Data Series Similarity Using Correlation-Aware Measures

Katsiaryna Mirylenka*
IBM Research - Zurich
Zurich, Switzerland
kmi@zurich.ibm.com

Michele Dallachiesa
Skysense
Trento, Italy
michele.dallachiesa@gmail.com

Themis Palpanas
Paris Descartes University
Paris, France
themis@mi.parisdescartes.fr

ABSTRACT

The increased availability of unprecedented amounts of sequential data (generated by Internet-of-Things, as well as scientific applications) has led in the past few years to a renewed interest and attention to the field of data series processing and analysis. Data series collections are processed and analyzed using a large variety of techniques, most of which are based on the computation of some distance function. In this study, we revisit this basic operation of data series distance calculation. We observe that the popular distance measures are oblivious to the correlations inherent in neighboring values in a data series. Therefore, we evaluate the plausibility and benefit of incorporating into the distance function measures of correlation, which enable us to capture the associations among neighboring values in the sequence. We propose four such measures, inspired by statistical and probabilistic approaches, which can effectively model these correlations. We analytically and experimentally demonstrate the benefits of the new measures using the 1NN classification task, and discuss the lessons learned. Finally, we propose future research directions for enabling the proposed measures to be used in practice.

CCS CONCEPTS

•Mathematics of computing → Time series analysis; •Information systems → Similarity measures; •Theory of computation → *Data modeling; Random walks and Markov chains; Nearest neighbor algorithms;*

KEYWORDS

data series, time series, sequences, similarity search, distance measure

ACM Reference format:
Katsiaryna Mirylenka, Michele Dallachiesa, and Themis Palpanas. 2017. Data Series Similarity Using Correlation-Aware Measures. In *Proceedings of SSDBM '17, Chicago, IL, USA, June 27-29, 2017,* 12 pages.

*Work done while at the University of Trento, Italy.

SSDBM '17, Chicago, IL, USA
 978-1-4503-5282-6/17/06...$15.00
DOI: http://dx.doi.org/10.1145/3085504.3085515

DOI: http://dx.doi.org/10.1145/3085504.3085515

1 INTRODUCTION

Data series (i.e., ordered sequences of values) [1] have gathered the attention of the data management community for almost two decades [7, 28, 36, 42, 47]. Data series are one of the most common types of data, and are present in virtually every scientific and social domain: they appear as audio sequences [24], shape and image data [45], financial [41], telecommunications [29, 38], environmental monitoring [40] and scientific data [1, 21], and they have many diverse applications, such as in health care, astronomy, biology, economics, and others.

The field of data series processing has seen a tremendous progress in the database community thanks to the increased availability of an unprecedented amount of data. Recent advances in sensing, networking, data processing and storage technologies have significantly eased the process of generating and collecting tremendous amounts of data series at extremely high rates and volumes. It is not unusual for applications to involve numbers of sequences in the order of hundreds of millions to billions [1, 2, 8, 29, 35, 39, 46].

These data series collections are then processed and analyzed using a large variety of techniques. Examples of such analysis operations are queries by content (range and similarity queries, nearest neighbors), clustering, classification, outlier patterns, frequent sub-sequences, and others. In this context, the nearest neighbor operation is of paramount importance, since it forms the basis of virtually every analysis technique. Any data series complex analysis task can be reduced to modeling a distance measure that captures the most discriminating features across different classes or patterns in the data [26].

The most widely used distance models are variations of the Euclidean distance and are characterized by the invariant properties that they support. For example, the Dynamic Time Warping (DTW) distance [5] allows accelerations and decelerations of the signal along the x-axis, and the Longest Common Subsequence (LCSS) distance [14] allows gaps in the sequence. The same applies to normalization [39] that makes any subsequent analysis invariant to shifts and scaling of the series. This invariance proved to be a very effective way to extract strong features.

[1] *Time series* are a special case of data series, where the values are measured over time, but a series can also be defined over other measures (e.g., angle in radial profiles in astronomy, mass in mass spectroscopy, position in genome sequences, etc.). For the rest of this paper, we are going to use the terms *data series* (or simply *series*) and *sequence* interchangeably.

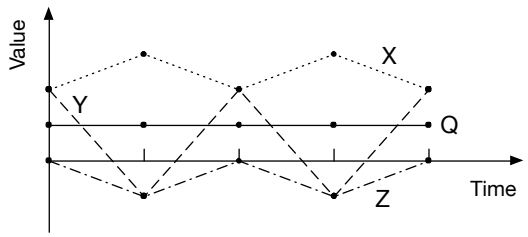


Figure 1: Euclidean distance fails to distinguish between the X and Y series, given query Q.

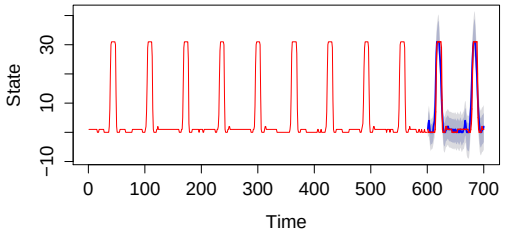
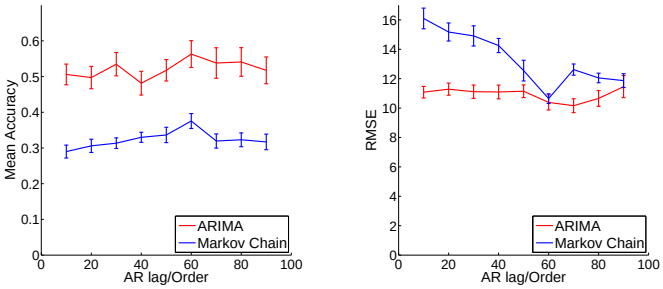


Figure 2: Predicting ECG Signal using an ARIMA model (with AR lag = 60, which is very close to the period of the signal).



(a) (b)

Figure 3: Effect of AR-Lag/Order on ARIMA/Markov Chain models in terms of (a) prediction accuracy and (b) root mean squared error. Results are averaged with 20-fold cross-validation.

The Euclidean distance is widely used, and has been shown to be very effective for large data collections, performing equally well or outperforming new distance models (such as SpADe and TQuEST), as well as traditional elastic distance measures (such as DTW) [16], especially when dealing with large data series collections. Therefore, in this work we will concentrate on Euclidean distance.

We observe that the distance measures mentioned above do not model the correlations that do exist among neighboring points in the series. Nevertheless, previous work has shown that modeling explicitly the correlation inherent in the data series leads to better results [11–13, 31, 32]. An example is illustrated in Figure 1. The graph shows four series, namely X, Y, Z and Q. The point values of the series are the following: $X =< 2, 3, 2, 3, 2 >$, $Y =< 2, -1, 2, -1, 2 >$, $Z =< -1, -2, -1, -2, -1 >$ and $Q =< 1, 1, 1, 1 >$. The Euclidean distance between Q and the other series X, Y and Z is the same, $\sqrt{11}$. The series X and Z are equally similar to the series Q. Despite the larger deviations in the values of series Y, the distance between Q and Y is exactly the same. A similar result can be obtained for other Minkowski distances and their extensions, such as the DTW and LCSS distances, as well as for z-normalized series. (A formal discussion of the invariant properties of these distance measures is presented in Section 3.1).

In this study, we answer the following question: can distance measures that take into account the neighboring-point correlations in the series outperform the Euclidean distance in mining tasks such as classification? As we will see, the answer to this question is *yes*.

In order to demonstrate the importance of taking into account the temporal correlation among values in a data series, we use a simple example from the biomedical domain. Our dataset represents thirty minutes of ECG annotated readings for forty-eight individuals [34]. The series has been discretized into thirty-two states using the iSAX symbolization algorithm [7]. First, we estimate the Auto-Regressive Integrated Moving Average (ARIMA) model, using the training part of ECG dataset. The modeled series together with true observations are shown in Figure 2. Figure 3(a) shows that the best ARIMA prediction accuracy is observed for AR lag 60 (very close to the period: 64), which also the value where the Root Mean Squared Error (RMSE) of prediction is minimum (Figure 3(b)). In addition, we tried a Markov chain model with 9 states, which was also used to estimate the ECG data. We observed that as the chain order increases up to 60, the model becomes more accurate in predicting the series values (refer to Figures 3(a) and (b)).

The above example shows that knowledge of the correlation among the values of a series can be exploited to improve effectiveness. While prediction is not the aim of this study, models that proved to be successful in this field can be also used to build correlation-aware distance measures.

In this work, we make the following contributions.

- We review the state of the art on similarity measures for data series, and describe the desired properties of distance measures, as well as the invariant properties of the Euclidean distance, in particular.
- We present four models inspired by statistical and probabilistic approaches that have been designed to capture the temporal correlation in data series: auto-correlation distance, Markov chain distance, local distance distribution (value-difference histograms defined over sliding windows), and probabilistic local

nearest neighbor distance (most probable nearest neighbor over sliding windows).
- We provide an analytical discussion of the proposed approaches, and complement that with a detailed experimental evaluation using real datasets.

The rest of this paper is organized as follows. We discuss related work in Section 2. Preliminaries and properties of common distance measures are introduced in Section 3. In Section 4, we describe the proposed distance measures, and in Section 5 we explain how they can be used. We present our experimental evaluation in Section 6, and we conclude in Section 7.

2 RELATED WORK

The class of Minkowski distance measures include the Manhattan (L_1), Euclidean (L_2) and Supremum (L_∞) distances. These distance measures are also referred to as L_p norms. Elastic distance measures based on the Euclidean distance include the Dynamic Time Warping (DTW) distance [5] and the class of edit distance measures. The DTW distance is invariant to temporal shifts by allowing accelerations and decelerations of the signal along the x-axis. The class of edit distances is inspired by the edit distance on strings and measure the minimum number of operations required to transform one series into the other. It includes the Longest Common Subsequence (LCSS) distance [14] and the Edit Distance on Real sequences (EDR) [9]. Novel distance models include the Spatial Assembling Distance (SpADe [10]) and threshold-based distance (TQuEST) [4]. A comprehensive discussion of distance measures for time series is included in [44].

An experimental comparison of the aforementioned techniques can be found in [16], which concludes that SpADe and TQuEST models are in general inferior to elastic distance measures. Moreover, there is nearly no difference between the elastic distance measures and the Euclidean distance on the time series classification task, when the size of the training dataset is large. For this reason, we have chosen the Euclidean distance to compare to.

Normalization [39] of the values along the series is usually applied as a standard preprocessing step, thus making any subsequent distance computation invariant to value shifts and amplitude changes. Invariant properties in the distance measure proved to be a good strategy in order to capture strong features, and model the underlying dynamics of the data series.

We observe that in the aforementioned methods there is no explicit modeling of time correlation, a well known property of data series. Previous studies have shown experimentally that explicit modeling of the local correlations inherent in data series is advantageous [12]. We argue that modeling the correlation within a distance measures for data series can provide added value.

An approach that uses correlation coefficients to track local correlations among data streams has been studied in the literature [37]. Data streams are processed using sliding windows, and a new score measurement is produced as the windows move forward. The LoCo correlation score is defined in this work as the difference of the auto-covariance matrices of the data points in the two windows using their eigen decomposition. We do not consider this approach in our setting, since we do not use streaming series. Though, we consider similar window-based and correlation-based approaches described in Sections 4.3 and 4.4.

There is a class of distance measures, which relate to the generation model of data series. One such distance measure uses cepstral coefficients, which are sensitive to the position of the autoregressive coefficients [23]. This measure was used on the task of clustering, and outperformed measures based on various representations of time series, namely Discrete Fourier Transform [3], which preserves the Euclidean distance; Discrete Wavelet Transform [43], which keeps local correlations of the data points; and Principal Component Analysis [19], where the Euclidean distance was calculated among principal components of time series. The results show that comparison of the statistical generation models of time series leads to more accurate results in cases where the data follow these models accurately. Put in other words, the time series used with an ARIMA-based similarity measure should be ARIMA-like [23]. We observe that the usage of cepstral coefficients is rather involved. For example, it requires that after a certain level of integration, the data series is stationary, while seasonal and trend components should be considered separately. Inspired by this approach, we use an autocorrelation distance measure, which is similar to cepstral coefficients for the autoregressive model, but is much more lightweight in terms of computation. We describe this approach in Section 4.1.

PLiF (Parsimonious Linear Fingerprinting) is a similarity measure that contains feature extraction and forecasting components [27]. This is a model-based distance measure, which estimates a Linear Dynamical System (LDS) to model multiple data series. Time correlations in PLiF are taken into account via hidden features that are used to build a distance function between data series. This model is designed specifically for motion sequences.

Ge and Smyth describe a model-based technique, where a segmental Hidden Markov Model (HMM) is used for pattern representation in order to solve a problem of waveform recognition [20]. This study shows that distance measures like DTW and Euclidean distance in several cases lack the desired properties of continuity, compactness and true representation (see Section 3.1). This means that small variations in the values of the two series may lead to large distance differences, while large variations in the series may lead to small distance differences. The approach described in [20] is based on probabilistic generative models that try to capture the probability with which a series is generated. In this approach, it is necessary to set the number of hidden and visible states, as well as the distribution of the duration of each state. Note also that segmental HMM considers only the first order dependencies between states, while our goal is to exploit high order dependencies. Consequently, we propose distance

measure based on a Markov chain model that can take high order dependencies into account, and where only number of states and order should be decided (see Section 4.2).

3 NEED FOR A NEW DISTANCE

A data series X is a sequence of real valued points $X = \{x_i\}_{i=1}^n$ where n is the length of X, and x_i is the value of data series X at position i. A data series is *z-normalized* (or simply *normalized*)if its mean is equal zero and its variance is equal to one. The Euclidean distance between data series X and Y is defined as follows: $D_{Eucl}(X,Y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$. Though it is very efficient in many applications, euclidean and euclidean-like distances cannot capture correlations among neighboring data points in the sequence.

We note that the Euclidean distance between two series X and Y, formally denoted by $D_{Eucl}(X,Y)$, is invariant to two transform rules as defined below. First, a pair of corresponding points x_i and y_i can be swapped with any other pair of points x_j and y_j, $i \neq j$ without any change in the distance value. For example, the Euclidean distance between series $X =< 1,2,3,4 >$ and $Q =< 5,6,7,8 >$ does not change if we swap the second and the fourth values (obtaining series $X' =< 1,4,3,2 >$ and $Q' =< 5,8,7,6 >$, respectively), though the new series are obviously not the same.

Second, the value of the Euclidean distance does not change when new values x'_i and x'_j are assigned respectively to points x_i and x_j, where x'_i and x'_j satisfy the following condition:

$$(x_i - y_i)^2 + (x_j - y_j)^2 = (x'_i - y_i)^2 + (x'_j - y_j)^2$$

Consider for instance, the Euclidean distance between series $Q = \{0,0,0,0\}$ and $X = \{5,5,5,5\}$ is the same to the Euclidean distance between the series Q and $Y = \{4.3563, 5.5698, 4.3563, 5.5698\}$. The Euclidean norm distance for both pairs Q, X and Q, Y is 10, while the shape of the series is drastically different.

We conclude that Euclidean distance fails to capture important semantics of data series, as shown in the above examples. In contrast, the correlation-aware distance measures presented in Section 4 aim to reveal such differences.

3.1 Properties of a Good Distance Measure

We now consider the important properties of distance measures. In general, a distance measure, or a dissimilarity representation, is a way to generalize the structural representation of an object. A distance measure is a good representation of objects if similar objects are close and dissimilar objects are distant. An ideal distance measure $D(x,y)$ between two objects x and y should have the following properties (where δ is a small positive number):

(1) *Non-negativity*: $D(x,y) \geq 0$;
(2) *Coincidence axiom*: $D(x,y) = 0$ iff $x = y$;
(3) *Symmetry*: $D(x,y) = D(x,y)$;
(4) *Triangle inequality*: $D(x,y) < D(x,z) + D(z,y)$, where z is another object;
(5) *Compactness*: if x and y are very similar then $D(x,y) < \delta$;
(6) *True representation*: if $D(x,y) < \delta$ then z and y are very similar;
(7) *Continuity* of D.

Properties 5, 6 and 7 were introduced several times in the pattern recognition literature. They force the preferred distance measure to have a large discriminative power, and at the same time the value of the distance measure should increase gradually [17], which means that the continuity property holds.

Metric distances satisfy the first four properties, which also enables them to serve as the basis of indexing techniques. At the same time, there are many cases when non-metric distance measures lead to much higher accuracy for real-world classification problems, such as the classification in the presence of partially occluded objects in computer vision [22], or the classification of proteins, when structural alignment is needed [18]. For this reason, in our study we also consider non-metric distances.

In this study, we compare different distance measures in terms of the 1-NN classification task. For this task, good distance measures should differentiate objects from different classes — the distance among objects in the same class should be smaller than the distance among objects from different classes. Therefore, the properties of compactness, true representation and continuity are important in this context.

4 PROPOSED DISTANCE MEASURES

In this section, we introduce and describe four distance measures which take into account the correlations among neighboring points in the series.

4.1 Autocorrelation Distance (ACD)

The distance measure based on autocorrelation coefficient is not new, it comes from the statistical domain and is widely exploited in a data mining community [23]. In this work, we calculate the autocorrelation vector $R = \{r(\tau)\}_{\tau=1}^n$, which consists of autocorrelation coefficients $r(\tau)$ with different lags up to n:

$$r(\tau) = \frac{E[(x_t - \mu)(x_{t+\tau} - \mu)]}{\sigma^2},$$

μ is a mean and σ^2 is a variance of a data series $X = x_i$. The distance between two series is defined as the Euclidean distance between their autocorrelation vectors. The length of autocorrelation vector n is a training parameter.

Let n be the series length, and k the number of series used for comparison. The time complexity of the ACD measure is equal to the time complexity of the computation of Euclidean distance $O(nk)$ plus $O(nk\log(n))$ for the preprocessing step (assuming that Fast Fourier transform is used for the computation of the correlation coefficients).

4.2 Markovian Distance (MD)

Markovian models are commonly used to capture correlations among points of a data series. A Markov chain of order k is

a sequence of random variables, which satisfy the Markovian property that the current state of the chain depends only on the previous k states. A Markov chain model is fully characterized by the initial probabilities of the states and the transition probability matrix, which consists of conditional probabilities of all the values of the alphabet. To have a Markov chain model of order k and alphabet size m, we need to estimate $|A|^{k+1}$ parameters, which are the values of transition probability matrix. In our study, we consider Markov chains with alphabet size $m = 32$, and treat the order as a parameter, which we need to estimate during the training phase. For the testing phase, we estimate a transition probability matrix M, which characterizes a Markov chain by estimating the conditional probabilities of the query X. We do this by looking across the series and first calculating the frequencies of all sequences of length k and $k+1$, and then calculating all the conditional probabilities:

$$M(x_{t-k}, x_{t-k+1}..., x_t) = Pr[x_t|x_{t-1}, ..., x_{t-k}] = \frac{Freq[x_t, x_{t-1}, ..., x_{t-k}]}{Freq[x_{t-1}, ..., x_{t-k}]},$$

where $t = k+1, ..., n$, n is the length of the series. We then identify the nearest neighbor, that is, the series Y with the highest probability of being generated by the model of the query series:

$$Pr(y_1, ..., y_n|M) = Pr[y_1, ..., y_k] \prod_{t=k+1}^{n} M(y_{t-k}, ..., y_t), \quad (1)$$

where $Pr[y_1, ..., y_k]$ is the initial state of the Markov chain.

Since we estimate the model using only one series, we cannot estimate the initial state, so we set all the states to be equiprobable. We can now choose the nearest neighbor only by computing the product of the entries in the transition probability matrix. Note, that the entries, which correspond to the same prefix $[x_{t-1}, ..., x_{t-k}]$, form a discrete probability density function, whose total mass should be 1. If a prefix was not observed in the query time series, we distribute the probability mass among all the entries equally, for each entry it is $1/m$. In order to avoid the accumulation of machine error caused by the multiplications in Equation 1, we calculate the log of the probabilities:

$$\log Pr(y_1, ..., y_n|M) \sim \sum_{t=k+1}^{n} \log[M(y_{t-k}, y_{t-k+1}..., y_t)]. \quad (2)$$

This leads to a natural distance measure, which is a probability that one sequence is generated using a model of another sequence. As $\log Pr$ defined by Equation 2 is a similarity measure, the distance between X and Y can be defined as $-\log Pr$.

This distance measure satisfies the properties of non-negativity, continuity (refer to Section 3.1), as well as compactness and true representation (assuming that two series are similar if they have similar generation models). The ACD measure captures similar phenomena as the MD, but it focuses on the correlations among values, rather than on the states of the series. This makes ACD more suitable for real-valued series with high correlations among all values up to n, while MD is preferable when the most significant correlation is equal to the order k.

The time complexity of MD is linear, $O(n + (m - k))$, as it requires only one scan over the query series of length n to build the model, and $(m - k)$ searches in a transition probability matrix, which can be implemented as a HashMap with average search time $O(1)$ (m is the length of the series we compare to, and k is the order of the Markov chain model). Note that this distance can also be efficiently computed in an online setting, where streaming series for very large alphabet sizes should be compared, using Conditional Heavy Hitters [30, 33] for estimating the most significant elements of the transition probability matrix.

4.3 Local Distance Distribution (LDD)

In this section, we propose the Local Distance Distribution (LDD), a ranking function that is based on the distribution of Euclidean distances determined on sub-sequences from candidate series X_i and query Q.

Given a series X_i, let $X_i^{[a,b]}$ be the sub-sequence of X_i between positions a and b. Let $W_h(X_i, w)$ be the content of the sliding window on series X_i of length w whose first point is x_h, i.e., $W_h(X_i, w) =< x_h, ..., x_{h+w-1} >$. The set of distance samples between X_i and Q is denoted by $D(Q, X_i)$ and is defined as:

$$D(Q, X_i) = \{Euclidean(W_h(X_i, w), W_h(Q, w)) : h \in \{1, ..., n - w + 1\}\},$$

where $Euclidean(X_i, X_j)$ denotes the Euclidean distance between series X_i and X_j and n is the length of the series. $D(Q, X_i)$ is a set of pairwise point distances along the series Q and X_i. Let H_i be the equi-width histogram composed of B buckets that summarizes the distance values in $D(Q, X_i)$.

Given two series X_i and X_j, the probability that a random distance value $d_i \in D(Q, X_i)$ is lower than a random distance value $d_j \in D(Q, X_j)$ can be estimated as follows:

$$Pr(d_i < d_j) = \sum_{b=1}^{B} H_{i,b} \sum_{l=b+1}^{b} H_{i,l},$$

where $H_{i,l}$ is the value of the lth bucket of the equi-width histogram H_i. We can now introduce the probability for a candidate series X_i to be the nearest neighbor to a query series Q as:

$$PNN(X_i, Q) = \prod_{j \neq i} Pr(d_i < d_j), \quad (3)$$

where d_i and d_j are two random distance values from $D(Q, X_i)$ and $D(Q, X_j)$, respectively. The function $PNN(X_i, Q)$ is a ranking function that can be used to implement a nearest neighbor classifier. We observe that the LDD ranking function is not a distance measure, but incorporates pairwise point distance values.

The LDD ranking function is invariant to swaps of pairs of values in the two series to be compared. Any modification in the series values can affect the overall pairwise point distance distribution $D(Q, X_i)$. We observe that the good distance

measure properties (refer to Section 3.1), compactness, true representation, and continiuty, are satisfied.

The time complexity of LDD is as follows. We need to compute Euclidean distance n times (n is the length of the series) over sub-sequences of size w, with an overall cost of $O(nw)$. The computation of equi-width histograms costs $O(n)$, and histogram comparisons cost $O(BN)$. Therefore, the LDD complexity is $O(nw + BN)$.

4.4 Probabilistic Local Nearest Neighbor (PLNN)

In this section, we introduce the Probabilistic Local Nearest Neighbor (PLNN) ranking function. PLNN is inspired by the L_p-norm as follows: the Euclidean distance between neighboring points induces a series of distance distributions that capture the local semantics of the series. These distance distributions are then aggregated by taking into considering their dependencies.

Given a query Q and series X_i, let $D^h(Q, X_i)$ be the set of pairwise point distances within the sub-sequences $X_i^{[h,h+w-1]}$ and $Q^{[h,h+w-1]}$:

$$D^h(Q, X_i) = \{|x_k - q_k|\}, \tag{4}$$

where $k \in \{h, ..., h + w - 1\}$. We observe that the sum of samples in $D^h(Q, X_i)$ is equivalent to the L_1-norm (Manhattan distance) between sub-sequences $W_h(X_i, w)$ and $W_h(Q, w)$.

Let $NN^h(i)$ be the event that X_i is the nearest neighbor to query Q within the content of sub-sequences $W_h(Q, w)$ and $W_h(X_i, w)$. The probability $Pr(NN^h(i))$ is formulated as follows:

$$Pr(NN^h(i)) = Pr\left(\bigwedge_{j \neq i} Euclidean(0, d_i^h) < Euclidean(0, d_j^h)\right),$$

where d_i and d_j are two random point values from $D^h(Q, X_i)$ and $D^h(Q, X_j)$, respectively. $Pr(NN^h(i))$ can be estimated as in [6], Eq.1 under the independence assumption. $Pr(NN(i))$ is formulated as follows.

$$Pr(NN(i)) = Pr\left(\bigwedge_i NN^h(i)\right). \tag{5}$$

$P(NN(i))$ is the marginal probability of i to be the nearest neighbor to query Q. The multiplicands in Eq. 5 are not independent. However, we can estimate their joint probability considering samples in $D^h(Q, X_i)$. For ease of exposition, we discuss how to deal with dependencies using a small dataset of three time series (X_0, X_1, X_2). Eq. 5 can be decomposed as follows:

$$\begin{aligned} P(NN(i)) = P(NN^0(i)) \cdot \\ P(NN^1(i)|NN^0(i)) \cdot \\ P(NN^2(i)|NN^0(i) \wedge NN^1(i)) \end{aligned} \tag{6}$$

We already know how to estimate $P(NN^0(i))$. To estimate $P(NN^1(i)|NN^0(i))$, we proceed as follows. During the evaluation of $NN^1(i)$ we consider only a subset of the distance samples in S_0^h, such that they match the distance constraints in $NN^0(i)$ (and in addition slide the windows). We proceed similarly for the other multiplicands in Eq. 6.

Similarly to LDD, PLNN is invariant to swaps of pairs of corresponding values in the compared series. Modification in the series values may affect the overall pairwise point distance distribution $D(Q, X_i)$. The desired distance measure properties, compactness, true representation, and continuity, are satisfied (refer to Section 3.1).

The time complexity of PLNN is as follows. Evaluating Eq. 5 costs $O(w^2N)$, where w is the window length, and N is the number of series. Eq. 5 is evaluated on all possible windows in Eq. 5, resulting in $O(nNw^2)$, where n is the series length. Assuming that we look back to the previous k windows in Eq. 6, the total cost is $O(knNw^2)$.

5 USING THE PROPOSED METHODS

Using the Euclidean distance for 1NN classification leads to the fastest and simplest classification. In this work, we combine Euclidean distance with the proposed techniques for 1NN classification: when the discrimination confidence of the Euclidean distance is low, then we switch to using our techniques. In this way, we aim to combine the speed of Euclidean with the accuracy of the proposed techniques.

Given an oracle, we can choose to use our techniques only when Euclidean fails. In practice though, we have to predict when this will happen. We use the following strategy for this classification failure prediction [15]. First, we compute a confidence value based on the distances to the two nearest neighbors belonging to two different classes:

$$Conf = 1 - \frac{d_i}{min_{i \neq j} d_j}, d_j = \min\{dist(Q, X_j) | j \in C\},$$

where C denotes a class. Then, we use the proposed distance measures when this confidence value is below some threshold. Our experiments show that the accuracy of this prediction is slightly above 75%, and fairly robust for thresholds between 0.2-0.8.

6 EXPERIMENTAL EVALUATION

We performed the experiments on a computer with an Intel Core i7-3612 CPU @ 2.1GHz x 2 processor, with 8GB RAM, running MS Windows 7. All algorithms were implemented in Matlab.

Datasets: We use the 43 UCR datasets with normalized series of different lengths from several domains [25]. All the time series are normalized. (The importance of normalization and its usefulness on real datasets has been demonstrated in earlier studies [39].) Most of the datasets have multiple labels, as well as data imbalance. We use the 1NN classifier because of its simplicity, as we do not need to optimize any parameters for the classifier per se.

Measures: We compare our methods to the widely-used Euclidean distance for the 1NN classification task. For the evaluation, we use the F1-measure, because it takes into

account both precision and recall. These measures are calculated as follows:

$$precision_i = \frac{tp}{tp + fp}, i \in C \quad (7)$$

$$recall_i = \frac{tp}{tp + fn}, i \in C \quad (8)$$

where, tp, fp and fn represent true positives, false positives and false negatives respectively, and C_i is the i-th class.

$$F1 = 2 * \frac{precision * recall}{precision + recall} \quad (9)$$

Precision and recall are calculated for each class C_i, and their arithmetic mean is used to calculate the mean F1 value.

6.1 Parameter Optimization

The proposed distance methods depend on some parameters. We optimize these parameters on training data, using leave-one-out cross-validation for each training dataset in order to avoid overtraining. Below we discuss the parameters and their domains:

- ACD depends on the lag, or number of autocorrelation coefficients, which we use. We performed grid search over the of 5 to 100 coefficients, with step 2 from 5 to 30, and step 5 from 30 to 100.
- Markovian distance depends on the order of the Markov chain, and on the number of states of the series. We decide on the number of states during the pre-processing step, which generates discretized series: we use iSAX[7] and generate 32 state for each dataset. During the training phase we focus on the order of Markov chain, and search for the optimal order in the range between 3 and 30 with a step size 2.
- LDD depends on the window and bin sizes of the histograms (for simplicity, the number of bins were set to 10). During the training phase, we optimize the window size over the range 3 to 30 with a step size 2.
- PLNN depends on the window size, which is optimized in the range between 5 and 50 using a step size of 5.

6.2 Description of Experiments

We organized our evaluation along the following four experiments. These experiments correspond to the four graphs (namely, a-d) presented in the figures with the results for the proposed measures.

Experiment (a) compares the correlation-aware distances with the Euclidean distance for the 1NN classification task. This experiment also help us to establish the importance of the Euclidean distance for these tasks.

Experiment (b) compares the proposed methods with a random classifier. We use the mean of 20 fold random classifier for comparison. This comparison evaluates whether the use of a classifier based on the proposed measures has advantage over random choice.

Experiment (c) uses an oracle that switches to one of the proposed techniques when a failure caused by the classifier using the Euclidean distance occurs. Here we evaluate the improvement in the accuracy of classification which can be achieved by building a classifier, which uses both the Euclidean distance and correlation-aware distances.

Experiment (d) builds a combined classifier based on the Euclidean and correlation-aware distances. Correlation-aware distances are used when the confidence of Euclidean classification is lower than a threshold. Evidently, the results of this experiment are upper bounded by the results of Experiment 3.

6.3 Results

In the first set of experiments, we perform a sanity check by comparing the accuracy of using the proposed distance measures in a 1NN classifier, against the accuracy of a random classifier. The results, depicted in Figures 4(b), 5(b), 6(b), and 7(b), show that all three methods consistently outperform the random classifier (i.e., points above the diagonal). This is especially true for the case where (with the help of an oracle) we use the three proposed methods only when Euclidean distance fails to identify the correct class (i.e., square green points).

6.3.1 Autocorrelation Distance (ACD)

. We now focus on the performance of the ACD distance, shown in Figure 4. As mentioned in Section 4.1, the autocorrelation function is a cross-correlation of a data series with itself within a given time lag. The resulting autocorrelation vectors are then used to compute the Euclidean distance between the series.

The results of Experiment (a), depicted in Figure 4(a), show that ACD performs better than Euclidean for a few datasets. The Euclidean distance works better on average, which establishes the fact that replacing the Euclidean distance entirely will not provide any statistical advantages.

Figure 4(c) shows that switching to ACD when we know for sure that the Euclidean distance will fail leads to a remarkable improvement in accuracy. Thus, using a perfect oracle for predicting failure of Euclidean distance based classification and then switching to ACD based classification shows significant accuracy improvement for all 43 datasets.

Figure 4(d) shows that the ACD distance assisted by failure-prediction performs better than Euclidean only for some of the datasets (i.e., points above the diagonal). Failure-prediction is used in the way described in Section 5, where we predict (with a less than perfect accuracy) the cases that the Euclidean-based classification fails. A close look at the experimental results reveals that ACD significantly improves the classification accuracy for several datasets. One such dataset is Trace, for which the classification accuracy with ACD is 100%, while the Euclidean distance based classification has an accuracy of only 76%.

6.3.2 Markovian Distance

. Classification based on the proposed Markovian distance

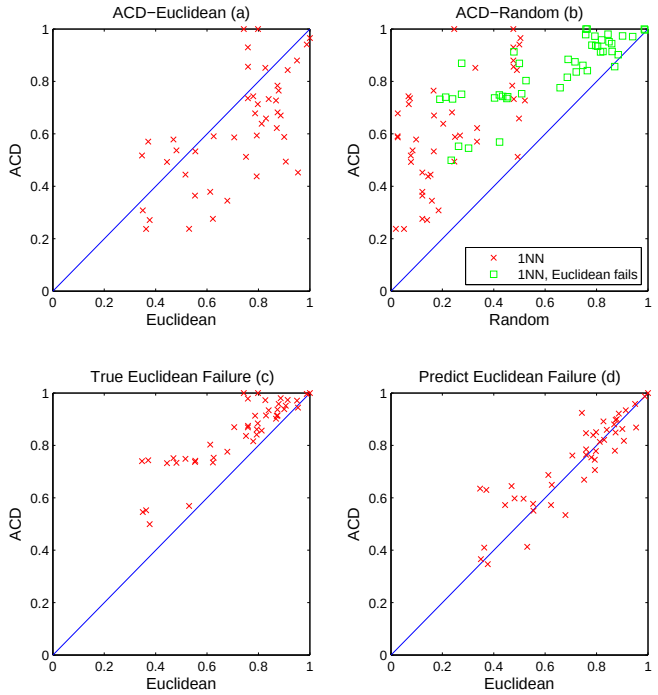


Figure 4: Results for ACD.

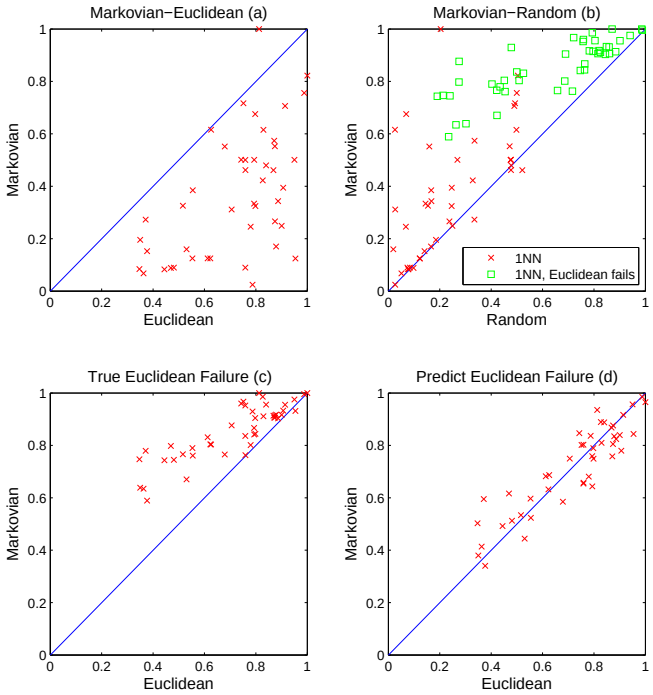


Figure 5: Results for MD (based on Jensen-Shannon divergence).

uses the transition probability matrix for each query data series in order to capture the correlation among adjacent points in the sequence. This transition probability matrix is used to find the series of the training set, which is the most likely to be generated by the query model. Since estimating the Markov model requires data series with discrete values, we used iSAX2.0 [7] to generate 32 discrete states for our data series. The experiments focus on the effect of the order of the Markov chain on classification accuracy. Our cross validation experiments showed that the transition matrix for chains of order 3 gives the best performance for most datasets (though some datasets produce better cross validation results when using different orders). Based on this, we used Markov chains of order 3 for the rest of our experiments with the Markovian distance.

Figure 5(a) shows that the Euclidean distance in almost all cases performs better than the proposed method. However, switching to the Markovian distance only when Euclidean truly fails (i.e., failure prediction with a perfect oracle) results in a significant improvement in almost all the datasets (refer to Figure 5(c)). This improvement signifies that the Markovian distance is able to capture semantics embedded in the series, which the Euclidean distance fails to uncover.

Figure 5(d) shows that the Markovian method with failure-prediction outperforms the Euclidean distance in 20 datasets.

6.3.3 Local Distance Distribution (LDD)

. We now turn our attention to the LDD distance. This method uses a series of distances calculated using a sliding window over the query series Q and each series X_i in the dataset. The distribution of the resulting sliding window based distances is represented as a histogram. We then calculate the joint probability of each X_i being the nearest neighbor (i.e., the corresponding LDD value is the smallest). Maximizing this probability gives us the most probable class C_i for a query Q. The sliding window sizes were set independently for each dataset, and were selected during the training phase by maximizing F1.

The results in Figure 6(a) show that Euclidean distance outperforms the proposed methodology, with very few exceptions. Once again, when the failure of the Euclidean distance based classifier can be perfectly predicted, then the advantage of switching to the LDD measure is significant for all datasets (see Figure 6(c)).

Figure 6(d) depicts the results of the comparison between the combination of LDD with Euclidean (i.e., LDD is used when Euclidean is predicted to fail), and Euclidean. As with the other two proposed measures, the methodology that uses the LDD distance is able to outperform Euclidean in some, but not all datasets we tested.

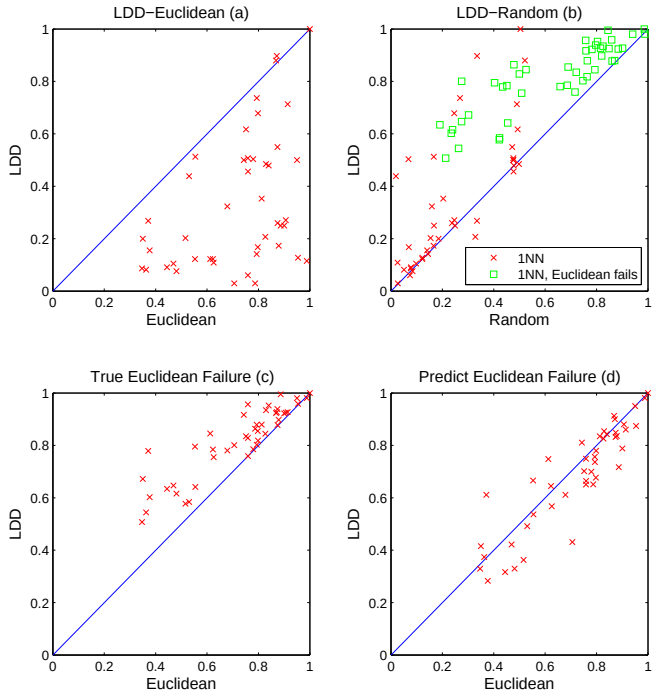


Figure 6: Results for LDD.

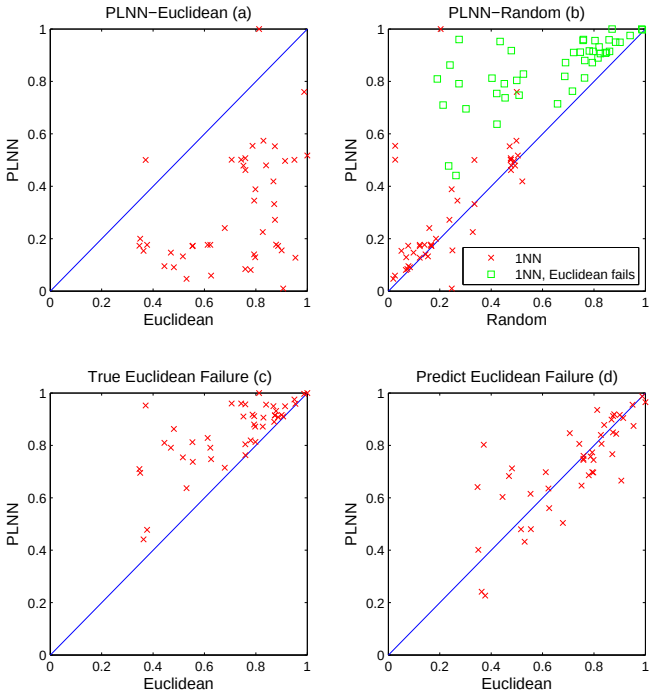


Figure 7: Results for PLNN.

6.3.4 *Probablistic Local Nearest Neighbor (PLNN)*

. PLNN employs the distance distributions across subsequences in order to capture local semantics. It examines all the data series in the collection, and identifies the series X_i that is the nearest neighbor across all subsequences.

Figure 7 provides the results for the four experiments. As we can see from the results of the first experiment (Figure 7(a)), the proposed method is unable to model the data series better than Euclidean distance for the given datasets, except for 2 datasets.

When we use PLNN on series, for which we know that Euclidean has failed, our method can lead to considerably better performance. These results are illustrated in Figure 7(c). Figure 7(d) depicts the case where we need to predict when Euclidean will fail and subsequently use PLNN. The results show that in this case PLNN achieves an average improvement of almost 9%, by providing more accurate results for nearly 60% of the cases.

6.3.5 *Time Performance*

. We now present the time performance of the proposed methods. The reported time values are the means over 5 executions of each method.

Figure 8 shows that all methods have a linear growth with respect to the series length (note that the y-axis is logarithmic). As expected, the Euclidean distance is the fastest measure, followed by ACD and MD. The LDD and PLNN measures are significantly slower. This is especially true for PLNN, which requires more than one second in order to perform a distance computation, even for moderately-sized series (i.e., 300 points long).

6.4 Discussion

The results presented in the previous sections indicate that there is a certain benefit to be gained when using correlation-aware distance measures, which can lead to a significant improvement in the quality of the results. Nevertheless, this benefit is not immediately available: for the 1NN classification task, the benefit emerges only under certain conditions.

In particular, after a careful inspection of the results and the datasets, we observed that the proposed distance measures work better only for some datasets, for which they can accurately model the inherent correlations in the sequences of values, while for other datasets they cannot provide a benefit with respect to Euclidean. Figure 9 illustrates one such example. The two graphs in this figure are the histograms of the 1NN distances for all the series in Class 1 of the FaceALL dataset. We observe that in this case, there is a clear separation between the classes, with all intra-class distances being smaller than the inter-class distances. This means that even a simple distance measure, such as Euclidean that does

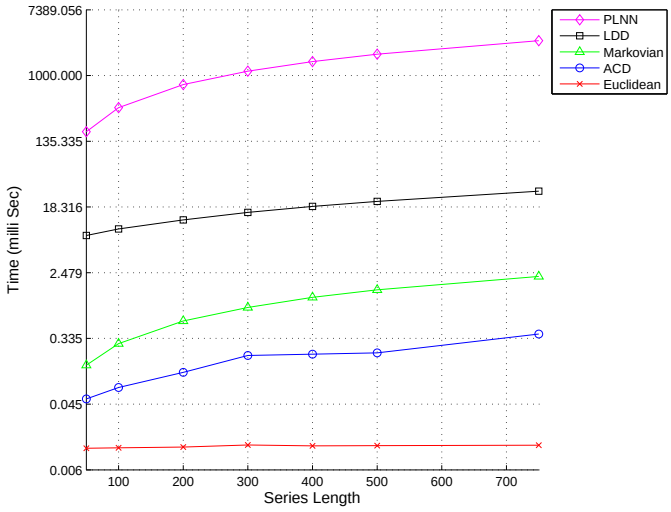


Figure 8: Comparison of running time, between Euclidean and proposed methods

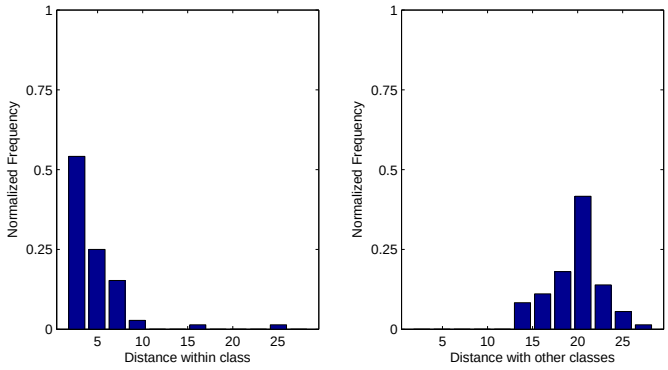


Figure 9: The 1NN histograms for intra-class (left) and inter-class distances, computed over all series in Class 1 of dataset FaceALL.

not take into account correlations, can correctly differentiate between the classes.

The benefit of correlation-aware distance measures becomes apparent in cases where the intra- and inter-class Euclidean distances are similar, but the characteristics of the data series in the various classes are different. This case is depicted in Figure 10, where we visualize several series from different classes of two distinct datasets. Even though the series in both datasets have intra- and inter-class Euclidean distances that are very close to one another, the series exhibit evolution characteristics that are similar for the same class, and distinct from the other classes. It is exactly these correlations (which determine the shape of each series) that our proposed distance measures take into consideration, and which can lead to improved results.

For example, when using ACD on the Trace dataset (Figure 10(b)), we obtain an F1 value of 100%, while the F1 for euclidean is only 80%. For the OSULeaf dataset (Figure 10(a)), even though the differences in the characteristics of the two classes are more subtle, the correlation-aware methods can enhance the performance of the 1NN classifier (when used in combination with Euclidean) by 19% for ACD, 21% for MD, 9% for LDD, and 18%(for PLNN, according to the F1-messure.

Table 1 provides the summarized comparison of the results for the 1NN classification task, where we use the proposed correlation-aware methods only when a failure is predicted in the classification done using Euclidean distance. The results are the improvement on precision and recall when compare to the case where we only use Euclidean distance.

The improvements in precision range between 12-42%, and in recall between 12-40%. These results showcase the added value that the proposed distance measures bring with them, by capturing the correlations in neighboring values of the series, which the traditional distance measures (such as Euclidean) ignore.

7 CONCLUSIONS AND FUTURE WORK

In this work, we argued about the utility of taking into account the correlations inherent among neighboring values of a sequence, when designing distance measures for data series. We proposed three different measures that are correlation aware, based on autocorrelation, Markov chains, and the subsequence distance distributions.

Our preliminary experimental results with 43 real datasets show that these more complex distance measures have the potential to compute distances more accurately, as demonstrated using the 1NN classification results. This result is explained by the fact that they can effectively encode information about the sequentiality of the points in a data series, which is completely ignored by the Euclidean distance.

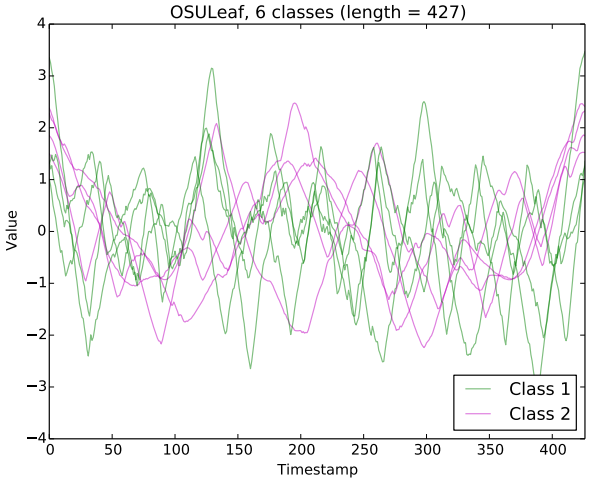
In our future work, we plan to conduct more detailed experiments for the characterization of the performance behavior of the proposed distances, as well as new ones. Moreover, we will study in depth the problem of when to use the correlation-aware measures, and how to combine them with other distance measures. This proves to be a critical step in order to exploit the benefits of the proposed distance measures.

Acknowledgements

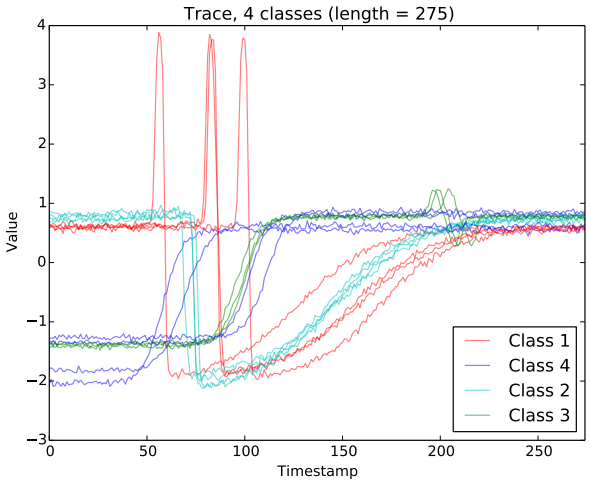
We would like to thank Muhammad Usman Akram, who contributed in the implementation and experimental evaluation of the techniques described in this paper.

	ACD	Markovian	LDD	PLNN
Improvement on Precision/Recall	12.09%/12.25%	38.14%/32.67%	41.77%/30.79%	42.13%/40.34%

Table 1: Improvement on precision and recall when combining each one of the correlation-aware methods with Euclidean.



(a) OSU Leaf



(b) Trace

Figure 10: Sample series from different classes, for two different datasets. (a) OSULeaf dataset: series form different classes have very similar characteristics. (b) Trace dataset: series form different classes have distinct characteristics.

REFERENCES

[1] Adhd-200. http://fcon_1000.projects.nitrc.org/indi/adhd200/, 2011.

[2] Sloan digital sky survey. https://www.sdss3.org/dr10/data_access/volume.php, 2015.

[3] R. Agrawal, C. Faloutsos, and A. Swami. Efficient similarity search in sequence databases. pages 69–84. Springer Verlag, 1993.

[4] J. Aßfalg, H. Kriegel, P. Kröger, P. Kunath, A. Pryakhin, and M. Renz. Similarity search on time series based on threshold queries. *Advances in Database Technology-EDBT 2006*, pages 276–294, 2006.

[5] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *AAAIWS*, pages 359–370, 1994.

[6] G. Beskales, M. Soliman, and I. IIyas. Efficient search for the top-k probable nearest neighbors in uncertain databases. *Proceedings of the VLDB Endowment*, 1(1):326–339, 2008.

[7] A. Camerra, T. Palpanas, J. Shieh, and E. Keogh. isax 2.0: Indexing and mining one billion time series. In *ICDM*, 2010.

[8] A. Camerra, J. Shieh, T. Palpanas, T. Rakthanmanon, and E. Keogh. Beyond one billion time series: indexing and mining very large time series collections with isax2+. *KAIS*, 39(1):123–151, 2014.

[9] L. Chen, M. T. Özsu, and V. Oria. Robust and fast similarity search for moving object trajectories. In *Proceedings of the ACM SIGMOD international conference on Management of data*, pages 491–502, 2005.

[10] Y. Chen, M. A. Nascimento, B. C. Ooi, and A. K. Tung. Spade: On shape-based pattern detection in streaming time series. In *International Conference on Data Engineering (ICDE).*, pages 786–795, 2007.

[11] M. Dallachiesa, B. Nushi, K. Mirylenka, and T. Palpanas. Similarity matching for uncertain time series: Analytical and experimental comparison. QUeST '11, pages 8–15. ACM, 2011.

[12] M. Dallachiesa, B. Nushi, K. Mirylenka, and T. Palpanas. Uncertain time-series similarity: return to the basics. *Proceedings of the VLDB Endowment*, 5(11):1662–1673, 2012.

[13] M. Dallachiesa, T. Palpanas, and I. F. Ilyas. Top-k nearest neighbor search in uncertain data series. *PVLDB*, 8(1):13–24, 2014.

[14] G. Das, D. Gunopulos, and H. Mannila. Pkdd. *Principles of Data Mining and Knowledge Discovery*, pages 88–100, 1997.

[15] B. Dasarathy. Nearest Unlike Neighbor (NUN): An Aid to Decision Confidence Estimation. In *Optical Engineering 34*, 1995.

[16] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh. Querying and mining of time series data:

experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment*, 1(2):1542–1552, 2008.

[17] M.-P. Dubuisson and A. Jain. A modified hausdorff distance for object matching. In *Pattern Recognition, 1994. Vol. 1 - Conference A: Computer Vision amp; Image Processing., Proceedings of the 12th IAPR International Conference on*, volume 1, pages 566–568 vol.1, 1994.

[18] R. P. W. Duin and P. Paclik. Prototype selection for dissimilarity-based classifiers. *Pattern Recognition*, 39:189–208, 2006.

[19] M. Gavrilov, D. Anguelov, P. Indyk, and R. Motwani. Mining the stock market (extended abstract): which measure is best? In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '00, pages 487–496. ACM, 2000.

[20] X. Ge and P. Smyth. Deformable markov model templates for time-series pattern matching. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '00, pages 81–90. ACM, 2000.

[21] P. Huijse, P. A. Estévez, P. Protopapas, J. C. Principe, and P. Zegers. Computational intelligence challenges and applications on large-scale astronomical time series databases. *IEEE Comp. Int. Mag.*, 9(3):27–39, 2014.

[22] D. W. Jacobs, D. Weinshall, and Y. Gdalyahu. Classification with non-metric distances: Image retrieval and class representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(6):583600, 2000.

[23] K. Kalpakis, D. Gada, and V. Puttagunta. Distance measures for effective clustering of arima time-series. In *ICDM*, pages 273–280, 2001.

[24] K. Kashino, G. Smith, and H. Murase. Time-series active search for quick retrieval of audio and video. In *ICASSP*, 1999.

[25] E. Keogh, X. Xi, L. Wei, and C. Ratanamahatana. The UCR Time Series Classification/Clustering Homepage, 2011.

[26] A. Kotsifakos, V. Athitsos, and P. Papapetrou. Query-sensitive distance measure selection for time series nearest neighbor classification. *IDA*, 20(1):5–27, 2016.

[27] L. Li, B. A. Prakash, and C. Faloutsos. Parsimonious linear fingerprinting for time series. *PVLDB*, 3(1):385–396, 2010.

[28] J. Lin, R. Khade, and Y. Li. Rotation-invariant similarity in time series using bag-of-patterns representation. *J. Intell. Inf. Syst.*, 39(2), 2012.

[29] K. Mirylenka, V. Christophides, T. Palpanas, I. Pefkianakis, and M. May. Characterizing home device usage from wireless traffic time series. In *EDBT*, pages 539–550, 2016.

[30] K. Mirylenka, G. Cormode, T. Palpanas, and D. Srivastava. Conditional heavy hitters: detecting interesting correlations in data streams. *The VLDB Journal*, 24(3):395–414, 2015.

[31] K. Mirylenka, M. Dallachiesa, and T. Palpanas. Correlation-aware distance measures for data series. In *Proceedings of the 20th International Conference on Extending Database Technology, EDBT 2017, Venice, Italy, March 21-24, 2017.*, pages 502–505, 2017.

[32] K. Mirylenka, C. Miksovic, and P. Scotton. Recurrent neural networks for modeling company-product time series. In *Proceedings of AALTD 2016: Second ECML/PKDD International Workshop on Advanced Analytics and Learning on Temporal Data*, pages 29–36, 2016.

[33] K. Mirylenka, T. Palpanas, G. Cormode, and D. Srivastava. Finding interesting correlations with conditional heavy hitters. In *ICDE*, pages 1069–1080, 2013.

[34] G. Moody and R. Mark. The impact of the mit-bih arrhythmia database. *Engineering in Medicine and Biology Magazine, IEEE*, 20(3):45 –50, may-june 2001.

[35] T. Palpanas. Data series management: The road to big sequence analytics. *SIGMOD Record*, 44(2):47–52, 2015.

[36] T. Palpanas. Big sequence management: A glimpse of the past, the present, and the future. In *International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM)*, pages 63–80, 2016.

[37] S. Papadimitriou, J. Sun, and P. S. Yu. Local correlation tracking in time series. In *ICDM*, pages 456–465, 2006.

[38] P. Paraskevopoulos, T.-C. Dinh, Z. Dashdorj, T. Palpanas, and L. Serafini. Identification and characterization of human behavior patterns from mobile phone data. In *D4D Challenge session, NetMob*, 2013.

[39] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, and E. Keogh. Searching and mining trillions of time series subsequences under dynamic time warping. In *KDD*, pages 262–270, 2012.

[40] U. Raza, A. Camerra, A. L. Murphy, T. Palpanas, and G. P. Picco. Practical data prediction for real-world wireless sensor networks. *IEEE Trans. Knowl. Data Eng.*, accepted for publication, 2015.

[41] D. Shasha. Tuning time series queries in finance: Case studies and recommendations. *IEEE Data Eng. Bull.*, 22(2):40–46, 1999.

[42] J. Shieh and E. J. Keogh. *i*sax: indexing and mining terabyte sized time series. In *KDD*, pages 623–631, 2008.

[43] Z. R. Struzik and A. Siebes. Measuring time series' similarity through large singular features revealed with wavelet transformation. In *International Workshop on Database & Expert Systems Applications (DEXA)*, 1999.

[44] T. Warren Liao. Clustering of time series data: a survey. *Pattern Recognition*, 38(11):1857–1874, 2005.

[45] L. Ye and E. J. Keogh. Time series shapelets: a new primitive for data mining. In *KDD*, 2009.

[46] K. Zoumpatianos, S. Idreos, and T. Palpanas. ADS: the adaptive data series index. *VLDB J.*, 25(6):843–866, 2016.

[47] K. Zoumpatianos, Y. Lou, T. Palpanas, and J. Gehrke. Query workloads for data series indexes. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13*, pages 1603–1612, 2015.