Applied Mathematics & Information Sciences
An International Journal

Isomorphism Distance in Multidimensional Time Series and Similarity Search

Guo Wensheng and Ji Lianen

College of Geophysics and Information Engineering, China University of Petroleum-Beijing, Beijing, China

Received: 19 Oct. 2012, Revised: 21 Nov. 2012, Accepted: 1 Dec. 2012
Published online: 1 Feb. 2013

Abstract: Describing the similarity of time series as distance is the basis for most of data mining research. Existing studies on similarity distance is based on the "point distance" without considering the geometric characteristics of time series, or is not a metric distance which doesn't meet the triangle inequality and can't be directly used in indexing and searching process. A method for time series approximation representation and similar measurement is proposed. Based on the subspace analysis representation, the time series are represented approximately with an isomorphic transformation. The basic concepts and properties of the included isomorphism distance are proposed and proved. This distance overcomes the problem when other non-metric distance is used as the similar measurement, such as the poor robustness and ambiguous concepts. The proposed method is also invariant to translation and rotation. A new pruning method for indexing in large time series databases is also proposed. Experimental results show that the proposed method is effective.

Keywords: Time series, similarity, metric space, data mining

1. Introduction

With the continuous development of the technical level, there are lots of time series data in commerce[1], science[2] and engineering[3]. Such as the sales of commodities in retail, stock price, number of security incidents detected by security facilities deployed in the network, and so on. Analysis of those data will often be relevant to an important issue: how to find the sequence which is similar to the given query from the time series of historical data. For example: In the financial field, people can search the time series similar to the recent stock price changes of one company in the historical time series data, and then predict the future stock price changes according to the historical time series data. Another example is in the field of network security, by looking up the historical time series records similar to recent network traffic and security events, people can identify network security posture and possible attacked events.

In 1993, Agralwal et al. first proposed a total matching algorithm in time series similarity search [4]. Faloutsos et al. who proposed a subsequence matching algorithm [5], promote the application of similarity search. The traditional methods are one-dimensional sequence similarity search, and achieve great success in their respective fields of application [6,7]. However, with the prevalence and popularity of audio and video equipment and the internet, most of the one-dimensional time series similarity search method does not apply to new data format, so the multi-dimensional similarity search is proposed.

Multidimensional time series, including graphics, images, audio, video and other information, is composed by a set of data vectors change over time. For example: In the financial field, the timing data for Chinese stock index recently points, can be searched not only in their own historical data set, but also can be tried to search for similar subsequence in other countries' stock index historical data set for policy making [8]. Searching process of timing data in the field of network security, which is formed by the number of hosts controlled by some kinds of Trojan horse in a region, in addition to concentrate in its own Trojan historical data, can also refer to other regions or country to find the similar behavior mode for further analysis and decision-making[9].

Extending similar search to the multidimensional scene can obtain the following two advantages: Firstly, known by the research on the data stream, generally, the recent data is more valuable than long time ago, so recent similar sequence found in multiple dimensions may have

* Corresponding author e-mail: gwsice@126.com

more value than those obsolete sequence found in one dimension. Secondly, with technology development, a variety of time series data in recent years is gradually increased. Because of this, there may not be sufficient historical data to support applications of similarity search in a single dimension. By searching in multi-dimensional time series data, we can increase the scale of search space to discover more valuable similar historical mode for decision making.

Our contributions are as follows:

We propose a distance function which we call isomorphism distance(ISO). This distance function maintain the geometric characteristics of time series through subspace isomorphic [10,11]. So it can support local time shifting, and is a metric. We present benchmark results showing that this distance function is natural for time series data.

We propose a new pruning strategy for isomorphism distance, which can be efficiently indexed with a standard B+-tree or other data structure. Given that ISO is a metric distance, we can use the triangle inequality in the pruning process.

We also develop a k-nearest neighbor(k-NN) algorithm that use the isomorphism distance. We give extensive experimental results in Section 5 showing that the algorithm gets the best of pruning power and scalability.

The rest of the paper is organized as follows.In Section 2 we introduce the related works about the distance measuring similarity and its indexing structure. In section 3 we present our isomorphism distance model and prove it's a metric distance. Because of its metric measure, some indexing and pruning algorithm are analyzed in Section 4. The experimental results are presented in Section 5. We conclude our paper and suggest some possible future directions in Section 6.

2. Related Work

Many researches focus on how to search similar sequence fast and accurate in time series database, especially large database once unable to load in memory. It includes how to represent time series, how to measure the similarity between sequences as well as how to index and search in database. The major role of sequence represented is dimensionality reduction and feature extraction, which can resolve dimensionality curse [12]. There are many commonly used methods in sequence represented field, such as discrete Fourier transform(DFT)[4,13], discrete wavelet transform(DWT)[14], singular value decomposition(SVD)[15], piecewise method[16,17], and so on. DFT can convert time series into the frequency domain, take the first few strong Fourier coefficient as a sequence represented in order to achieve dimensionality reduction. DFT is suitable for those natrually occurring sinusoidal signals, but not for discontinuous signal. Haar wavelet transform is the most common one in all of the DWT methods. However, for the basis function is not smooth, Haar wavelet can only use staircase approximation to the analog signal. Therefore, a continuous function can not be well approximated by only a small number of Haar wavelet transform coefficients, so more wavelet coefficients are needed. SVD is a dimensionality reduction method depends on data content. By calculation of a given data set of eigenvalues and eigenvectors, SVD converts data to make the most of information in some dimension, then take the data in the coordinates of those dimensions as compression of the original data set. Main weakness of SVD is the eigenvectors need to be recalculate when data changes. Therefore, SVD is not suitable for dynamic changes in database. Piecewise method uses piecewise sequence and its feature(the extreme points and trends) to represent original time series.

The motivation for seeking new similarity measures is that the Euclidean distance can not effectively reflect the shape and dynamic characteristics of time series. It is too weak to handle noise and local time shifting. Berndt and Clifford [18] introduced Dynamic time wraping(DTW) to allow a time series to be "stretched" to provide a better match with another time series. Das et al. [19] and Vlachos et al. [20] applied the LCSS measure to time series matching. Chen et al. [21] applied EDR to trajectories. However, none of DTW, LCSS and EDR is a metric distance function for time series.

Most of the time sequence index is based on the GEMINI framework. However, if the distance measure is a metric, a large number of index structure and technology for the measure can be used. For example, the MVP-tree [22], the M-tree [23], the Sa-tree [24] and the OMNI-family of access methods [25]. A survey of metric space indexing is given in [26].

3. Isomorphism Distance

Existing time series similarity measure is based on two major types of distance function. The first type consists of the L_p-norms (e.g. Euclidean distance and edit distance[27,28]), which are metric distance but cannot support local time shifting. The second type consists of distance functions which are capable of handling local time shifting but are non-metric. Figure 3 compares the nature of difference in these types of distance.

On this basis, assuming that multidimensional time series lie in a linear manifold in the data space,we propose a new distance function, which is called isomorphism distance. This distance function maintain the geometric characteristics of time series through subspace isomorphic. So it can support local time shifting, and is a metric. To begin with, for any two time series $[s_1,...,s_m]$ and $[t_1,...,t_n]$, consider them as two linear subspaces S and T in $\mathscr{R}^d$. Since discuss similarity issue, we first assume that S and T have the same dimensionality. Let $s_1,...,s_p$ and $t_1,...,t_q$ be standard

$D(S,T)$	$=$	$\left(\sum_{i=1}^{n} \lvert s_i - t_i \rvert^2\right)^{1/2}$	s,t *is time series point*
$EDR(s_i,t_i)$	$=$	$\begin{cases} 0 \\ 1 \\ 1 \end{cases}$	$s_i = t_i$ *if* s_i *or* t_i *must add a symbol* *otherwise*
$DTW(s_i,t_i)$	$=$	$\lvert r_i - s_i \rvert + min \begin{cases} DTW(s_i, t_{i-1}) \\ DTW(s_{i-1}, t_i) \\ DTW(s_{i-1}, t_{i-1}) \end{cases}$	
$ISO(S,T)$	$=$	$\left(n - \sum_{i=1}^{n}\sum_{j=1}^{n} (s_i^T t_i)^2\right)^{1/2}$	s,t *is the SOD of* S,T

Figure 1 Comparing the Distance Functions: Euclidean Distance(D), Edit Distance(EDR), Dynamic Time Warping Distance(DTW) and Isomorphism Distance(ISO)

orthogonal basis of S and T, respectively. Let $d(s_i,T)$ denote the so-called *isomorphism distance* from the end point of vector s_i to subspace T. That is,

$$d(s_i,T) = \min_{t \in T} ||s_i - t|| \tag{1}$$

We then define the subspace isomorphism distance $d(S,T)$ for p-dimensional subspaces S and q-dimensional subspaces T as

$$d(S,T) = \sqrt{\sum_{i=1}^{p} d^2(s_i,T)} \tag{2}$$

Since $t_1,t_2,\ldots,t_q$ is a standard orthogonal basis of T, it is easy to see that

$$d(S,T) = \sqrt{\sum_{i=1}^{p}\left[||s_i||^2 - \sum_{j=1}^{q}(s_i^T t_j)^2\right]} \tag{3}$$

With the above-mentioned Distance, we need prove the following properties:

Theorem 1.*The isomorphism distance defined above is invariant to the choice of standard orthogonal basis.*

*Proof.*Let $s_1,s_2,\ldots,s_p$ and $\widetilde{s}_1,\widetilde{s}_2,\ldots,\widetilde{s}_p$ be two standard orthogonal basis of S. Let $t_1,t_2,\ldots,t_q$ be a standard orthogonal basis of T. To prove the theorem, it suffices to show that

$$\sqrt{p - \sum_{i=1}^{p}\sum_{j=1}^{q}(s_i^T t_j)^2} = \sqrt{p - \sum_{i=1}^{p}\sum_{j=1}^{q}(\widetilde{s}_i^T t_j)^2} \tag{4}$$

Let $P_S^{t_j}$ is the projection of t_j onto subspace S. By the Parseval equation and the uniqueness of projection, following equality holds:

$$\sum_{i=1}^{p}(s_i^T t_j)^2 = \sum_{i=1}^{p}(\widetilde{s}_i^T t_j)^2 \tag{5}$$

In fact, we can see that for every $j, j = 1,2,\ldots,q$, above equality always holds. So it completes the proof.

Theorem 2.*Non-negativity:*$0 \le d(S,T) \le \sqrt{\max(p,q)}$

The proof of Theorem 2 is immediate.

Theorem 3.*Symmetry:*$d(S,T) = d(T,S)$

The proof of Theorem 3 is immediate.

Theorem 4.*Triangle Inequality:*

$d(S,T) \le d(S,\Gamma) + d(T,\Gamma)$

Let $S = (s_1,\ldots,s_p)$, $T = (t_1,\ldots,t_q)$, $\Gamma = (\gamma_1,\ldots,\gamma_r)$ be the matrices composed by the orthogonal basis of arbitrary subspaces S,T,Γ, respectively.

Lemma 1.*Let A^H denote the conjugate transpose matrix of A. The trace of Matrix $A^H A$ and AA^H is equivalent, i.e.*

$$tr(A^H A) = tr(AA^H) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}^* a_{ji} \tag{6}$$

The proof of Lemma 1 is immediate according to the properties of trace in [29]. With Lemma 1, we can rewrite the Equation 3 in terms of matrix as follows:

$$d(S,T) = \sqrt{\max(p,q) - tr(TT^T SS^T)} \tag{7}$$

Then Theorem 4 can be written as a matrix format.

Lemma 2.*Denote Λ_p the diagonal matrix, which first p diagonal elements are 1, and rest elements are 0, that is,$\Lambda_p = diag(1,\ldots,1,0,\ldots,0)$.*

$max(p,q) = tr(\Lambda_p + \Lambda_q - \Lambda_p\Lambda_q)$

The proof of Lemma 2 is immediate.

Lemma 3.*If we denote $\widetilde{S} = (s_1,\ldots,s_p,\ldots,s_d)$ the orthogonal basis matrix of $\mathscr{R}^d$ extended from $s_1,\ldots,s_p$, then*

$(SS^T) = \widetilde{S}\Lambda_p\widetilde{S}^T$

The proof of Lemma 3 is immediate.

Lemma 4. *If we denote:*

$$A = tr[(\Lambda_p - \Lambda_r)(\Lambda_q - \Lambda_r) + (M - \Lambda_r)(N - \Lambda_r)]$$
$$B = tr[(M - \Lambda_r)^2 + (\Lambda_p - \Lambda_r)^2]$$
$$C = tr[(N - \Lambda_r)^2 + (\Lambda_q - \Lambda_r)^2]$$

Suppose that matrices U,V meet the following conditions: $\widetilde{S} = \widetilde{\Gamma}U$; $\widetilde{T} = \widetilde{\Gamma}V$, *then Theorem 4 is equivalent to :*

$$A \le \sqrt{B \cdot C} \tag{9}$$

Proof. Let $M = U\Lambda_p U^T$, $N = V\Lambda_q V^T$. According to Lemma 1 and the orthogonality of $\widetilde{\Gamma}$, we can obtain that:

$$\begin{aligned} tr(TT^TSS^T) &= tr(\widetilde{\Gamma}V\Lambda_q V^T\widetilde{\Gamma}^T\widetilde{\Gamma}U\Lambda_p U^T\widetilde{\Gamma}^T) \\ &= tr(\widetilde{\Gamma}V\Lambda_q V^T U\Lambda_p U^T\widetilde{\Gamma}^T) \\ &= tr(V\Lambda_q V^T U\Lambda_p U^T\widetilde{\Gamma}^T\widetilde{\Gamma}) \\ &= tr(V\Lambda_q V^T U\Lambda_p U^T) = tr(NM) \end{aligned}$$

and similarly,

$$\begin{aligned} tr(TT^T\Gamma\Gamma^T) &= tr(\widetilde{\Gamma}V\Lambda_q V^T\widetilde{\Gamma}^T\widetilde{\Gamma}\Lambda_r\widetilde{\Gamma}^T) \\ &= tr(\widetilde{\Gamma}V\Lambda_q V^T\Lambda_r\widetilde{\Gamma}^T) \\ &= tr(V\Lambda_q V^T\Lambda_r) = tr(N\Lambda_r) \end{aligned}$$

$$\begin{aligned} tr(SS^T\Gamma\Gamma^T) &= tr(\widetilde{\Gamma}U\Lambda_p U^T\widetilde{\Gamma}^T\widetilde{\Gamma}\Lambda_r\widetilde{\Gamma}^T) \\ &= tr(\widetilde{\Gamma}U\Lambda_p U^T\Lambda_r\widetilde{\Gamma}^T) \\ &= tr(U\Lambda_p U^T\Lambda_r) = tr(M\Lambda_r) \end{aligned}$$

According to Equation 7 and Lemma 1 we can obtain:

$$d(S,T) = \sqrt{tr(\Lambda_p + \Lambda_q - \Lambda_p\Lambda_q - NM)}$$
$$d(S,\Gamma) = \sqrt{tr(\Lambda_p + \Lambda_r - \Lambda_p\Lambda_r - M\Lambda_r)}$$
$$d(T,\Gamma) = \sqrt{tr(\Lambda_q + \Lambda_r - \Lambda_q\Lambda_r - N\Lambda_r)}$$

Therefore, Theorem 4 is equivalent to

$d^2(S,T) \le d^2(S,\Gamma) + d^2(T,\Gamma) + 2 \cdot d(S,\Gamma) \cdot d(T,\Gamma)$

That is

$d^2(S,T) - d^2(S,\Gamma) - d^2(T,\Gamma) \le 2 \cdot d(S,\Gamma) \cdot d(T,\Gamma)$

Let left side of this inequality as LS:

$$\begin{aligned} LS = tr[&(\Lambda_p + \Lambda_q - \Lambda_p\Lambda_q - NM) - (\Lambda_p + \Lambda_r \\ &- \Lambda_p\Lambda_r - M\Lambda_r) - (\Lambda_q + \Lambda_r - \Lambda_q\Lambda_r - N\Lambda_r) \\ &= tr(-\Lambda_p\Lambda_q - NM - \Lambda_r + \Lambda_p\Lambda_r \\ &+ M\Lambda_r - \Lambda_r + \Lambda_q\Lambda_r + N\Lambda_r) \\ &= tr[(\Lambda_p - \Lambda_r)(\Lambda_q - \Lambda_r) + (S - \Lambda_r)(T - \Lambda_r)] \end{aligned}$$

Let right side of this inequality as RS:

$$RS = \sqrt{tr(2\Lambda_p + 2\Lambda_r - 2\Lambda_p\Lambda_r - 2M\Lambda_r)} \cdot \sqrt{tr(2\Lambda_q + 2\Lambda_r - 2\Lambda_q\Lambda_r - 2N\Lambda_r)}$$

From $A^2 = A$, $\Lambda^2 = \Lambda$, we get

$$\begin{aligned} &tr(2\Lambda_p + 2\Lambda_r - 2\Lambda_p\Lambda_r - 2M\Lambda_r) \\ &= tr(\Lambda_p^2 + M^2 + \Lambda_r^2 + \Lambda_r^2 - 2\Lambda_p\Lambda_r - 2M\Lambda_r) \\ &= tr[(M - \Lambda_r)^2 + (\Lambda_p - \Lambda_r)^2] \end{aligned}$$

So

$$RS = \sqrt{tr[(M - \Lambda_r)^2 + (\Lambda_p - \Lambda_r)^2]} \cdot \sqrt{tr[(N - \Lambda_r)^2 + (\Lambda_q - \Lambda_r)^2]}$$

So Lemma 4 completes the proof. Now let's proof the Theorem 4: Set

$$a_1 = vec(M - \Lambda_r) = tr[(M - \Lambda_r)^2];$$
$$a_2 = vec(\Lambda_p - \Lambda_r) = tr[(\Lambda_p - \Lambda_r)^2];$$
$$a_3 = vec(N - \Lambda_r) = tr[(N - \Lambda_r)^2];$$
$$a_4 = vec(\Lambda_q - \Lambda_r) = tr[(\Lambda_q - \Lambda_r)^2];$$

where vec(A) indicates the vector that span with all of the matrix A's column vectors head to tail.According to above lemmas and Cauchy-Schwarz inequality, Theorem 4 is equivalent to

$$\begin{aligned} (a_1^T a_3 + a_2^T a_4)^2 &\le (\|a_1\|\|a_3\| + \|a_2\|\|a_4\|)^2 \\ &\le (\|a_1\|^2 + \|a_2\|^2)(\|a_3\|^2 + \|a_4\|^2) \end{aligned}$$

The whole proof is completed.

As mentioned above, the symmetry and non-negativity of isomorphism distance can be seen easily. Particularly, together with matrix analysis techniques, we show the triangular inequality of ISO. Therefore, it is proved to be a metric distance undoubtedly. The measure of the difference between different time series is the basis of many machine learning algorithm. Since it is proved to be distance, isomorphism distance becomes a natural distance measure to characterize the similarity between time series.

Specific solution of isomorphism distance is borrowed from the implementation in [30,31]: Firstly, according to solution idea of LDA algorithm, we can transform the solution process into the following generalized linear equation of eigenvalue and eigenvector problem. Secondly, assumptions to obtain the eigenvalues in ascending order, select the eigenvectors $s_1, s_2, \ldots, s_p$ corresponding to the first p(generally$p < m$), and then carry out the Gram-schmidt orthogonalization on $s_1, s_2, \ldots, s_p$ to meet the orthogonality.

4. Indexing and Searching

Recall from Figure 3 that isomorphism distance has the same computational behavior with EDR and DTW. it takes $O(mn)$ time to compute the distance for time series S, T of length m, n respectively. For huge time series database, it is important for a given query Q, we try to minimize the computation of the true distance between Q and S to measure the similarity of them for all sequence S in the database. The topic explore here is indexing and searching for k-NN(the k-Nearest Neighbor) query algorithm. An extension to the range queries is rather straightforward, so we omit its details. Given that isomorphism distance is a metric distance function,one obvious way to prune is to apply the triangle inequality. Metric or not, another common way to prune is to apply the GEMINI framework of Faloutsos et al. - that is, using lower bounds to guarantee no false dismissals. In fact, virtually all approaches to indexing time series under the Euclidean distance do that[13,5,32,33]. In this section, we can use a new solution to index and search similarity time series with isomorphism distance. The beauty of isomorphism distance is that it can be indexed by a simple B+-tree or R-tree.

4.1. Pruning by the Triangle Inequality

The algorithm 1 shows a skeleton of how the Triangle inequality is applied. S is the current time series, while Q is the query time series. The two-dimensional array *matrix* is used to stored the precomputed pairwise distance between two time series. The array *queue* is the array of time series with computed true distance to Q. It means that if the isomorphism distance $ISO(Q,R_i)$ of time series $\{R_1,\dots,R_n\}$ has been computed, it will be stored in *queue*. For time series S which is currently being evaluated, the triangle inequality ensures that $ISO(Q,S) \geq ISO(Q,R_i) - ISO(R_i,S)$, for all $1 \leq i \leq n$. Thus, it is necessary that

$$ISO(Q,S) \geq \max_{1\leq i\leq n} \{ISO(Q,R_i) - ISO(R_i,S)\}$$

If the calculated result *maxPruneDis* is even worse than the current k-NN distance stored in *result*, S can be skipped entirely. Otherwise, the true distance $ISO(Q,S)$ is computed, and *queue* array is updated if necessary to reflect the current k-nearest neighbors and distances in stored order.

For large databases, the algorithm 1 makes two assumptions. Firstly, the matrix *matrix* must be enough small to complete be loaded in memory. This may not be able to meet this condition for large databases. Secondly, the larger the size of *queue*, the more time series can be used for purning. In the next section, we'll make a detailed description of how to determine the specific size of *matrix* and *queue*.

Algorithm 1: $TrianglePruning(S,Q,k,queue,matrix)$

Input: $S,Q,k,queue,matrix$
Output: *result*

$maxPruneDist = 0$;
for $i = 1$ **to** *queue.length* **do**
  if $queue[i].dist - matrix[i][S] > maxPruneDist$ **then**
    $maxPruneDist = queue[i] - matrix[i][S]$;
  end
end
$best = result[k].dist$;
if $maxPruneDist \leq best$ **then**
  $dist = ISO(Q,S)$;
  insert S and $dist$ into *queue*;
  if $dist < best$ **then**
    insert S and $dist$ into *result* and sort in order to ISO distance;
  end
end

4.2. Multidimensional KNN Search

A number of similarity search speed-up techniques also use indexing structures(e.g., R-trees in [34] or sequential structures in [35]). As isomorphism distance is independent of any underlying indexing approach, it can get efficiency benefit from those data structures. Algorithm 2 shows a skeleton of the algorithm for using the B+-tree index for k-NN search. It first conduct a standard search for $\|Q\|_{ISO}$ in the *tree* which is structured in B+-tree. The search result is a leaf node L. The first k time series pointed to by L are used to initialize the *result* array. Next we make a traverse operation in the tree. All the data values bigger than $\|Q\|_{ISO}$ are visited in ascending order. Similarity, all the data values smaller than $\|Q\|_{ISO}$ are visited in descending order. If the current computed distance is smaller than the best one stored, the *queue* will be updated if necessary. Otherwise, the remaining data values can be skipped entirely.

5. Experiments

In this section, we verify the validity of the proposed approach with a comprehensive set of experiments. All experiments were executed on AMD Athlon 64 PC 3600+ (2.09GHz), 1GB memory size, CentOS 6.4 operating system and running JAVA implementations. We used Euclidean distance, DTW, DTW with anticipatory pruning (AP_DTW) [36] as well as isomorphism distance to measure the similarity search's time cost and effect. If using Euclidean distance, we can take advantage of the multidimensional space index[5,37] to speed up the search. Because the DTW distance does not meet the triangle inequality, it is not possible to use a similar indexing techniques. Here we use sequential scan and sliding window to match the subsequence. The

Algorithm 2: $KNNSearch(Q,k,tree)$

Input: $Q,k,tree$
Output: $result$

conduct a standard B+-tree search on $tree$ using $\|Q\|_{ISO}$ and let L be the leaf node which the search ends up with;
pick the first k time series as $init_q$ to which are pointed by L and initialize $result$ with those sequences' isomorphism distance;
let $v_1,\ldots,v_h$ be the data values in all leaf nodes larger than $init_q$. $v_1,\ldots,v_h$ are sorted in ascending order;
initialize $queue$,$matrix$;
for $i=1$ **to** h **do**
    pick all l time series as array S to which are pointed by v_i;
    TrianglePruning($S,Q,k,queue,matrix$);
end
let $w_1,\ldots,w_j$ be the data values in all leaf nodes larger than $init_q$. $w_1,\ldots,w_h$ are sorted in ascending order;
for $i=1$ **to** j **do**
    $best=result[k].dist$;
    if $(w_i-\|Q\|_{ISO}>best)$ **then**
        pick all l time series as array S to which are pointed by w_i;
        for $j=1$ **to** l **do**
            $dist=ISO(Q,S[j])$;
            if $dist<best$ **then**
                insert $S[j]$ and $dist$ into result that is sorted in descending order of isomorphism distance;
                $best=result[k].dist$
            end
        end
    end
    else
        break;
    end
end
return $result$;

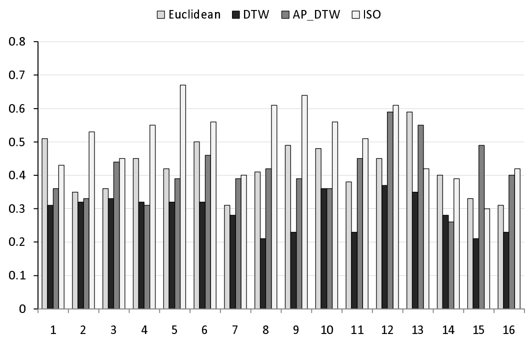


Figure 2 The Mean value of P(pruning power) for the four distance under consideration for 16 data sets when k=1.

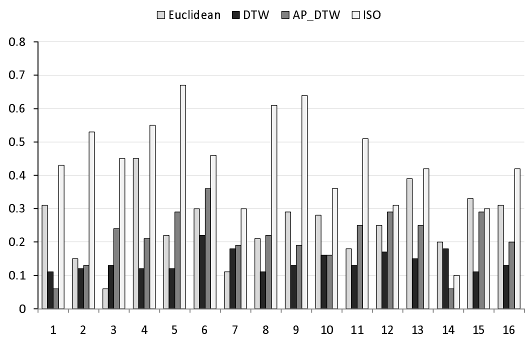


Figure 3 The Mean value of P(pruning power) for the four distance under consideration for 16 data sets when k=5.

experimental data is an earthquake data sets provided by eamonn(http://www.stat.pitt.edu/stoffer/tsa3/).

5.1. Pruning Power

Given a k-NN query Q, the pruning power is defined to be the fraction of the time series S in the data set that can be skipped. Follow [34,38], we measure pruning power(P) because this is an indicator nothing to do with implementation details. To compare the pruning power of those four distance under consideration, we measure P as follow:

$$P=\frac{N_{skipped}}{N_{all}} \tag{17}$$

The results shows the pruning power of Euclidean distance, AP_DTW and isomorphism distance on the 16 benchmark data sets for k =1,5,20. All metric distance such as Euclidean and isomorphism distance is more efficient at pruning than other DTW-like algorithm. This is due to the DTW-like distances don't meet the triangle inequality. On average, it was able to prune 1.31 times when k=1, 1.95 times when k=5 and 1.96 times when k=20. Once again, however the most obvious result is the dominance of ISO distance. It wins on most data sets and is able to prune 1.39 times as many items as Euclidean, 2.52 times as many items as DTW and 1.59 times as many items as AP_DTW.

5.2. Database Size

In order to verify the algorithm scalability on massive data sets, we'll expect the fraction of pruned sequences to increase on larger data sets. The reason is because the larger the data set, the greater the chance there is of a good match being found, and we are able to prune a larger fraction of the data. To demonstrate this effect, we run the same experiment above on increasingly larger time series data set. The results are shown in Figure 5.

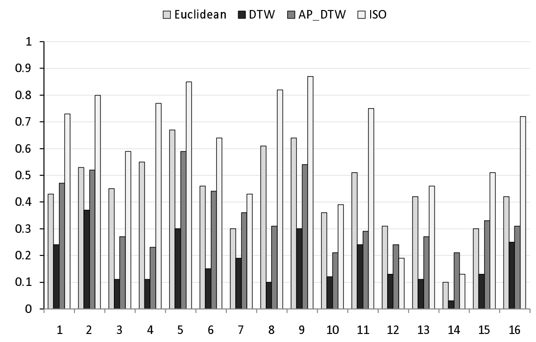


Figure 4 The Mean value of *P*(pruning power) for the four distance under consideration for 16 data sets when k=20.

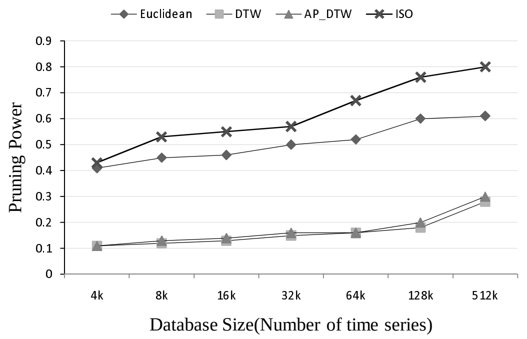


Figure 5 The effect of database size on pruning power.

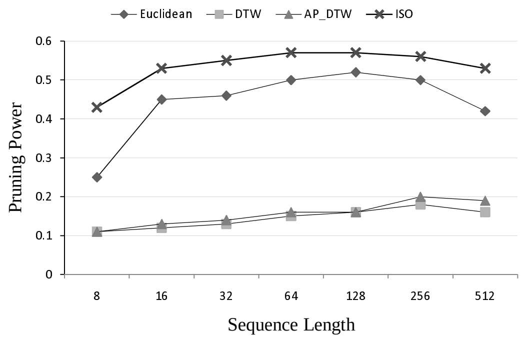


Figure 6 The effect of length of time series on pruning power.

5.3. Length of Time Series

Our next study empirically validates the scalability of isomorphism pruning with respect to the length of time series. Figure 6 shows algorithm's performance gains scale very well with the length of the time series.

5.4. Number of Nearest Neighbors

During the experiment, we also evaluate the number of nearest neighbors which have a influence on pruning. The facts show that, with the increase of parameters k values, the pruning effect of the algorithm is not a significant drop. This result is consistent with the performance of another metric distance, Euclidean distance.

6. Conclusion and Outlook

Distance measure between time series is the basis for further study of the time series data mining tasks. Looking for a good distance measure has a crucial importance for improving the efficiency and accuracy of these data mining tasks. We propose a isomorphism distance measure which can remain the geometric features in high-dimensional space to study the similarity of time series. Our approach is particularly attractive because it is a true metric distance in similarity search. Be compare with Dynamic time warping distance, our approach does not degrade performance, at the same time, the search pruning effect is greatly improved. And compared to the Euclidean distance, our method can better describe the geometric shape of high-dimensional space time series. In the future work, we will attempt to explain the geometric meaning of the time series low-dimensional manifold, research effectively isomorphic transform, compare with other similar distance, and extend it to the multivariate time sequence flow.

Acknowledgement

The author is grateful to the anonymous referee for a careful checking of the details and for helpful comments that improved this paper.

References

[1] C. Sophocleous and P.G.L. Leach, Algebraic Aspects of Evolution Partial Differential Equations Arising in Financial Mathematics, Applied Mathematics & Information Sciences, **4(3)**, (2010), 289-305.

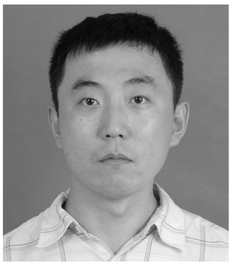
[2] H.L. Wong and J.M. Shiu, Comparisons of Fuzzy Time Series and Hybrid Grey Model for Non-stationary Data Forecasting, Applied Mathematics & Information Sciences, **4**, (2012), 409-416.

[3] M. Keyanpour and T. Akbarian, Optimal control of fredholm integral equations, Applied Mathematics & Information Sciences, **5(3)**, (2011), 514-524.

[4] R. Agrawal, C. Faloutsos and A. Swami, Efficient similarity search in sequences database, In: Proc. of the 4th Int'l Conf. on Foundations of Data Organization and Algorithms, BL. David(Ed.), (1993), 69-84.

[5] C. Faloutsos, M. Ranganathan and Y. Manolopoulos, Fast subsequence matching in time-series databases, In: Proc. of the ACM SIGMOD Conf., (1994), 419-429.
[6] Y. Sakurai, C. Faloutsos and M. Yamamuro, Stream monitoring under the time warping distance, In: Proc. of the IEEE 23rd Int'l Conf. on Data Engineering, IEEE Computer Society, (2007), 1046-1055.
[7] V. Athitsos, P. Papapetrou and M. Potamias, Approximate embedding-based subsequence matching of time series, In: Proc. of the ACM SIGMOD Int'l Conf. on Management of Data, (2008), 365-378.
[8] M. Shih, S. Hsiao and F. Chen, The association of stock index among the market of China, US and Japan, In: Proc. of the Int'l Conf. on Convergence Information Technology, (2007), 2276-2285.
[9] W. Cheng, X. Xu, J. Yan and P. Zou, Network Dynamic Risk Assessment Based on the Threat Stream Analysis, In: the 9th Int'l Conf. on Web-Age Information Management, (2008), 532-538.
[10] A. Razani and M. Samanipour, Common Fixed Point Theorems for Families of Weakly Compatible Maps in 2-Metric Spaces, Applied Mathematics & Information Sciences, **2(3)**, (2008), 275-289.
[11] E. Karapinar, I.M. Erhan and A.Y. Ulus, Fixed Point Theorem for Cyclic Maps on Partial Metric Spaces, Applied Mathematics & Information Sciences, **6(2)**, (2012), 239-244.
[12] R. Weber, H-J. Schek and S. Blott, A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces, In Proc. of 24th Int'l Conf. on Very Large Data Bases, (1998), 194-205.
[13] K.P. Chan, A. Fu and C. Yu, Haar wavelets for efficient similarity search of time-series: with and without time warping, In: IEEE Transactions on Knowledge and Data Engineering, **15**, 3 (2003), 686-705.
[14] T. Kahveei and A. Singh, Variable length queries for time series data, In: Proc. of 17th Intnational Conf. on Data Engineering, (2001), 273-282.
[15] F. Korn, H. Jagadish and C. Faloutsos, Efficiently supporting ad hoc queries in large datasets of time sequences, In: Proc. of the ACM SIGMOD Conf. on Management of Data, J. Peckham(Ed.), (1997), 289-300.
[16] H. Huang, F.L. Xiong, X.S. Hang and K. Huang, Research on a fast retrieval of similarity patterns in a time series database, Journal of Pattern Recognition and Artificial Intelligence, **16(2)**, (2003),169-173.
[17] D. Toshniwal and R.C. Joshi, Similarity search in time series data using time weighted slopes, Informatica, **29(1)**, (2005), 79-88.
[18] D.J. Berndt and J. Clifford, Finding patterns in time series: A dynamic programming approach, Advances in Knowledge Discovery and Data Mining, (1996), 229-248.
[19] G. Das, D. Gunopulos and H. Mannila, Finding similar time series, In: Proc. of the 1st European Symposium on Principles of Data Mining and Knowledge Discovery, (1997), 88-100.
[20] M. Vlachos, G. Kollios and D. Gunopulos, Discovering similar multidimensional trajectories, In: Proc. 18th Int'l Conf. on Data Engineering, (2002), 673-684.
[21] L. Chen, M.T. Özsu and V. Oria, Robust and efficient similarity search for moving object trajectories. In: CS Tech. Report. CS-2003-30, School of Compter Science, University of Waterloo.
[22] T. Bozkaya and M. Ozsoyoglu, Indexing large metric spaces for similarity search queries, ACM Transactions Database System, **24(3)**, (1999), 361-404.
[23] P. Ciaccia, M. Patella and P. Zezula, M-tree: An efficient access method for similarity search in metric spaces, In Proc. of 23th Int'l Conf. on Very Large Databases, (1997), 426-435.
[24] G. Navarro, Searching in metric spaces by spatial approximation, The VLDB Journal, **11**, (2002), 28-46.
[25] R.F.S. Filho, A.J.M. Traina, C. Traina Jr and C. Faloutsos, Similarity search without tears: The OMNI family of all-purpose access methods. In Proc. of 17th Int'l Conf. on Data Engineering, (2001), 623-630.
[26] E. Chv́ez, G. Navarro, R. Baeza-Yates and JL. Marroquń, Searching in metric spaces, ACM Computing Surveys, **33(3)**, (2001), 273-321.
[27] M.S. Waterman, T.F. Smith and W.A. Beyer, Some biological sequence metrics, Advances in Mathematics, **20**, (1976), 367-387.
[28] C. Xiao, W. Wang and X.M. Lin, Ed-Join: An efficient algorithm for similarity joins with edit distance constraints, In: Proc. of the 34th Int'l. Conf. on Very Large Data Bases(VLDB), (2008), 933-944
[29] X.D. Zhang, Matrix Analysis and Applications, Tsinghua University Press, (2004), 55.
[30] F.X. Song, S.H. Liu and J.Y. Yang, Orthogonalized fisher discriminant, Pattern Recognition, **38(2)**, (2005), 311-313.
[31] L. Zhu and S.N. Zhu, Face recognition based on orthogonal discriminant locality preserving projections, Neurocomputing, **70(7)**, (2007), 1543-1546.
[32] E. Keogh, K. Chakrabarti, M. Pazzani and S. Mehrotra, Locally adaptive dimensionality reduction for indexing large time series databases. In: Proc. of the ACM SIGMOD Conf. on management of data, (2001), 151-162.
[33] B.K. Yi and C. Faloutsos, Fast time sequence indexing for arbitrary L-p norms, In: Proc. of the 26th Int'l Conf. on very large databases, (2000), 385-394.
[34] E. Keogh, Exact indexing of dynamic time warping, In VLDB, (2002), 406-417.
[35] Y. Sakurai, M. Yoshikawa and C. Faloutsos, FTW: fast similarity search under the time warping distance, In PODS, (2005), 326-337.
[36] I. Assent, M. Wichterich, R. Krieger, H. Kremer and T. Seidl, Anticipatory DTW for efficient similarity search in time series databases, Journal Proc. of the VLDB Endowment, **2(1)**, (2009), 826-837.
[37] E. Keogh, M. Pazzani, An Indexing scheme for fast similarity search in large time series database. In Proceeding of the 11th Int'l Conf. on Scientific and Statistical Database Management, Los Alamitos, CA: IEEE Computer Society Press, (1999), 56-67.
[38] Y. Zhu and D. Shasha, Warping indexes with envelope transforms for query by humming. In Proc. ACM SIGMOD Int. Conf. on Management of Data, (2003), 181-192.
[39] S.A. Manavski and G. Valle, CUDA compatible GPU cards as efficient hardware accelerators for Smith-Waterman sequence alignment, BMC Bioinformatics, **9**, (2008).
[40] K. Jiang, O. Thosen and A. Peters, An efficient parallel implementation of the hidden Markov methods for genomic sequence search on a massively parallel system, IEEE Trans on Parallel and Distributed System, **19(1)**, (2008), 15-23.

[41] V. Athitsos, M. Potamias and P. Papapetrou, Nearest neighbor retrieval using distance-based hashing, In: Proc. of the 24th Int'l Conf. on Data Engineeriing, (2008), 327-336.
[42] A. Sacan and L.H. Toroslu, Approximate similarity search in genomic sequence databasee using landmark-guided embedding, In: Proc. of the 24th Int'l Conf. on Data Engineeriing Workshops, (2008), 338-345.
[43] X.C. Yang, B. Wang and C. Li, Cost-based variable-length-gram selection for string collections to support approximate queries efficiently, In: Proc. of the ACM SIGMOD Int'l Conf. on Management of Data, (2008), 353-364.
[44] C. Li, B. Wang and X.C. Yang, VGRAM: Inproving performance of approximate queries on string collections using variable-length grams, In: Proc. of the 33rd Int'l Conf. on Very Large Data Bases(VLDB), (2007), 303-314.
[45] C. Li, J.H. Lu and M.Y. Lu, Efficient merging and filtering algorithms for approximate string searches, In: Proc. of the 24th Conf. on Data Engineering, (2008), 257-266



Guo Wensheng was born in 1977. He is a lecturer of computer science at the China University of Petroleum, Beijing. He received his Ph.D. degree in University of Science and Technology Beijing in 2007. His research interests are in machine learning ,data mining and natural language processing.

Ji Lianen is an associate professor of computer science at the China University of Petroleum, Beijing. His research interests are in virtual reality and scientific visualization, human-computer interaction and software engineering.