# Parametric Approximation Algorithms for High-Dimensional Euclidean Similarity

Ömer Eğecioğlu*

Department of Computer Science,
University of California, Santa Barbara, CA 93106 USA
omer@cs.ucsb.edu

**Abstract**

We introduce a spectrum of algorithms for measuring the similarity of high-dimensional vectors in Euclidean space. The algorithms proposed consist of a convex combination of two measures: one which contains summary data about the *shape* of a vector, and the other about the relative *magnitudes* of the coordinates. We present experiments on time-series data on labor statistics unemployment figures that show the effectiveness of the algorithm as a function of the parameter that combines the two parts.

## 1 Introduction

Modern databases and applications use multiple types of digital data, such as documents, images, audio, video, etc. The general approach is to represent the data objects as multi-dimensional points in Euclidean space, and to measure the similarity between objects by the distance between the corresponding multi-dimensional points [8, 3]. To this end, a number of index structures for retrieval of multi-dimensional data along with associated algorithms for similarity search have been developed [6, 10, 2]. Common approaches for dimensionality reduction are linear-algebraic methods such as the singular value decomposition, or truncated discrete versions of various mathematical transforms [1].

This paper introduces a spectrum of similarity algorithms which consist of a convex combination of a *shape* measure on high-dimensional vectors and a symmetric *magnitude* measure.

## 2 The Main Decomposition

An $n$-dimensional real vector $x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ can be decomposed as a pair $(s(x), \pi(x))$ where $s(x)$ is the sorted version of $x$ into weakly increasing coordinates, and $\pi(x)$ is the permutation of the indices $\{1, 2, \ldots, n\}$ that achieves

*Supported in part by NSF Grant No. CCR–9821038.

this ordering. We impose the additional condition that the elements of the permutation $\pi(x)$ are put in increasing order on any set of indices for which the value of the coordinate is constant. For example when $x = (3, 3, 1, 5, 2, 0, 1, 6, 1)$, $s(x) = (0, 1, 1, 1, 2, 3, 3, 5, 6)$, and in one line notation, $\pi(x) = 6\ 3\ 7\ 9\ 5\ 1\ 2\ 4\ 8$. Note that in $x$ the smallest coordinate value is $x_6 = 0$, the next smallest is $x_3 = x_7 = x_9 = 1$, etc. Given $x, y \in \mathbb{R}^n$, we aim to approximate the Euclidean distance $\|x - y\|$ as a convex combination

$$\lambda s(x, y) + (1 - \lambda)\pi(x, y)\ , \quad \text{where} \tag{1}$$

- $s(x, y)$ is a measure of distance between $s(x)$ and $y(x)$ which is a symmetric function of the coordinates separately in $x$ and $y$ (we refer to this as the *magnitude* or the *symmetric* part),
- $\pi(x, y)$ is a measure of the distance between the permutations $\pi(x)$ and $\pi(y)$ (we refer to this as the *shape* part),
- $0 \leq \lambda \leq 1$ is a parameter that controls the bias of the algorithm towards magnitude/symmetry versus shape.

In order for such a scheme to be useful, the individual functions $s(x, y)$ and $\pi(x, y)$ must be computed using data with reduced dimensionality $\ll n$. In the technique proposed here, this reduced dimension can be selected separately for the two parts. First we discuss the construction of these parts and then present the results of the experiments.

# 3 The *magnitude* part: power symmetric functions

Our representation of data in $\mathbb{R}^n$ with reduced number of dimensions $m$ with $m \ll n$ for the computation of the magnitude part $s(x, y)$ in (1) is based on a novel approximation for the inner product introduced in [4] and developed in [5]. For integers $n, p > 0$ and $z \in \mathbb{R}^n$, the $p$-th power symmetric function is defined by $\psi_p(z) = z_1^p + z_2^p + \cdots + z_n^p$. Note that the ordinary Euclidean distance between $x$ and $y$ and the power symmetric functions are related by

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

where $< x, y >= x_1 y_1 + x_2 y_2 + \ldots + x_n y_n$ is the standard inner-product. Using the $\psi_p(z)$ precomputed for each vector $z$ in the dataset, we look for an estimate for $< x, y >$ by approximating its $m$-th power in the form

$$< x, y >^m \ \approx \ b_1\psi_1(x)\psi_1(y) + b_2\psi_2(x)\psi_2(y) + \cdots + b_m\psi_m(x)\psi_m(y) \tag{3}$$

for large $n$, where the $b_i$ are constants chosen independently of $x$ and $y$. For each high-dimensional vector $x$, we calculate $\psi_1(x), \psi_2(x), \ldots, \psi_m(x)$, and keep these $m$ real numbers as a representative of the original vector $x$. For a given query vector $y$, we compute $\psi_1(y), \psi_2(y), \ldots, \psi_m(y)$ and approximate $< x, y >$ via (3), and the Euclidean distance via (2).

Our assumption on the structure of the dataset for the computation of $s(x,y)$ by this method is as follows: it consists of $n$-dimensional vectors whose components are independently drawn from a common (but possibly unknown) distribution with density [7]. In [4] the best set of constants $b_1, b_2, \ldots, b_m$ for the approximation (3) in the sense of least-squares was computed. In particular for the uniform distribution and $m = 2$ the optimal values are $b_1 = -\frac{1}{16}$, $b_2 = \frac{45}{64}$. In fact in the general case of a density with $i$-th moment $\mu_i$ (about the origin) it can be proved [4] that

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

The moments of the uniform distribution are $\mu_i = 1/(i+1)$, for which the formulas in (4) reduce to the values above. We remark that one *need not* know the density of the distribution of the coordinates parametrically. The moments $u_i$ can be estimated as the limit of the $N$-th estimate $\bar{\mu_i}(N)$ as the dataset is accumulated via

$$\bar{\mu_i}(N+1) = \frac{1}{N+1}\left(N\bar{\mu_i}(N) + t_{N+1}^i\right). \tag{5}$$

where $t_N$ is the $N$-th sample coordinate observed.

# 4 The *shape* part: bin-score permutations

For a permutation $\sigma = \sigma_1\sigma_2\cdots\sigma_n$ of the integers $\{1, 2, \ldots, n\}$, an *inversion* is a pair $\sigma_i > \sigma_j$ corresponding to a pair of indices $i < j$. Let $Inv(\sigma)$ denote the total number of inversions of $\sigma$. For example for $\sigma = 4\ 3\ 5\ 2\ 1$ the set of inversions is $\{(5,2), (5,1), (4,3), (4,2), (4,1), (3,2), (3,1), (2,1)\}$ and thus $Inv(\sigma) = 8$. For any permutation $\sigma$, $0 \le Inv(\sigma) \le \frac{1}{2}n(n-1)$ with $Inv(\sigma) = 0$ iff $\sigma = 1\,2\cdots n$ is the identity permutation and $Inv(\sigma) = \frac{1}{2}n(n-1)$ iff $\sigma = n\cdots 2\,1$ is the reverse of the identity permutation. For the details of the underlying partially ordered set see [9]. Using the idea of inversions, we can put a measure of similarity on permutations. Given two permutations $\sigma$ and $\tau$, we count the number of inversions $\sigma$ would have if we were to use $\tau_1\tau_2\cdots\tau_n$ as the index set. In other words we compute $Inv(\sigma\tau^{-1})$. Put

$$\pi(\sigma,\tau) = \frac{2}{n(n-1)} Inv(\sigma\tau^{-1}) \tag{6}$$

to normalize this measure to the unit interval. It can be shown that $\pi$ is a *metric* on permutations. However, we cannot realistically use the permutations $\sigma = \pi(x)$ and $\tau = \pi(y)$ introduced in section 2 to compute this distance, since then there is no reduction in the dimension. To overcome this, we consider *bin-score* permutations. For simplicity, assume $n = 2^r$ and $\sigma$ is a permutation on $\{1, 2, \ldots, n\}$. For any integer $s = 0, 1, \ldots, r$, we may divide the index set into $b = 2^s$ consecutive subsets (bins) of length $b' = \frac{n}{b} = 2^{r-s}$ each. The $i$-th bin is described by the indices $i_1 = (i-1)b' + 1$, $i_2 = (i-1)b' + 2, \ldots, i_{b'} = (i-1)b' + b'$.

The *score* of this bin is the sum $\sigma_{i_1} + \sigma_{i_2} + \cdots + \sigma_{i_{b'}}$. In this way we construct a myopic version of $\sigma$ on $\{1, 2, \ldots, b\}$ obtained by placing 1 for the smallest entry among the bin-scores computed, 2 for the next smallest, etc. In case of ties, we make the indices increase from left to right, as in the case of the construction of $\pi(x)$ described in section 2 (in fact, this permutation is simply $\pi(x')$ where $x'$ is the $b$-dimensional vector of bin-scores of $x$). The bin-score permutation corresponding to $b = n$ is $\sigma$ itself, and for $b = 1$ it is the singleton 1. As an example, for $n = 8$, the bin-score permutations of $\sigma = 5\ 8\ 2\ 6\ 4\ 3\ 1\ 7$ for $b = 4, 2$ are obtained from the scores $13, 8, 7, 8$, and $21, 15$ as the permutations $4\ 2\ 1\ 3$ and $2\ 1$, respectively. Note that any bin-score permutation can be obtained by repeated application of the $b' = 2$ case.

# 5 Experiments

For the experiments, we have used time series data of the seasonally adjusted local area unemployment rate figures (Local Area Unemployment Statistics) for the 51 states supplied online by the U.S. Department of Labor's *Bureau of Labor Statistics*. The monthly rates were extracted for 256 months, covering a period between January 1979 through April 2000 for each state[1]. The dataset we used for the experiments conducted consisted of 51 vectors in $\mathbb{R}^{256}$ of the states, alphabetically ordered and indexed as $x[1], x[2], \ldots, x[51]$. For the query vector $y$, we used the unemployment rate figures for the same period for the seasonally adjusted national average figures. The purpose of the experiments can be thought of as determining which state in the union has a unemployment rate history that is closest to the national average, where closest can be given different meanings by altering the bias parameter $\lambda$ of the algorithm.

**Estimation of the parameters: the magnitude part**

The maximum coordinate in the dataset was found to be 19.5 and the minimum entry as 2.1. Since we have no reason to expect the data to be uniform in this interval, we computed $b_1$ and $b_2$ using (4) after a linear normalization to the unit interval. To compute the number of sample vectors of dimension

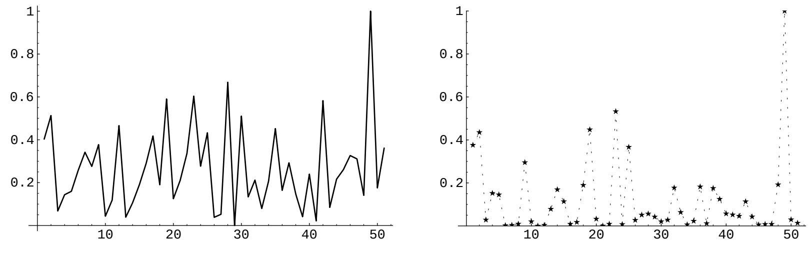


Figure 1: The magnitude part: normalized actual distances (left), normalized approximations (right).

256 required to obtain a meaningful estimate using (4) and the estimates of

[1] In California unemployment rates $x[5]$, the 1979 year data supplied were 0.0. These were all replaced with the January 1980 value of 5.9.

the first four moments of the density obtained through (5), we first experimented with uniform distribution for which we know the asymptotic values $b_1 = -\frac{1}{16}$, $b_2 = \frac{45}{64}$. Two to three 256-dimensional vectors were enough for convergence. Computing with 51x256=13056 normalized sample coordinates in the dataset, the approximate moments were calculated by Mathematica to be $\mu_1 = 0.236$, $\mu_2 = 0.072$, $\mu_3 = 0.026$, $\mu_4 = 0.012$. Using the formulas (4) gives $b_1 = 0.017$, $b_2 = 0.415$. With these values, we computed the the summary data $\psi_1(x[1]), \ldots, \psi_1(x[51])$ and $\psi_2(x[1]), \ldots, \psi_2(x[51])$ required. For every vector $x$ in the dataset, we calculate the approximation to $< x, y >$ with the computed values of $b_1$ and $b_2$ as $(b_1\psi_1(x)\psi_1(y) + b_2\psi_2(x)\psi_2(y))^{\frac{1}{2}}$ , and then computed the approximate distance $s(x, y)$ through (2). To see how the approximations to the magnitude part and the actual Euclidean distance values compare, we plotted the normalized values for the 51 states in Figure 1. Considering that we are only using the $m = 2$ algorithm for the computation of $s(x, y)$, i.e. the dimension is vastly reduced, the results are satisfactory.

**Estimation of the parameters: the shape part**

To get an idea on the number of bins $b$ to use for the computation of the approximate values $\pi(x, y)$, we calculated vectors of distances $\pi(x, y)$ through the expression (6) with bin-score permutations instead of the actual permutations. Bin-score permutations for each vector $x[1], \ldots, x[51]$ and the query vector $y$ was computed for $b$ ranging from 4 to 256. The resulting distances are plotted in Figure 2. From the figure, it is clear that even $b = 8$ is a reasonable approximation to the actual curve (i.e. $b = 256$). In the experiments we used the $b = 16$ bins.

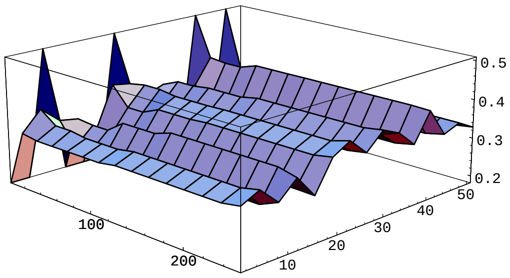


Figure 2: The shape part: plot of bin–score permutation distances for 4–256 bins.

**Parametric Experiments and Conclusions**

In Figures 3–6, the data plotted is the seasonally adjusted monthly unemployment rates for 256 months spanning January 1979 through April 2000. In each figure, the plot on the left is the actual rates through this time period, and the one on the right is the plot of the rates sorted in increasing order. Consider the function (1) for $\lambda$ changing from 0 to 1 in increments of 0.01, where for each $\lambda$ value, $s(x, y)$ makes use of the approximate distance computation described for $m = 2$, and $\pi(x, y)$ is the distance between the bin-score permutations $\pi(x)$ and $\pi(y)$ for $b = 16$ bins. For each value of $\lambda$ in this range we computed the state (i.e. the vector $x[i]$) where the minimum is obtained.

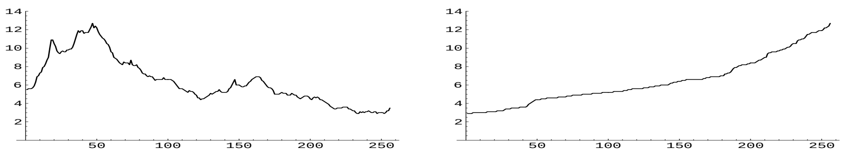


Figure 3: $x[15] =$ State of Indiana unemployment rate data.

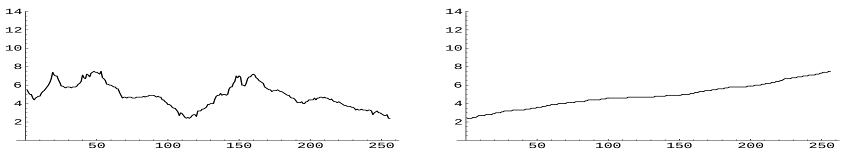


Figure 4: $x[46] =$ State of Vermont unemployment rate data.

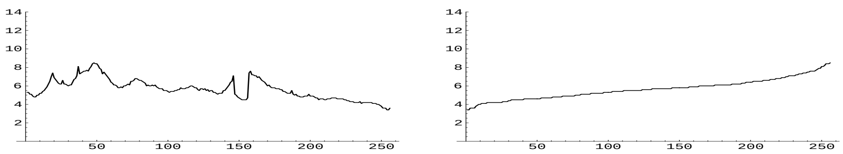


Figure 5: $x[11] =$ State of Georgia unemployment rate data.

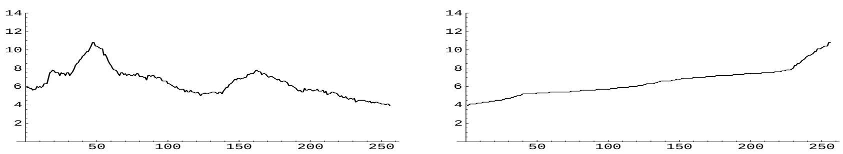


Figure 6: Query data $y =$ National average unemployment rates.

- For $0.0 \le \lambda < 0.5$, the minimum is obtained at $x[15]$, the state of Indiana,
- For $0.5 \le \lambda \le 0.9$, the minimum is obtained at $x[46]$, the state of Vermont,
- For $0.9 < \lambda \le 1.0$, the minimum is obtained at $x[11]$, the state of Georgia.

The observed "continuity" of these results as a function of $\lambda$ is a desirable aspect of any such family of algorithms.

Figures 3–6 indicate the behavior of the algorithm on the dataset. For small values of $\lambda$, the bias is towards the *shapes* of the curves. In these cases the algorithm finds the time-series data of the national rates (Figure 6, left), resemble most that of Indiana (Figure 3, left) out of the 51 states in the dataset. On the other extreme, for values of $\lambda$ close to 1, the bias is towards the *magnitudes*, and the algorithm finds the sorted data of the national rates (Figure 6, right), resemble most that of the state of Georgia (Figure 5, right). The intermediate values pick the state of Vermont (Figure 4) as the closest to the national average rates.

In conclusion, we proposed a spectrum of dynamic dimensionality reduction algorithms based on the approximation of the standard inner-product, and bin-score permutations based on an inversion measure on permutations. The experiments on time-series data show that with this technique, the similarity between two objects in high-dimensional space can be well approximated by a significantly lower dimensional representation. We remark that even though we used a convex combination of the two measures controlled by a parameter $\lambda$, it is possible to combine $s(x, y)$ and $\pi(x, y)$ for the final similarity measure in many other ways, $s(x, y)^{\lambda}\pi(x, y)^{1-\lambda}$, for example.

## 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*, pp. 69–84, 1993.

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

[3] 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.

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

[5] Ö. Eğecioğlu and H. Ferhatosmanoğlu, Dimensionality reduction and similarity computation by inner product approximations. *Proc. 9th Int. Conf. on Information and Knowledge Management (CIKM'00)*, Nov. 2000, Washington DC.

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

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

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

[9] J. E. Humphreys. *Reflection Groups and Coxeter Groups,* Cambridge Studies in Advanced Mathematics, No. 29, Cambridge Univ. Press, Cambridge, 1990.

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