i}(N) = \frac{1}{N(i+1)}\left(-t_1^{i+1} - t_2^{i+1} - \cdots - t_N^{i+1} + N\right).$$

□

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

$$\bar{\mu_i}(N+1) = \frac{N}{N+1}\bar{\mu_i}(N) + \frac{1 - t_{N+1}^{i+1}}{(N+1)(i+1)}.$$

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 ith 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

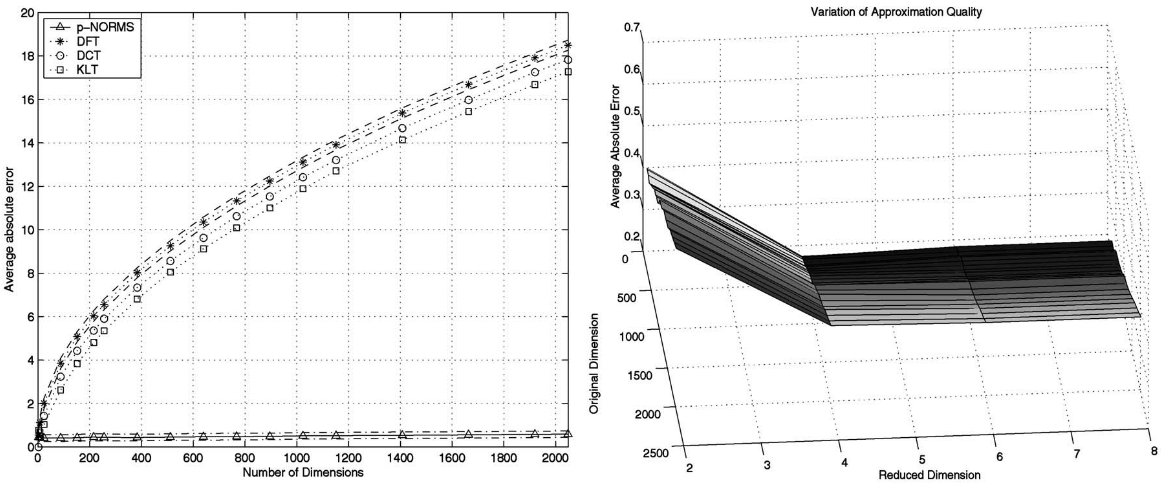
$$\bar{\mu_i}[r+1] = \frac{r}{r+1}\bar{\mu_i}[r] + \frac{n - \psi_{i+1}(x)}{n(r+1)(i+1)}. \qquad (18)$$

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

7 Performance Evaluation

The techniques presented in this paper can be readily used for the approximation of the similarity both with respect to inner product and 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 $< x, y >$ to find the distance between two feature vectors. By using the stored ψ values we approximate $< x, y >$ and, hence, the original distance. Next, we compare the performance of our technique 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. Since the reduced dimensional vectors are representatives of original high-dimensional vectors, we approximate the real distance by computing the distance between each pair of vectors of smaller dimensions. For each technique, we compute the *absolute error*, i.e., the difference between the approximate and real distances, 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. 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. In the figures, we refer to our approximation as *p-NORMS*. First, we reduce the number of dimensions to $m = 2$ using p-NORMS. Due to the symmetry property of the DFT, stated as $X(k) = X^*(-k)$, the knowledge of $X(2)$ reveals $X(N-1)$, where $X(k)$ is the DFT of the original data. For this reason, we reduce the dimensionality of the other techniques to 3. For each technique, the average absolute approximation error is computed over all pairs of points (25,000). This average error gives the quality of the approximations achieved by each technique. Even when the other techni-



(a)

(b)

Fig. 6. Error comparisons of dimensionality reduction techniques (for $m = 2$ and higher dimensions). The standard deviations (STD) for p-NORMS and DFT are also plotted. The STDs of KLT and DCT are not shown for the clarity of the figures since they produced almost identical results. (a) illustrates the variation of the error for increasing m and (b) results when $m = 4$.

ques use three coefficients, in this case their approximation quality appears much worse than our technique.

We repeated the experiments by varying the number of dimensions n from 2 to 2,048 and analyzing the resulting approximations. Fig. 6a 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 200-dimensional data, for instance, the new technique's approximation is 10.2 times better than the current best approach. For 2,048 dimensional data, the average absolute error of p-NORMS is 0.6 and the average absolute error of the SVD technique, the best of the three is 17.2. Similar experiments with $m = 4$ for all techniques were also performed. Fig. 6b illustrates the results of these.

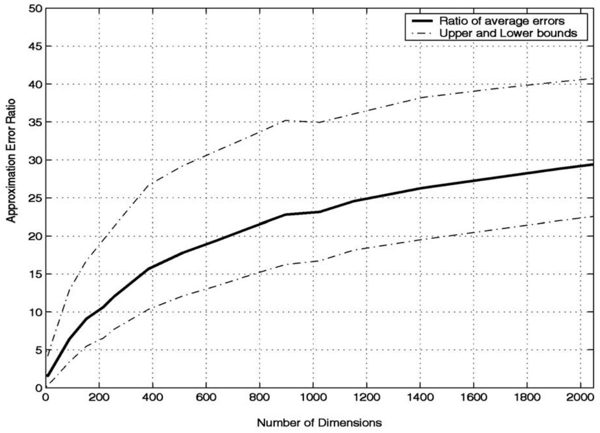
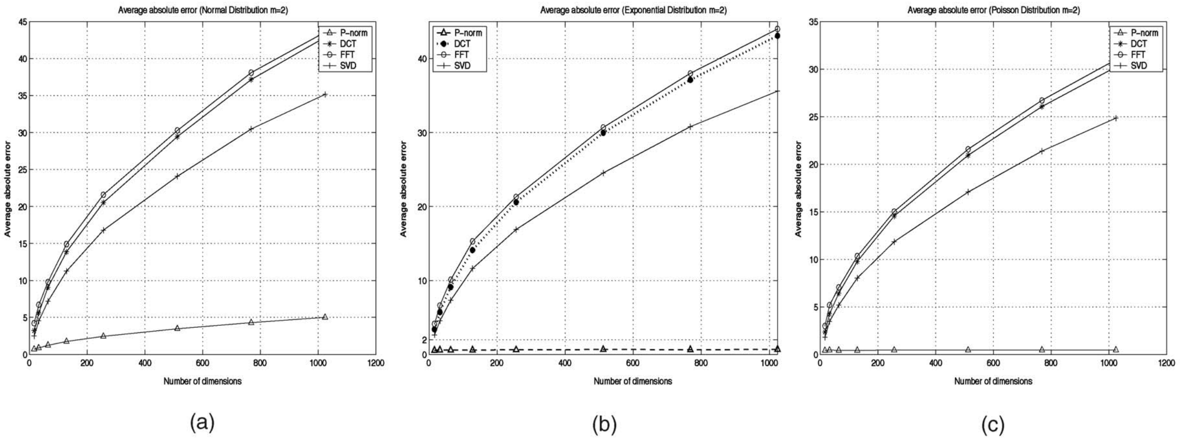


Fig. 7. Scalability comparison of SVD and p-NORMS.

We also compute the approximation quality ratio of our technique with SVD as dimensionality increases in order to illustrate the scalability of our approach. Fig. 7 illustrates the superiority of p-NORMS over SVD as a function of dimensionality.

We also analyzed the quality of the approximations developed for data sets where the components are drawn from a *normal*, *exponential*, and *poisson* distributions. We generated 500 random points from a normal distribution with mean 0 and variance 1. We note that since the data is not restricted to be within the range $[0 \ldots 1]$ as 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. We computed the average absolute error as in the previous case. Approximations based on p-NORMS gave an error 12 times lower than the best of the three other techniques, in this case SVD. Fig. 8a illustrates the results of these experiments. For exponential distribution, $f(x) = (1/b)\exp(-x/b)$, we selected the parameter $b = 1$ and followed the same steps. Finally, for poisson distribution, $f(x) = \lambda^x \exp(-\lambda)/x!$, we set $\lambda = 0.5$. We obtained similar results as shown in the graphs of Figs. 8b and 8c, respectively.

The techniques were also compared on real data sets. The first data set is the stock market data [43] which is a time-series containing 256 days stock price movements of 2,000 companies, i.e., 2,000 data points with dimensionality 256. We reduce the number of dimensions to $m = 2$ using (18) and Theorem 1. 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 six times as well as SVD and 6.2 times better than DCT. We note that SVD performs better than DCT on real data as well. We also found the nearest 100 neighbors for each vector in the data set using the methods mentioned above. Then, we compared them with the actual results and computed the number of false



(a) (b) (c)

Fig. 8. Comparison of average absolute errors for (a) Normal distribution $\eta(0,1)$, (b) Exponential distribution ($b = 1$), and (c) Poison distribution ($\lambda = 0.5$).

hits. Finally, we took the average of the number of false hits over the data set. Similar to previous experiments, the p-NORMS has considerably higher performance than the other methods. The average number of false hits obtained with p-NORMS is 22.05, which is almost one third of the result of other methods. Fig. 9 shows the average absolute errors and average number of false hits for each technique. We performed additional experiments, where the number of nearest neighbors is varied from 10 to 500 and observed the number of false hits. We repeated this experiment for 50 different query points and averaged the results. Fig. 10 shows that the quality of the result increases with the increasing query size. Furthermore, p-NORMS performs better than KLT like the previous experiments. For the clarity of the figure, other methods which produced weaker results are not shown. Further experiments are performed with other real data sets. The average absolute errors obtained for wireless telephony data, which stores the data received by 64 stations over 1,000 periods, is shown in Fig. 11. It can be observed that the p-NORMS results in lower error compared to the other methods. Furthermore, the average error plus its standard deviation is always smaller than the average errors minus standard deviations of other methods. p-NORMS has an average absolute error of 262.5+-78.5, KLT has an error of 865+ -240.2. With respect to the average number of false hits, p-NORMS achieves consistently better results and it achieves on the average 7 to 10 less false hits than other techniques. We also performed experiments where the query is not decomposed into the same number of coefficients as the summary. Using all of the coefficients of the query improves the approximation of errors about 50 percent for the DCT and DFT, and 40 percent for KLT. However, the number of false hits was not affected since the relative distances of the data points remains the same.

Metric	p-NORMS	DCT	FFT	KLT
Average absolute error	48.4	300.9	391.7	292.5
Average number of false hits	22.05	61.45	73.43	59.48

Fig. 9. Comparison of the dimensionality reduction methods with respect to average absolute error and average number of false hits using real data.

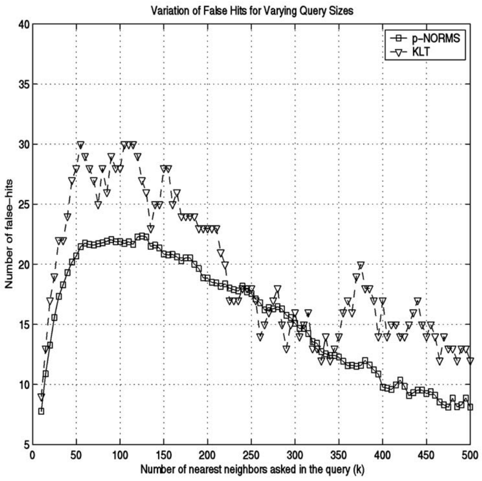


Fig. 10. The variation of number of false hits for different query sizes for the stock market data. Number of neighbors asked is varied from 10 to 500.

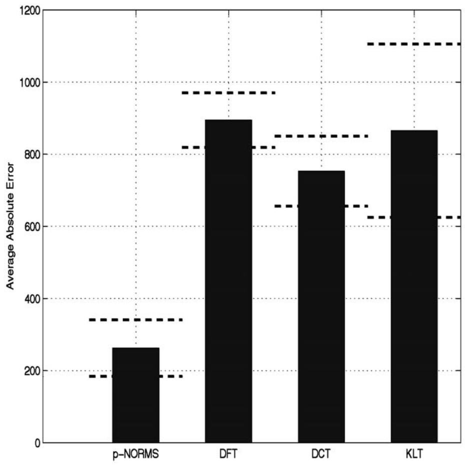


Fig. 11. Average absolute errors and standard deviations for wireless telephony data.

Metric	p-NORMS	DCT	FFT	KLT
Preprocessing time (sec)	0.0510	0.29	1.062	3.455

Fig. 12. The preprocessing times for the methods.

We also compared the preprocessing times of the methods mentioned in the experiments. We computed the ψ values for p-NORMS approximation and the DFT, DCT, and KLT of stock market data of size $360 \times 1,000$ using MATLAB©. The results are summarized in Fig. 12. p-NORMS has a preprocessing time 5.7 times lower than the DFT, which is considerably faster than the other methods. Particularly, p-NORMS is 67.7 times faster than KLT, which has the closest performance to it in terms of approximation quality and number of false hits.

8 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 pth 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

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

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. Since p-norm reduction directly targets to minimize the error made in the approximations, it achieves consistently better performance than the well-known methods such as the KLT (SVD), DFT, and DCT. The approximation error is better than well-known methods as verified by experiments on synthetic and real data sets.

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 that can be pursued. 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 $(q_1, q_2, \dots, q_n)$, and hybrid approaches which can take advantage of various methods currently available for dimensionality reduction.

Acknowledgments

The authors would like to acknowledge the constructive comments made by the reviewers, which have improved the presentation and the scope of our method significantly. This work was partially supported by DOE Career Award DE-FG02-03ER25573.

References

[1] A. Acharya, M. Uysal, and J. Saltz, "Active Disks: Programming Model, Algorithms and Evaluation," *Proc. Eighth Int'l Conf. Architectural Support for Programming Languages and Operating Systems,* pp. 81-91, May 1998.

[2] R. Agrawal, C. Faloutsos, and A. Swami, "Efficient Similarity Search in Sequence Databases," *Proc. Fourth Int'l Conf. Foundations of Data Organization and Algorithms,* pp. 69-84, 1993.

[3] N. Beckmann, H. Kriegel, R. Schneider, and B. Seeger, "The R*Tree: An Efficient and Robust Access Method for Points and Rectangles," *Proc. ACM SIGMOD Int'l Conf. Management of Data,* pp. 322-331, May 1990.

[4] S. Berchtold, C. Bohm, D. Keim, and H. Kriegel, "A Cost Model for Nearest Neighbor Search in High-Dimensional Data Space," *Proc. ACM Symp. Principles of Database Systems,* 1997.

[5] S. Berchtold, C. Bohm, and H.-P. Kriegel, "The Pyramid-Technique: Towards Breaking the Curse of Dimensionality," *Proc. ACM SIGMOD Int'l Conf. Management of Data,* pp. 142-153, June 1998.

[6] S. Berchtold, D. Keim, and H. Kriegel, "The X-Tree: An Index Structure for High-Dimensional Data," *Proc. Int'l Conf. Very Large Data Bases,* pp. 28-39, 1996.

[7] 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," *Sigmod Record,* vol. 27, no. 4, Dec. 1998.

[8] E. Bingham, H. Mannila, "Random Projection in Dimensionality Reduction: Applications to Image and Text Data," *Proc. Int'l Conf. Knowledge Discovery and Data Mining,* 2001.

[9] K.R. Castleman, *Digital Image Processing.* Prentice-Hall 1996.

[10] K. Chakrabarti and S. Mehrotra, "Local Dimensionality Reduction: A New Approach to Indexing High Dimensional Spaces," *The VLDB J.,* pp. 89-100, 2000.

[11] M.S. Charikar, "Similarity Estimation Techniques from Rounding Algorithms," *Proc. 34th Ann. ACM Symp. Theory of Computing,* 2002.

[12] 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," *IEEE Proc. Int'l Conf. Advances in Digital Libraries (ADL),* pp. 70-81, 1997.

[13] S. Dasgupta and A. Gupta, "An Elementary Proof of the Johnson-Lindenstrauss Lemma," Technical Report TR-99-006, Int'l Computer Science Inst., Berkeley, 1999.

[14] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Launder, and R. Harshman, "Indexing by Latent Semantic Analysis," *J. Am. Soc. for Information Science,* vol. 41, pp. 391-407, 1990.

[15] D. Hull, "Improving Text Retrieval for the Routing Problem Using Latent Semantic Indexing," *Proc. 17th ACM-SIGIR Conf.,* pp. 282-291, 1994.

[16] S.T. Dumais, "Improving the Retrieval of Information from External Sources," *Behavior Research Methods, Instruments and Computers,* vol. 23, pp. 229-236, 1991.

[17] Ö. Egecioglu, "How to Approximate the Inner-Product: Fast Dynamic Algorithms for Similarity," Technical Report TRCS98-37, Dept. of Computer Science, Univ. of California at Santa Barbara, Dec. 1998.

[18] Ö. Egecioglu and H. Ferhatosmanoglu, "Dimensionality Reduction and Similarity Distance Computation by Inner Product Approximations," *Proc. Ninth ACM Int'l Conf. Information and Knowledge Management,* pp. 219-226, Nov. 2000.

[19] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz, "Efficient and Effective Querying by Image Content," *J. Intelligent Information Systems,* vol. 3, pp. 231-262, 1994.

[20] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast Subsequence Matching in Time-Series Databases," *Proc. ACM SIGMOD Int'l Conf. Management of Data*, pp. 419-429, May 1994.

[21] H. Ferhatosmanoglu, E. Tuncel, D. Agrawal, and A. El Abbadi, "Vector Approximation Based Indexing for Non-Uniform High Dimensional Data Sets," *Proc. Ninth ACM Int'l Conf. Information and Knowledge Management*, pp. 202-209, Nov. 2000.

[22] H. Ferhatosmanoglu, E. Tuncel, D. Agrawal, and A. El Abbadi, "Approximate Nearest Neighbor Searching in Multimedia Databases," *Proc. 17th IEEE Int'l Conf. Data Eng. (ICDE)*, pp. 503-511, Apr. 2001.

[23] A. Gionis, P. Indyk, and R. Motwani, "Similarity Searching in High Dimensions via Hashing," *Proc. Int'l Conf. Very Large Data Bases*, pp. 518-529, Sept. 1999.

[24] A. Guttman, "R-Trees: A Dynamic Index Structure for Spatial Searching," *Proc. ACM SIGMOD Int'l Conf. Management of Data*, pp. 47-57, 1984.

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

[26] N.S. Jayant and P. Noll, *Digital Coding of Waveforms.* Prentice-Hall, 1984.

[27] T. Kailath, *Modern Signal Processing.* Springer Verlag, 1985.

[28] K.V.R. Kanth, D. Agrawal, and A. Singh, "Dimensionality Reduction for Similarity Searching in Dynamic Databases," *Proc. ACM SIGMOD Int'l Conf. Management of Data*, 1998.

[29] F. Korn, N. Sidiropoulos, C. Faloutsos, E. Siegel, and Z. Protopapas, "Fast Nearest Neighbor Search in Medical Image Databases," *Proc. Int'l Conf. Very Large Data Bases*, pp. 215-226, 1996.

[30] K. Lin, H.V. Jagadish, and C. Faloutsos, "The TV-Tree: An Index Structure for High-Dimensional Data," *VLDB J.*, vol. 3, pp. 517-542, 1995.

[31] D.B. Lomet and B. Salzberg, "The hb-Tree: A Multi-Attribute Indexing Method with Good Guaranteed Performance," *ACM Trans. Database Systems*, vol. 15, no. 4, pp. 625-658, Dec. 1990.

[32] B.S. Manjunath and W.Y. Ma, "Texture Features for Browsing and Retrieval of Image Data," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 837-42, Aug. 1996.

[33] 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," *Proc. SPIE Conf. 1908 on Storage and Retrieval for Image and Video Databases*, vol. 1908, pp. 173-187, Feb. 1993.

[34] A.V. Oppenheim and R.W. Schafer, *Discrete-Time Signal Processing.* Prentice-Hall, 1989.

[35] J.T. Robinson, "The kdb-Tree: A Search Structure for Large Multi-Dimensional Dynamic Indexes," *Proc. ACM SIGMOD Int'l Conf. Management of Data*, pp. 10-18, 1981.

[36] SciDAC, Scientific data management center, http://sdm.lbl.gov/sdmcenter/, 2002.

[37] T. Seidl and H.-P. Kriegel, "Efficient User-Adaptable Similarity Search in Large Multimedia Databases," *Proc. Int'l Conf. Very Large Data Bases*, pp. 506-515, 1997.

[38] T. Seidl and H.P. Kriegel, "Optimal Multistep k-Nearest Neighbor Search," *Proc. ACM SIGMOD Int'l Conf. Management of Data*, June 1998.

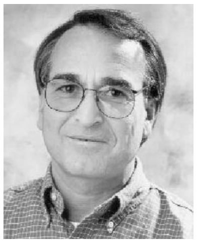
[39] V.S. Subrahmanian, *Principles of Multimedia Database Systems.* Morgan Kaufmann Publishers, 1999.

[40] M. Vlachos, C. Domeniconi, D. Gunopulos, G. Kollios, and N. Koudas, "Non-Linear Dimensionality Reduction Techniques for Classification and Visualization," *Proc. Eighth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, July 2002.

[41] D. White and R. Jain, "Similarity Indexing with the SS-Tree," *Proc. Int'l Conf. Data Eng.*, pp. 516-523, 1996.

[42] D. Wu, D. Agrawal, A. El Abbadi, and T.R. Smith, "Efficient Retrieval for Browsing Large Image Databases," *Proc. Conf. Information and Knowledge Management*, pp. 11-18, Nov. 1996.

[43] Y. Wu, D. Agrawal, and A. El Abbadi, "A Comparison of DFT and DWT Based Similarity Search in Time-Series Databases," *Proc. Ninth Int'l Conf. Information and Knowledge Management*, 2000.



Ömer Egecioglu received the PhD degree in mathematics from the University of California, San Diego, in 1984. At present, he is a professor in the Computer Science Department at the University of California, Santa Barbara, where he has been on the faculty since 1985. His principal areas of research are algorithms, bijective and enumerative combinatorics, and combinatorial algorithms. His current interest in parallel algorithms involves approximation and numerical techniques on distributed memory systems while his combinatorial interests center around computational geometry, algorithms on strings, bijective methods, and ranking algorithms for combinatorial structures.

Hakan Ferhatosmanoglu received the PhD degree in 2001 from the Computer Science Department at the University of California, Santa Barbara. He is an assistant professor of computer and information science at The Ohio State University (OSU). Before joining OSU, he worked as an intern at AT&T Research Labs. During his PhD studies, he proposed several techniques for efficient retrieval and scalable storage of large-scale multidimensional data. His current research interest is to develop data management systems for modern applications including biological, multimedia, and scientific databases, and sensor networks. He leads projects on microarray and clinical trial databases, online compression and analysis of multiple data streams, and high performance databases for multidimensional data repositories. Dr. Ferhatosmanoglu is a recipient of the Early Career Principal Investigator award from the Department of Energy. He is a member of the IEEE.

Umit Ogras received the BS degree in electrical engineering from Middle East Technical University, Turkey, in 2000 and the MS degree in electrical engineering from The Ohio State University, Columbus, in 2002. Since 2002, he has been a graduate research assistant in the Database Group The Ohio State University. His research interests include online compression, analysis of data streams, and dimensionality reduction. He is a student member of the IEEE.