*Poster Paper*

# Correlation-Aware Distance Measures for Data Series

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

Michele Dallachiesa
Skysense
michele@skysense.co

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

## ABSTRACT

The field of data series processing has attracted lots of attention thanks to the increased availability of unprecedented amounts of sequential data. These data are then 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 evaluate the benefits of incorporating into the distance functions correlation measures, which enable us to capture the associations among neighboring values in the sequence. We propose three such measures, inspired by statistical and probabilistic approaches. We analytically and experimentally demonstrate the benefits of the new measures using the 1NN classification task, and discuss the lessons learned.

## 1. INTRODUCTION

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 [17, 3, 16, 18, 13]. 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 [12].

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 [1] allows accelerations and decelerations of the signal along the x-axis, and the Longest Common Subsequence (LCSS) distance [7] allows gaps in the sequence. 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)[9]. 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 [4, 5, 6]. 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.

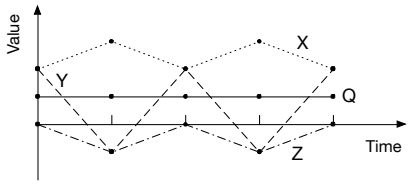


**Figure 1: Euclidean distance fails to distinguish between the $X$ and $Y$ series, given query $Q$.**

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 this work, we make the following contributions. We present distance models inspired by statistical and probabilistic approaches that have been designed to capture the correlation among neighboring points in a data series: autocorrelation, Markov chains and value-difference histograms defined over sliding windows. We combine the proposed models with the Euclidean distance and provide an experimental evaluation with real datasets, which demonstrates the utility of the correlation-aware distance measures.

## 2. 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 $X$

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

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 3 aim to reveal such differences.

# 3. 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.

## 3.1 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 [10]. 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 $n$: $r(\tau) = \frac{E[(x_t-\mu)(x_{t+\tau}-\mu)]}{\sigma^2}$, $\mu$ is a mean and $\sigma^2$ 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 aoutocorrelation vector $n$ is a training parameter.

## 3.2 Markovian Distance

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. 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. 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 $-\log Pr$. 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 [15, 14] for estimating the most significant elements of the transition probability matrix.

## 3.3 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 $l$th 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.

## 3.4 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 [8]. First, we compute a confidence value based on the distances to the two nearest neighbors belonging to two different classes: $Conf = 1 - \frac{d_i}{min_{i \neq j} d_j}, d_j = \min\{dist(Q, X_j) | j \in C\}$. 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.

# 4. EXPERIMENTAL EVALUATION

We compare our methods to the simple and widely-used Euclidean distance for the 1NN classification task. We report the F1 measure: $F1 = 2 * \frac{precision * recall}{precision + recall}$, with $precision = \frac{tp}{tp+fp}$ and $recall = \frac{tp}{tp+fn}$, where $tp$, $fp$ and $fn$ represent true positives, false positives, and false negatives, respectively. Precision and recall are calculated for each class separately, and their arithmetic mean is used to calculate the mean F1 value.

We use 43 UCR datasets with normalized series of different lengths from several domains [11].

## 4.1 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 Figure 2, 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).

We now focus on the performance of the ACD distance, shown in Figure 3. As mentioned in Section 3.1, 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. Figure 3(a) 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 3.4, 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%.

Figure 3(b) 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.

Classification based on the proposed Markovian distance 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 [2] 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 4(a) shows that the Markovian method with failure-prediction outperforms the Euclidean distance in 20 datasets. Moreover, 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 4(b)). This improvement signifies that the Markovian distance is able to capture semantics embedded in the series, which the Euclidean distance fails to uncover.

Finally, we 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.

Figure 5(a) 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. 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.

# 5. CONCLUSIONS

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

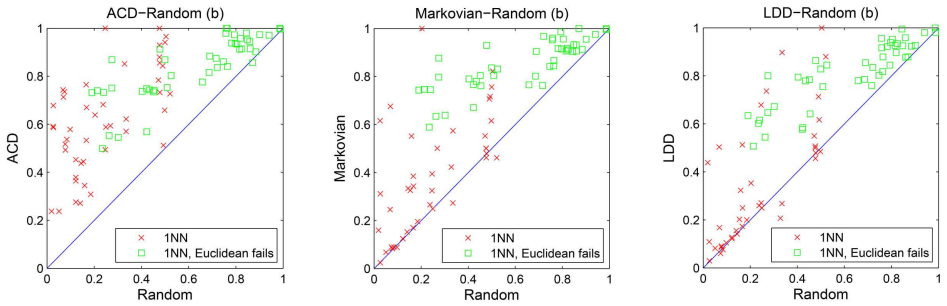


**Figure 2: Comparison of ACD, Markovian, and LDD to a random classifier**

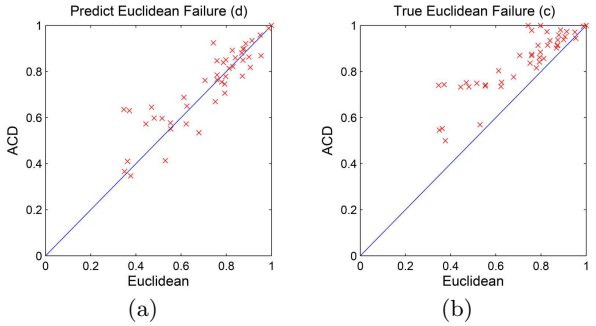


**Figure 3: Comparison for ACD distance**

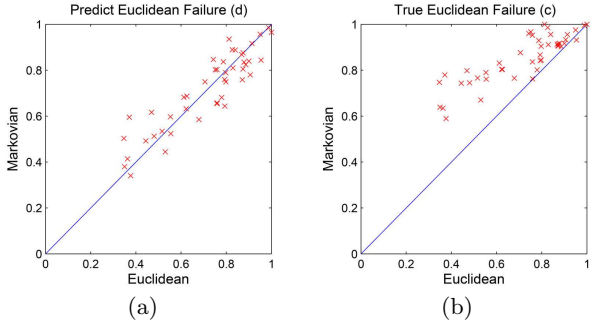


**Figure 4: Comparison for Markovian distance**

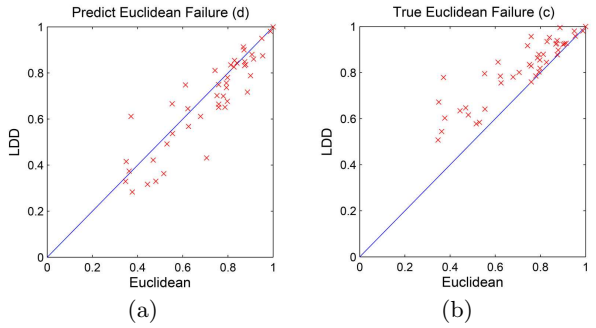


**Figure 5: Comparison for LDD distance**

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.

## References

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

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

[3] 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.

[4] 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.

[5] 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.

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

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

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

[9] 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.

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

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

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

[13] 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.

[14] 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.

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

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

[17] 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.

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