An Effective Martin Kernel for Time Series Classification

Liangang Zhang, Yang Li, and Huanhuan Chen(✉)

UBRI, School of Computer Science, University of Science and Technology of China, Hefei 230027, Anhui, China
{liangang,csly}@mail.ustc.edu.cn, hchen@ustc.edu.cn

Abstract. Time series classification has attracted a lot of attention in recent years. However, the original data often corrupted with noise. To alleviate this problem, many approaches try to perform nonlinear transformation, such that the resulting space could give out the most relevant features. Since the resulting space is not a Euclidean space, strong assumptions are needed for many kernel-based methods for the purpose of obtaining a reasonable measurement. In this paper we propose a novel approach based on Martin distance. The Martin distance is applied to measure the pairwise distance in the resulting space, without imposing strong assumptions on model states. Experiments on several benchmark datasets demonstrate the advantages of the proposed kernel on its effectiveness and performance.

Keywords: Model-based learning · Martin distance · Time series analysis

1 Introduction

Time series appear in many scientific tasks. In practice, most time series are assumed to be generated from fixed but unknown sources. Based on this assumption, learning becomes more subtler as more attention is focused on the underlying but unknown sources. Learning becomes nontrivial since it needs to understand the intricate nature of sources. Among all the learning tasks on time series, classification has been widely recognized as an efficient way.

Among related work, classifications based on Euclidean Distance (ED) or relevant measures are the most popular ones. ED treats every time series as a vector and computes the dissimilarity between two vectors by Euclidean rules. Short time series distance (STS) [1] approximates every time series with piecewise linear functions and measures slope difference between functions. Compared to ED, STS can better capture the temporal difference between two time series. Large Margin Nearest Neighbor (LMNN) [2] provides a way to learn a Mahalanobis distance metric. It builds measurement based on the intuition that neighbors in the same class and the examples from different classes should be separated by a margin. ED, STS, and LMNN are efficient in cases where the time series are

D. Liu et al. (Eds.): ICONIP 2017, Part I, LNCS 10634, pp. 384–393, 2017.
https://doi.org/10.1007/978-3-319-70087-8_41

of equal length. But in practical applications, time series of variable-length are quite often.

Dynamic Time Warping (DTW) [3] is able to process variable-length sequences. It uses nonlinear wrapping in order to find an alignment between variable-length time series. However, DTW can lead to unintuitive alignment, which means a single point at one time series is mapped to a large subsequence in the counterpart [4]. Longest Common Subsequence (LCSS) [5], Edit Distance on Real sequence (EDR) [6], and Edit distance with Real Penalty (ERP) [7] are measurements based on edit distance. LCSS employs *longest common subsequence model* [5] and introduces a threshold parameter which states that two points from different time series are considered to be matched when the distance is no more than the threshold. Unlike LCSS, EDR penalizes the mismatched segments or gaps according to their length. ERP computes the distance between gaps using a *constant reference point* without introducing an additional threshold parameter. DTW, EDR, LCSS and ERP are elastic measures that can better tolerate local time shifting yet they all suffer from high computing complexity.

In order to reduce impact of noise in the data space, instead of computing the similarities between time series in the time domain, an alternative is to compute similarities in a high dimensional space. This methodology is substantiated by the "kernel-trick", i.e. mapping data from original data space to target space by a nonlinear kernel function.

In this methodology, generative models are often employed to fit time series and then the dissimilarities are redefined on the obtained model parameters. Available researches include Kullback-Leibler divergence based kernel (KL) [8], Autoregressive kernel [9], probability product kernel [10], Fisher kernel [11] etc. Those methods use generative probability models in order to obtain highly explicable results. This is advisable if the data is known to following certain distribution. However, the assumption of the particular generative model underlying the data could be too strong for general cases.

If the model that generates the data is unknown, it is sensible to apply "nonparametric" method which is applicable on a wide range of model classes. This idea has been implemented in [12], which employs Echo State Network (ESN) [13] in mapping time series to the model space. The core idea is to carry out classification in the model space, which is filled with the readout weights of trained ESN's. As models are trained in a way of regenerating statistically similar time series without referring to the exterior variables, the trainable part (readout weights in the case of ESN) provides a representation for the training data. The discriminative information of time series is thus assembled in the model space, which in turn provides platform for carrying out classification and other discriminative analysis.

This learning scheme raises the question about how to measure the difference between models. In [3], authors measure the dissimilarities between models in the form of integral of reserver states. The integral is computed under different probabilistic assumptions, which reconsider the model population such that the integral is less affected by outliers. For the reservoir states, the uniform distrib-

ution rationalizes the usage of L_2 norm in computing the dissimilarity between models. The author also poses other kernel-based measurements or probabilistic density functions for more general cases, i.e. a mixture of Gaussian to the reservoir state for the non-uniform distributions and assumes Gaussian form for the residuals in the predefined Fisher kernel. As ESN is a highly nonlinear function, the reservoir is unlikely to satisfy the preconditions of the measurement. Hence, the assumptions greatly limit the scope of applications and may lead to unsatisfactory results. Moreover, these assumptions may be impractical in many situations.

To tackle the problem mentioned above without imposing additional assumptions, we propose a novel kernel based on Martin distance [14] for time series classification. Martin distance, which is designed for dynamical system, is employed to measure the discrepancy of two time series in the model space. This metric relaxes strong assumptions on the reservoir state distributions. Our work keeps in line with learning in the model space and inherits its merits. Compared with work from [12], our method does not assume much on the form of reservoir state and relaxes the Gaussian assumption on the residuals. In this respect, we greatly enlarge the scope of applicable tasks.

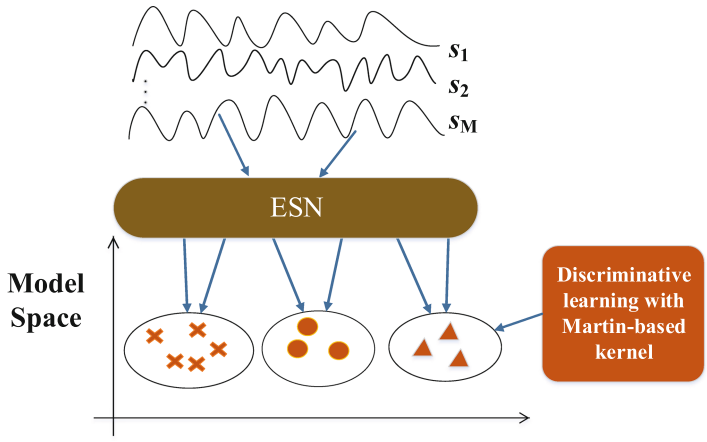

Fig. 1. Illustration of learning in the model space. ESN is firstly employed to map time series to the model space. Each time series is represented by a learned model. The discriminative learning is performed in the model space.

2 Proposed Model

2.1 From Time Series to Model Space

The main idea of carrying out learning in the model space, as illustrated in Fig. 1, is that each model in the model space gives representation to an instance

of the training data. The discriminative learning (e.g. Support Vector Machine (SVM)) is performed in the model space rather than in the data space.

The ESN is subsumed into Reservoir Computing (RC) [15], which provides a principled way for training recurrent neural networks. RC drives a randomly-generated recurrent neural network (the hidden layers are also known as *reservoir*) with the inputs, thereby inducing in "reservoir" a nonlinear response function. The output signals are obtained by linearly combining trainable weights from individual neuron to approximate the response. A typical ESN consists of three layers: input, reservoir and readout. Its topology is shown in Fig. 2. The input and internal weights are randomly generated. Only linear readout weights are trainable. The ESN provides a platform with wide applicability for many learning strategies.

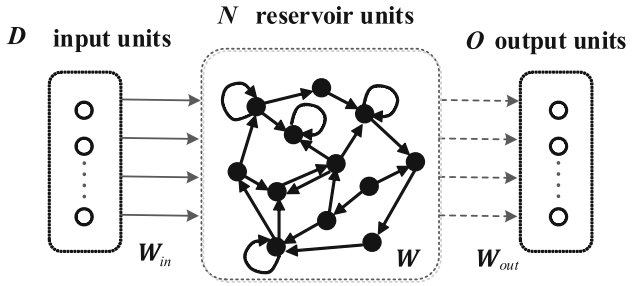

Fig. 2. Illustration of the topology of ESN. The internal units in reservoir are sparsely connected. Solid arrows indicate fixed, random connections and dotted arrows for trainable connections.

The ESN reservoir model with N internal states and without output-reservoir feedback can be formulated as Eq. (1) [16].

$$\begin{cases} \boldsymbol{x}(t+1) = f\left(W\boldsymbol{x}\left(t\right) + W^{in}\boldsymbol{s}(t)\right) \\ \quad\quad \boldsymbol{y}(t) = g(W^{out}\boldsymbol{z}(t)) \end{cases} \tag{1}$$

where $\boldsymbol{x}(t) = [x_1(t), x_2(t), \cdots, x_N(t)]$ is the N-dimensional reservoir state vector at time t, f and g are state activation and output function respectively. In this paper, we use $tanh$ for f. g is set to be identity as routine. $\boldsymbol{s}(t)$ is input time series, W is the $N \times N$ reservoir weight matrix, W^{in} is the $D \times N$ input weight matrix, D is the size of input units, $\boldsymbol{y}(t)$ is the output vector, W^{out} is the $O \times (N + D)$ output weight matrix, $\boldsymbol{z}(t) = [\boldsymbol{x}(t); \boldsymbol{s}(t)]$ is the extended system state at time t and O is the size of output units.

The benefit brought by ESN is that while the general and fixed reservoir offers a shared and rich "pool" for the whole data set and the topologies are independent of any external factors, the individual readout from ESN provides an insight into the intricate nature of each sequence in the training data. The

learning in the model space benefits from the flexibility of ESN in representing specifics of different time series. In the case of ESN, the model is trained to predict the future observation(s) based on the history. As the model states are randomly generated, they are associated no identifiable information of the training data. The output matrix, or the readout weights, act as distinct combiner from a random, large, fixed pool for the nonlinear responses. The readout weights are identifiable for a particular training instance. The readout weights, the only trainable parts, are assembled in the model space, where measurements could be devised. For more information, refer to [12].

2.2 From Martin Distance between Dynamical System to Martin Kernel

In the methodology of learning in the model space, as every ESN is trained to fit an instance of time series and only the readout part is trainable, it is sensible to conduct learning directly on the readout weights. For the purpose of performing discriminative learning, it is necessary to calculate the distance between different models in the obtained model space. A general formation for metric could be formulated as Eq. (2).

$$L_2\left(\boldsymbol{y}_1(\boldsymbol{x}(t)), \boldsymbol{y}_2(\boldsymbol{x}(t))\right) = \left(\int_F \|\boldsymbol{y}_1(\boldsymbol{x}(t)) - \boldsymbol{y}_2(\boldsymbol{x}(t))\| \, d\mu(\boldsymbol{x}(t))\right)^{1/2} \tag{2}$$

where $\mu(\boldsymbol{x})$ is the probability density function defined on the feasible domain F.

In [12], the authors explore cases where closed-form solutions are readily available including uniform and mixtures of Gaussian distributions. The technique of sampling is adopted as an alternative for cases where closed-form solutions are nonexistent or hard to obtain. However, as we have pointed out that the state space is mapped with nonlinear functions, the yielded space is unlikely to be a well-defined space. The experimental results in [12] also confirm this point, and show that the technique based on sampling succeeds in cases where other predefined hypotheses on the distribution $\mu(\boldsymbol{x})$ fail. In a word, the assumption on $\mu(\boldsymbol{x})$ is of no practical advantage in some situations.

To relax the assumptions in [12], we employ Martin distance in the comparison between pairwise models. Martin distance is raised in the behavioral framework [17] and its main advantage is on the independence of particular parameterization of systems. The conventional ways to measure discrepancy between models often rely heavily on specific parameterization of models, but this condition could be easily violated by counter examples which have quite different functional dependency but have identical system behavior. Martin distance measures discrepancy based on system behaviors and could be applicable even when the unique parameterization is unavailable. Moreover, since the distance is based on the behavioral discrepancy, it relies little on the distribution of state space, which partly explains its better classification performance (detailed in Sect. 3.2).

The Martin distance concerns the system behaviors. The computation of similarity involves a linearized system transition matrix A and system output matrix

C. In the case of ESN, as its topology is randomly generated, it carries no information on the training data. The output matrix, or the readout weights, C acts as distinct combiner from a random, large, fixed pool for the nonlinear responses. The readout weights are identifiable for a particular training instance. Based on this discovery, one could safely fix the transition matrix A. The variables for Martin distance are expressed as Eq. (3).

$$\begin{cases} C = W^{out} \\ A = I + \epsilon R \end{cases} \tag{3}$$

where W^{out} is system output matrix of ESN. I is a $(N+D) \times (N+D)$ identity matrix. ϵR is a $(N+D) \times (N+D)$ random matrix of small magnitude in order to ensure the stability in solving the Lyapunov function.

For ease of implementation, we adopt the idea from [18], which points out the relations between distances defined in behavioral framework and subspace angles. In [18] the authors bridge the computation of Martin distance and cosines of subspace angular with Lyapunov equation. The proof is skipped and main result is sketched here. Given the system matrix and output matrix pair $\{A, C\}$, the cosines of largest n subspace principal angles $\{\theta_1, \theta_2, \cdots, \theta_n\}$ of M_i and M_j are equal to the largest n eigenvalues of

$$\begin{pmatrix} 0 & Q_{11}^{-1}Q_{12} \\ Q_{22}^{-1}Q_{21} & 0 \end{pmatrix} \in R^{2n \times 2n},$$

where $Q = \begin{pmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{pmatrix}$ is the unique solution of Lyapunov equation

$$\begin{pmatrix} A_1^T & 0 \\ 0 & A_2^T \end{pmatrix} Q \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix} - Q = \begin{pmatrix} C_1^T \\ C_2^T \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \end{pmatrix}. \tag{4}$$

The Martin distance between systems M_i and M_j is formulated as:

$$d_M(M_i, M_j) = ln \prod_{i=0}^{n} \frac{1}{cos^2 \theta_i} \tag{5}$$

where θ_i is the i-th subspace principal angle between M_i and M_j.

Based on the above formula, after having defined the distance in the model space, any distance-based classification scheme could be used. In this paper, we adopt the "kernel-trick" and define proximity matrix.

$$\mathcal{K}(M_i, M_j) = \frac{1}{d_M(M_i, M_j)} \tag{6}$$

where $\mathcal{K}(M_i, M_j)$ is proximity matrix recording pairwise similarities. For simplicity, the proximity is described as the reciprocal of the distance. The diagonal elements which record the proximities of the same models are assigned with maximum integer in our algorithm. The Martin distance between models M_i and M_j, $d_M(M_i, M_j)$, is defined as Eq. (5). The main algorithm is summarized as Algorithm 1.

Algorithm 1. Kernel based ESN and Martin distance

Input: set of time series $\{\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_M\}$; parameters (number of reservoir units N; number of input units D; number of output units O; size of sliding window w)
Output: Kernel matrix $\mathcal{K}$
1: **for** each time series $\mathbf{s}_i$, $i = 1, \cdots, M$ **do**
2: Slide input data with overlaps.
3: Drive the reservoir state evolution with input data (Eq. (1)).
4: Train W^{out} matrix for $\mathbf{s}_i$.
5: **end for**
6: Calculate the pairwise Martin distance $d_M(M_i, M_j)$ between the i^{th} and j^{th} models $i, j = 1, \cdots, M$.
7: Calculate the Martin kernel matrix as $\mathcal{K}(\mathbf{s}_i, \mathbf{s}_j)$ via Eq. (6).
8: Carry out discriminative learning with SVM based on the obtained $\mathcal{K}(\mathbf{s}_i, \mathbf{s}_j)$.

3 Experiments

3.1 Experiment Setup

Euclidean Distance (ED), Large Margin Nearest Neighbor (LMNN) [2], Dynamic Time Warping (DTW) [3], and Reservoir Based Kernel (RV) [12] are taken as baseline methods in our experiments. All the hyper-parameters of RV are set by 5-fold-cross-validation. The search range of kernel width γ is $\{10^{-6}, 10^{-5}, \cdots, 10^{1}\}$; hyper-parameter of ridge regression $\lambda \in \{10^{-5}, 10^{-4}, \cdots, 10^{1}\}$. In the Martin distance based kernel, we use a fixed reservoir topology with $N = 30$ neurons for all datasets. The size of sliding window is 8. LIBSVM [19] is employed in our method. The slack weight in SVM is set by cross-validation and the search range is $\{10^{-3}, 10^{-2}, \cdots, 10^{3}\}$. One-against-one strategy is selected to perform multi-classification.

3.2 Experiment Results

We perform time series classification task on 15 UCR datasets [20] to validate the efficiency of our proposed kernel based on Martin distance. All the datasets have been divided into training and test sets. The detailed information is presented in Table 1.

Table 2 shows the classification error rates on the benchmark datasets. In order to evaluate the performance of our method, Euclidean Distance (ED), Large Margin Nearest Neighbor (LMNN), Dynamic Time Warping (DTW), and Reservoir Based Kernel (RV) are selected as baseline methods and the lowest error rate on each dataset has been boldfaced. These results demonstrate that, in terms of classification accuracy, our proposed kernel surpasses ED and DTW on 15 datasets and also outperforms RV and LMNN on the most of datasets, especially on *RefrigerationDevices*, *Ham*, and *Beef*. For long time series, e.g. *WormsTwoClass*, *SmallKitchenAppliances* of length 900 and 720 respectively (see Table 1), our kernel method still has lower classification error rates than the baseline methods.

Table 1. Datasets from UCR time series Repository

Dataset	# Classes	# Training set	# Testing set	Length
Coffee	2	28	28	286
Computers	2	250	250	720
Earthquakes	2	139	322	512
Meat	3	60	60	448
OliveOil	4	30	30	570
RefrigerationDevices	3	375	375	720
Herring	2	64	64	512
Ham	2	109	105	431
Wine	2	57	54	234
ScreenType	3	375	375	720
ShapesAll	60	600	600	512
SmallKitchenAppliances	3	375	375	720
WormsTwoClass	2	77	181	900
BeetleFly	2	20	20	512
Beef	2	30	30	470

Table 2. Comparison of Euclidean Distance (ED), Large Margin Nearest Neighbor (LMNN), Dynamic Time Warping (DTW), Reservoir Based Kernel (RV), and our model on fifteen UCR datasets by classification error rates.

Dataset	ED	LMNN	DTW	RV	Our model
Coffee	**0.000**	**0.000**	**0.000**	0.142	**0.000**
Computers	0.424	0.472	0.380	0.304	**0.224**
Earthquakes	0.326	0.245	0.258	**0.202**	0.224
Meat	0.067	**0.033**	0.067	0.083	**0.033**
OliveOil	0.133	0.133	0.133	**0.067**	0.100
RefrigerationDevices	0.605	0.589	0.560	0.533	**0.003**
Herring	0.484	0.438	0.469	0.578	**0.422**
Ham	0.400	0.362	0.400	0.324	**0.009**
Wine	0.389	0.167	0.389	0.148	**0.074**
ScreenType	0.640	0.624	0.589	0.459	**0.219**
ShapesAll	0.248	0.323	0.198	0.283	**0.158**
SmallKitchenAppliances	0.659	0.656	0.328	0.328	**0.227**
WormsTwoClass	0.414	0.481	0.414	0.282	**0.271**
BeetleFly	0.250	0.200	0.300	0.150	**0.050**
Beef	0.333	0.167	0.333	0.367	**0.067**

4 Conclusion

We propose a novel time series kernel based on Martin distance, which measures the pairwise model distance. Our approach is in line with learning in the model space and inherits merits from its learning scheme. Compared with prior work in [12], our method relaxes strong assumptions on the model state. The experimental results confirm its better performance compared with several baseline methods including ED, LMNN, DTW and RV.

Acknowledgments. This work is supported by the National Key Research and Development Program of China (Grant No. 2016YFB1000905), and the National Natural Science Foundation of China (Grants Nos. 91546116, and 61673363).

References

1. Möller-Levet, C.S., Klawonn, F., Cho, K.H., Wolkenhauer, O.: Fuzzy clustering of short time-series and unevenly distributed sampling points. In: Berthold, M.R., Lenz, H.J., Bradley, E., Kruse, R., Borgelt, C. (eds.) IDA 2003. LNCS, vol. 2810, pp. 330–340. Springer, Heidelberg (2003). doi:10.1007/978-3-540-45231-7_31
2. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. J. Mach. Learn. Res. **10**, 207–244 (2009)
3. Berndt, D.J., Clifford, J.: Using dynamic time warping to find patterns in time series. In: KDD Workshop, Seattle, WA, vol. 10, pp. 359–370 (1994)
4. Keogh, E.J., Pazzani, M.J.: Derivative dynamic time warping. In: Proceedings of the 2001 SIAM International Conference on Data Mining, pp. 1–11 (2001)
5. Vlachos, M., Kollios, G., Gunopulos, D.: Discovering similar multidimensional trajectories. In: 18th International Conference on Data Engineering, pp. 673–684. IEEE (2002)
6. Chen, L., Özsu, M.T., Oria, V.: Robust and fast similarity search for moving object trajectories. In: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, pp. 491–502. ACM (2005)
7. Chen, L., Ng, R.: On the marriage of lp-norms and edit distance. In: Proceedings of the Thirtieth International Conference on Very Large Data Bases, pp. 792–803 (2004)
8. Moreno, P.J., Ho, P.P., Vasconcelos, N.: A Kullback-Leibler divergence based kernel for SVM classification in multimedia applications. In: Neural Information Processing Systems, pp. 1385–1392 (2004)
9. Cuturi, M., Doucet, A.: Autoregressive kernels for time series. arXiv preprint arXiv:1101.0673 (2011)
10. Jebara, T., Kondor, R., Howard, A.: Probability product kernels. J. Mach. Learn. Res. **5**, 819–844 (2004)
11. Jaakkola, T.S., Diekhans, M., Haussler, D.: Using the fisher kernel method to detect remote protein homologies. In: ISMB-99 Proceedings, vol. 99, pp. 149–158 (2000)
12. Chen, H., Tang, F., Tino, P., Yao, X.: Model-based kernel for efficient time series analysis. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 392–400. ACM (2013)
13. Jaeger, H.: The echo state approach to analysing and training recurrent neural networks-with an erratum note. German National Research Center for Information Technology GMD Technical Report, vol. 148(34), p. 13 (2001)

14. Martin, R.J.: A metric for arma processes. IEEE Trans. Signal Process. **48**(4), 1164–1170 (2000)
15. Lukoševičius, M., Jaeger, H.: Reservoir computing approaches to recurrent neural network training. Comput. Sci. Rev. **3**(3), 127–149 (2009)
16. Jaeger, H.: Echo state network. Scholarpedia **2**(9), 2330 (2007)
17. Willems, J.C.: From time series to linear system-part I. Finite dimensional linear time invariant systems. Automatica **22**(5), 561–580 (1986)
18. De Cock, K., De Moor, B.: Subspace angles between arma models. Syst. Control Lett. **46**(4), 265–270 (2002)
19. Chang, C.C., Lin, C.J.: Libsvm: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**(3), 27 (2011)
20. Chen, Y., Keogh, E., Hu, B., Begum, N., Bagnall, A., Mueen, A., Batista, G.: The UCR time series classification archive (2015). www.cs.ucr.edu/~eamonn/time_series_data/