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. The former is based on a concept called *bin-score permutations* and a metric to quantify similarity of permutations, the latter on another novel approximation for inner-product computations based on power symmetric functions, which generalizes the Cauchy-Schwarz inequality. 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. Some examples of such applications are document databases [6], medical imaging [16], and multimedia information systems [18]. 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 [13, 6]. 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 systems. This support depends on the development of efficient techniques to support approximate searching. To this end, a number of index structures for retrieval of multi-dimensional data along with associated algorithms for similarity search have been developed [11,19,4]. For time-series data, there are a number of proposed ways to measure similarity. These range from the Euclidean distance to non-Euclidean metrics and the representation of the sequence by appropriate selection of local extremal points [17]. Agrawal, Lin, Sawhney, and Shim [1] considered fast similarity search in the presence of noise, scaling, and translation by making use of the L_∞ norm. Bollobas,

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

L. De Raedt and A. Siebes (Eds.): PKDD 2001, LNAI 2168, pp. 79–90, 2001.

Das, Gunopulos, and Mannila [2] considered similarity definitions based on the concept of well-separated geometric sets. It has been noted in the literature however, that as dimensionality increases, query performance degrades significantly, an anomaly known as the dimensionality curse [5,10]. Common approaches for overcoming the dimensionality curse by dimension 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 [3].

This paper introduces a spectrum of similarity algorithms which consist of a convex combination of two different measures. A *shape* measure on high-dimensional vectors based on the similarity of permutations through inversion pairs, followed by an associated dimension reduction by bin-score permutations; and a symmetric *magnitude* measure based on the computation of the inner-product and consequently the cosine of the angle between two vectors by a low dimensional representation.

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 $(s(x), \sigma(x))$ where $s(x)$ is the sorted version of x into weakly increasing coordinates, and $\sigma(x)$ is the permutation of the indices $\{1, 2, \ldots, n\}$ that achieves this ordering. We impose the additional condition that the elements of the permutation $\sigma(x)$ 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, $\sigma(x) = 6\ 3\ 7\ 9\ 5\ 1\ 2\ 4\ 8$. 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

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

where

- $s(x, y)$ is a measure of distance between $s(x)$ and $s(y)$ 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 $\sigma(x)$ and $\sigma(y)$ (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 amenable to computation using data with reduced dimensionality $\ll n$. In the technique proposed here, this reduced dimension can be selected separately and independently for the two parts. First we discuss the construction of the parts themselves and then present the results of the experiments.

The outline of this paper is as follows. In Sect. 3 we consider the fast approximate calculation of $s(x, y)$ which is based on a novel low-dimensional representation to compute the inner product introduced in [7] and developed in [8]. Section 4 describes how to measure the distance $\pi(x, y)$ on permutations with a low-dimensional representation. This is based on a metric on permutations that we introduce, and the approximation of the metric by *bin-score* permutations. Experiments on labor statistics time-series data are presented in Sect. 5, and conclusions in Sect. 6.

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 [7] and further developed in [8]. 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

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

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

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

for large n, where the b_i are universal 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 [12]. In [7] 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 shown to be

$$b_1 = -\frac{1}{16}\ , \qquad b_2 = \frac{45}{64}. \qquad (4)$$

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

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

In fact in the general case of a density with i-th moment μ_i (about the origin), it can be proved [7] that the constants b_1, b_2 are functions of the first four moments of the density $f(x)$. They are given by the formulas

$$\begin{aligned} b_1 &= \mu_1^2 \cdot \frac{2\mu_2^3 + \mu_1^2\mu_4 - 3\mu_1\mu_2\mu_3}{\mu_2^3 + \mu_1^2\mu_4 - 2\mu_1\mu_2\mu_3}, \\ b_2 &= \frac{\mu_1^4}{\mu_2} \cdot \frac{\mu_1\mu_3 - \mu_2^2}{\mu_2^3 + \mu_1^2\mu_4 - 2\mu_1\mu_2\mu_3} . \end{aligned} \tag{5}$$

The moments of the uniform distribution are $\mu_i = 1/(i + 1)$, for which the formulas in (5) reduce to the values in (4) above.

A secondary problem of interest in the context of the determination of the best set of constants is dynamic in nature. When the contents of the database changes by adding new data vectors, for example, the parameters used for the approximation problem to the inner-product calculation can be adjusted efficiently. In particular, one *need not* know the density of the distribution of the coordinates in the dataset parametrically. The moments u_i can be estimated as the limit of the N-th estimate $\overline{\mu_i}(N)$ as the dataset is accumulated via

$$\overline{\mu_i}(N+1) = \frac{1}{N+1}\left(N\overline{\mu_i}(N) + t_{N+1}^i\right). \tag{6}$$

where t_N is the N-th sample coordinate observed.

4 The *Shape* Part: Bin-Score Permutations

For a permutation $\rho = \rho_1\rho_2\cdots\rho_n$ of the integers $\{1, 2, \ldots, n\}$ in one-line notation, an *inversion* is a pair $\rho_i > \rho_j$ corresponding to a pair of indices $i < j$. Let $Inv(\rho)$ denote the total number of inversions of ρ. For example for $\rho = 4\ 3\ 5\ 2\ 1$ the set of inversions is $\{(5,2), (5,1), (4,3), (4,2), (4,1), (3,2), (3,1), (2,1)\}$ and thus $Inv(\rho) = 8$. For any permutation ρ,

$$0 \leq Inv(\rho) \leq \tfrac{1}{2}n(n-1)$$

with $Inv(\rho) = 0$ iff $\rho = 1\ 2\cdots n$ is the identity permutation and $Inv(\rho) = \frac{1}{2}n(n-1)$ iff $\rho = n\cdots 2\ 1$ is the reverse of the identity permutation. For the details of the underlying partially ordered set see [14]. Inversions arise naturally in the context of sorting as a measure of presortedness [15] when the number of comparisons is the basic measure. The idea of counting inversions is one of many ways of putting a measure of similarity on permutations [9]. Given two permutations ρ and τ, we count the number of inversions ρ would have if we were to use $\tau_1 \tau_2 \cdots \tau_n$ as the index set. In other words we compute $Inv(\rho\tau^{-1})$. Put

$$\pi(\rho,\tau) = \frac{2}{n(n-1)} Inv(\rho\tau^{-1}) \tag{7}$$

to normalize this measure to the unit interval. Some relevant properties of π are as follows

1. $0 \leq \pi(\rho,\tau) \leq 1$,
2. $\pi(\rho,\tau) = 0$ iff $\rho = \tau$,
3. $\pi(\rho,\tau) = 1$ iff $\rho + \tau_i = n+1$ for $i = 1, 2, \ldots n$,
4. $\pi(\rho,\tau) = \pi(\tau,\rho)$,
5. $\pi(\rho,\tau) \leq \pi(\rho,\delta) + \pi(\delta,\tau)$ for any permutation δ.

In particular π is a *metric* on permutations. However, we cannot realistically use the permutations $\rho = \sigma(x)$ and $\tau = \sigma(y)$ introduced in Sect. 2 to compute this distance, since then there is no reduction in the dimension. The question is then whether or not approximations to the permutations ρ and τ by some lower dimensional representation can be made, that would allow us to compute this measure without much deviation from the actual value.

To this end, we consider *bin-score* permutations. For simplicity, assume $n = 2^r$ and ρ 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 b' consecutive 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 $\rho_{i_1} + \rho_{i_2} + \cdots + \rho_{i_{b'}}$. In this way we construct a myopic version of ρ 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 $\sigma(x)$ described in Sect. 2 (in fact, this permutation is simply $\sigma(x')$ where x' is the b-dimensional vector of bin-scores of x). The bin-score permutation corresponding to $b = n$ is ρ itself, and for $b = 1$ it is the singleton 1. As an example, for $n = 8$, the bin-score permutations of $\rho = 5\ 8\ 2\ 6\ 4\ 3\ 1\ 7$ 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 the 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 as *Alabama* through *Wyoming*, and indexed as $x[1], x[2], \ldots, x[51]$. Thus each $x[i]$ is a vector in $n = 256$ dimensional space. 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 had an unemployment rate history that is closest to the national average for the period of time in question, where closest can be given different meanings by altering the bias parameter λ of the algorithm.

5.1 Estimation of the Parameters: The Magnitude Part

The maximum coordinate over all the vectors 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 (5) after a linear normalization to the unit interval.

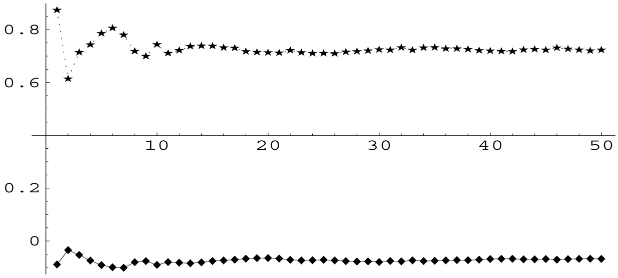

Fig. 1. The estimate to b_1 and b_2 computed for 50 vectors, each of dimension 16 with entries from the uniform distribution on [0,1]. The theoretically obtained asymptotic values are $b_1 = -0.0625$, $b_2 = 0.703125$

To compute the number of sample vectors of dimension 256 required to obtain a meaningful estimate using (5) and the estimates of the first four moments of the density obtained through (6), 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. We generated 50 vectors of dimension 16 each. The corresponding values of b_1 and b_2 calculated from the estimates of the moments are shown in Fig. 1. We see that the convergence requires about 20 vectors or about 300 samples. This means that the lower bound on the number of 256 dimensional vectors we need to obtain a reasonable estimates to b_1 and b_2 in the general case is very small.

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

Computing with 51 × 256 = 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 (5) gives

$$b_1 = 0.017 \ , \quad b_2 = 0.415 \tag{8}$$

With these values, we computed the the summary data $\psi_1(x[1]), \dots, \psi_1(x[51])$ and $\psi_2(x[1]), \dots, \psi_2(x[51])$ required.

The following vector of length 51 gives the (approximate) ψ_1 values of the normalized vectors computed in this fashion:

85.3, 95.5, 58.4, 74.1, 74.1, 47.9, 43.7, 48.5, 86.0, 58.3, 52.1, 43.3, 66.6, 73.2, 63.1, 43.3, 36.7,
75.2, 92.3, 58.7, 48.2, 47.1, 90.9, 41.2, 87.8, 56.9, 65.0, 22.7, 60.1, 34.2, 58.6, 78.2, 66.2, 45.0,
34.4, 72.0, 52.1, 74.6, 69.1, 60.4, 60.6, 27.2, 66.7, 62.9, 44.4, 40.5, 40.3, 75.7, 117.7, 51.6, 53.1

and the following the ψ_2 values computed

34.6, 37.6, 15.3, 23.8, 23.5, 10.6, 9.0, 11.8, 31.1, 14.7, 11.5, 8.8, 19.4, 24.6, 21.0, 10.1, 5.8,
25.7, 38.0, 15.6, 10.4, 11.4, 41.6, 8.4, 34.3, 15.0, 17.6, 3.1, 16.5, 6.9, 15.3, 25.3, 18.5, 10.2,
5.5, 25.1, 12.9, 24.9, 22.1, 17.5, 17.2, 3.5, 21.3, 16.9, 9.9, 7.8, 7.5, 25.8, 62.0, 14.4, 13.2

For example for the state of Alabama, the summary magnitude information is

$$\psi_1(x[1]) = 85.3 \quad \text{and} \quad \psi_2(x[1]) = 34.6.$$

We also calculated that for the query vector y of normalized national average rates, the two ψ values are

$$\psi_1(y) = 63.9 \quad \text{and} \quad \psi_2(y) = 17.8.$$

Now for every vector x in the dataset, we calculate the approximation to $< x, y >$ as

$$\sqrt{b_1\psi_1(x)\psi_1(y) + b_2\psi_2(x)\psi_2(y)}$$

where b_1 and b_2 are as given in (8). Therefore as a measure of distance of the symmetric part, we set

$$s(x, y) = \sqrt{|\psi_2(x) + \psi_2(y) - 0.0350342\psi_1(x)\psi_1(y) - 0.82938\psi_2(x)\psi_2(y)|}$$

by using (2).

To see how the approximations to the magnitude part and the actual Euclidean distance values compare, we plotted the normalized actual values and the normalized approximations for the 51 states in Fig. 2. 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.

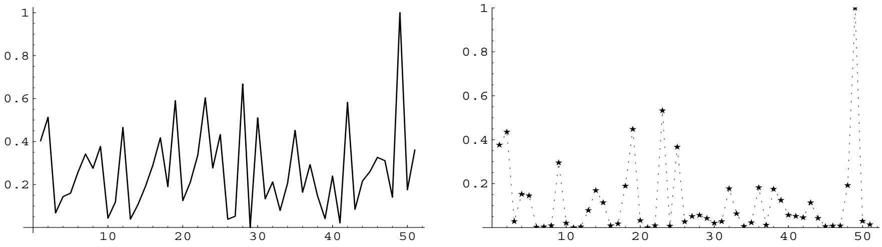

Fig. 2. The magnitude part: normalized actual distances (left), normalized approximations (right)

5.2 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 (7) 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 Fig. 3. 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 case of $b = 16$ bins.

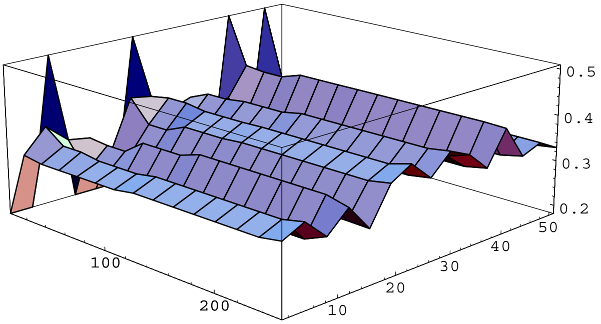

Fig. 3. The shape part: plot of bin–score permutation distances for 4–256 bins

6 Parametric Experiments and Conclusions

In Figs. 5–8, 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 λ changing from 0 to 1 in increments of 0.01, where for each λ 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 $\sigma(x)$ and $\sigma(y)$ for $b = 16$ bins. For each value of λ in this range we computed the state (i.e. the vector $x[i]$) where the minimum approximate distance is obtained.

- For $0.0 \leq \lambda < 0.5$, the minimum is obtained at $x[15]$, which corresponds to the state of Indiana,

- For $0.5 \leq \lambda \leq 0.9$, the minimum is obtained at $x[46]$, which corresponds to the state of Vermont,

- For $0.9 < \lambda \leq 1.0$, the minimum is obtained at $x[11]$, which corresponds to the state of Georgia.

The observed “continuity” of these results as a function of λ is a desirable aspect of any such family of algorithms.

Figures 5–8 indicate the behavior of the algorithm on the dataset. For small values of λ, the bias is towards the *shapes* of the curves. In these cases the algorithm finds the time-series data of the national rates (Fig. 8, left), resemble most that of Indiana (Fig. 5, left) out of the 51 states in the dataset. On the other extreme, for values of λ close to 1, the bias is towards the *magnitudes*, and the algorithm finds the sorted data of the national rates (Fig. 8, right), resemble most that of the state of Georgia (Fig. 7, right). The intermediate values pick the state of Vermont (Fig. 6) 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 λ, 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. In any such formulation, the determination of the best value of λ for a given application will most likely require the experimental evaluation of the behavior of the approximate distance function by sampling from the dataset.

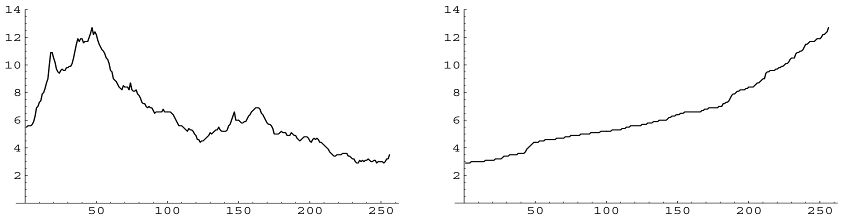

Fig. 4. $x[15] =$ State of Indiana unemployment rate data: actual (left), sorted (right). For parameter λ in the range $0.0 \leq \lambda < 0.5$, the algorithm picks the state of Indiana's data as the one closest to the national average data in Fig. 7.

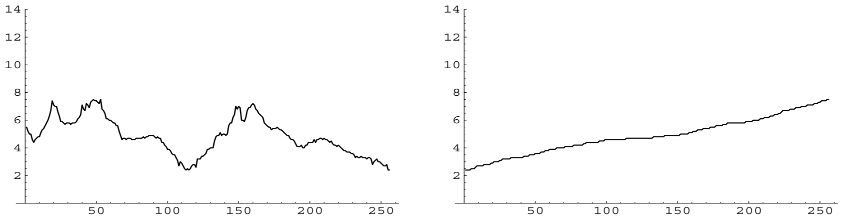

Fig. 5. $x[46] =$ State of Vermont unemployment rate data: actual (left), sorted (right). For parameter λ in the range $0.5 \leq \lambda \leq 0.9$, the algorithm picks the state of Vermont's data as the one closest to the national average data in Fig. 7.

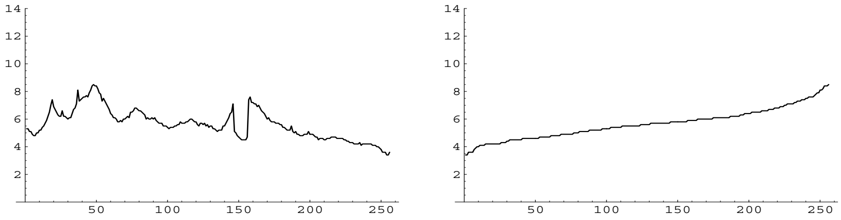

Fig. 6. $x[11] =$ State of Georgia unemployment rate data: actual (left), sorted (right). For parameter λ in the range $0.9 < \lambda \leq 1.0$, the algorithm picks the state of Georgia's data as the one closest to the national average data in Fig. 7

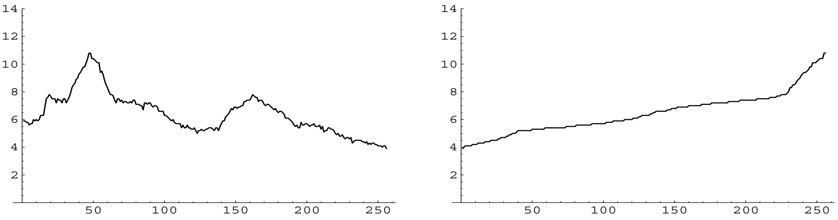

Fig. 7. Query data $y =$ National average unemployment rates: actual (left), sorted (right). The dataset is the seasonally adjusted monthly unemployment rates for 256 months spanning January 1979 through April 2000

References

1. R. Agrawal, K-I. Lin, H. S. Sawhney, and K. Shim. Fast Similarity Search in the Presence of Noise, Scaling, and Translation in Time-Series Databases. *The VLDB Journal*, pp. 490–501, 1995.
2. B. Bollobas, G. Das, D. Gunopulos, and H. Mannila. Time-Series Similarity Problems and Well-Separated Geometric Sets. *Proc. of 13th Annual ACM Symposium on Computational Geometry*, Nice, France, pp. 454–456, 1997.
3. 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.
4. 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.
5. S. Berchtold, C. Bohm, D. Keim, and H. Kriegel. A cost model for nearest neighbor search in high-dimensional data space. In *Proc. ACM Symp. on Principles of Database Systems*, Tuscon, Arizona, 1997.
6. 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.
7. 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.
8. 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.
9. V. Estivill-Castro and D. Wood. A Survey of Adaptive Sorting Algorithms. *ACM Computing Surveys*, Vol. 24, No. 4, pp. 441–476, 1992.
10. 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.
11. 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.
12. N.A.J. Hastings and J.B. Peacock. *Statistical Distributions,* Halsted Press, New York, 1975.

13. 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.
14. J. E. Humphreys. *Reflection Groups and Coxeter Groups,* Cambridge Studies in Advanced Mathematics, No. 29, Cambridge Univ. Press, Cambridge, 1990.
15. D. Knuth. *The art of computer programming* (Vol. III), Addison-Wesley, Reading, MA, 1973.
16. Korn F., Sidiropoulos N., Faloutsos C., Siegel E., and Protopapas Z. Fast nearest neighbor search in medical image databases. In *Proceedings of the Int. Conf. on Very Large Data Bases*, pages 215–226, Mumbai, India, 1996.
17. C-S. Perng, H. Wang, S. R. Zhang, and D. S. Parker. Landmarks: a new model for similarity-basedpattern querying in time-series databases. *Proc. of the 16-th ICDE*, San Diego, CA, 2000.
18. 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.
19. D. White and R. Jain. Similarity indexing with the SS-tree. In *Proc. Int. Conf. Data Engineering*, pp. 516–523, 1996.