An Online Incremental Learning Algorithm For Time Series

Haoran Xu, Youlu Xing, Furao Shen* and Jinxi Zhao
Robotic Intelligence and Neural Computing Laboratory (http://cs.nju.edu.cn/rinc/RINC.Lab),
National Key Laboratory for Novel Software Technology,
Nanjing University, China.
xu3117@gmail.com, youluxing@sina.com, frshen, jxzhao@nju.edu.cn

***Abstract*—Mining time series data has been revived in the last decade due to the increasing availability of time series datasets. This paper presents an online incremental learning algorithm for time series based on the self-organizing incremental neural network (SOINN) and fast dynamic time warping (FastDTW), referred to as OILFTS. The proposed method OILFTS adopts FastDTW distance as the similarity measure, meeting the requirements of most real-time applications. Moreover, OILFTS achieves online and incremental learning of data series which are of equal or unequal length. We test our method with UCR time series datasets, and experimental results show that, from the respect of classification accuracy, the proposed OILFTS is much better than the state-of-the-art similarity measure approaches and widely investigated kernel-based SVMs.**

I. Introduction

Time series is a sequence of observations which are ordered in time or space and can be easily obtained from many applications as diverse as scientific, financial and medical. The character of time series includes: high dimensionality, large in data size and update continuously. Mining time series data has been revived in the last decade due to the increasing availability of time series datasets, finding applications in domains such as medicine, finance, multimedia, and bioinformatics. Many approaches have been proposed for time series learning (clustering or classification), including K-MEDIODS [1], neural networks [2], decision trees [3], SVMs [4], [5], [6], etc.

However, these approaches are all involved with batch learning, utilizing all the input data during the learning process. In the meantime, time series data are usually large in data size and updating continuously. Thus, it would be useful to achieve time series learning in an online way, which means renewing the learning machine step by step following the online input time series data and the latter renewal is independent of the learned data.

Likewise, these approaches do not achieve incremental learning. Incremental learning represents the capability of learning from new data without forgetting already learned knowledge. The main difference between incremental learning and traditional machine learning is that it does not suppose the availability of enough training data before the learning process, but modifies the model during learning process. In industry, data usually become available gradually, this situation demands data analysis systems are capable of learning information in an incrementally way. Furthermore, time series data are usually updating continuously. Therefore, it would be helpful to achieve incremental learning of time series.

In this paper, we propose an online incremental learning algorithm for time series based on SOINN and FastDTW, called OILFTS. The proposed method OILFTS adopts FastDTW distance as the similarity measure, meeting the requirements of most real-time applications. Moreover, OILFTS achieves online and incremental learning of data series which are of equal or unequal length. Section 2 briefly introduces the related work. Then section 3 details the proposed method OILFTS. In section 4 we test our method on standard datasets UCR time series datasets [12], comparing with the state-of-the-art similarity measure approaches and widely investigated kernel-based SVMs. Finally, Section 5 concludes the paper.

II. Related Work

A. The Self-organizing Incremental Neural Network

The self-organizing incremental neural network (SOINN) [7] is an efficient online incremental learning algorithm. By computing the Euclidean distance among the input data, it divides the input data into different clusters, learning the density distribution of the input data and utilizing nodes and edges to denote that distribution. Generally, SOINN is an unsupervised learning algorithm that requires no prior knowledge, such as the number of classes and the distribution of input data. It is capable of dividing clusters with complicated shape, reporting an appropriate number of clusters and representing topology structure of the input data.

For each input data, SOINN seeks the nearest node (winner) and the second nearest node (second winner) from the node set. Then the Euclidean distances between input data and the winner, and the second winner are calculated. By comparing with the similarity thresholds, it determines whether the input data belongs to the same cluster as the winner.

If the input data does not belong to the same cluster as the winner, the input data will be inserted into the network as a new node. Otherwise, the weight of the winner and its neighbors will be updated, i.e., move the winner and its neighbors toward the input data. Besides, the winner and second winner are connected with an edge if there is no edge links them.

B. Dynamic Time Warping

Dynamic time warping (DTW) seeks the optimal alignment between two time series, and obtains flexible distances by

*Corresponding author is Furao Shen

978-1-4799-1959-8/15/$31.00

aligning the elements inside both time series. Let $A = (a_1, ..., a_M)$ and $B = (b_1, ..., b_N)$ represent two time series with length of M and N. To compute the DTW distance between time series A and B, the first step is to obtain a $M \times N$ distance matrix D which contains pairwise distance between the elements inside A and B, where $D_{(i,j)} = (a_i - b_j)^2$ is generally used, i = 1,...,M, j = 1,...,N. The target is to seek a path through the matrix D which begins at $(1, 1)$ and ends at (M, N) so that the grand total distance between A and B is minimized. This minimized grand total distance is the DTW distance between A and B.

Let $W_{(i,j)}$ represent the DTW distance between subsequences $(a_1, ..., a_i)$ and $(b_1, ..., b_j)$, then $W_{(i,j)}$ is obtained by settling the following recursion formula:

$$W_{(i,j)} = D_{(i,j)} + min(W_{(i-1,j)}, W_{(i,j-1)}, W_{(i-1,j-1)}). \quad (1)$$

Finally, we obtain the DTW distance between time series A and B with $dist_{DTW}(A, B) = W_{(M,N)}$.

C. Fast Dynamic Time Warping

Fast dynamic time warping (FastDTW) is an approximate DTW algorithm that provides optimal or near-optimal alignments, which applies a multilevel method with three critical operations [11]:

(1) $Coarsening$: Shrink a time series into a shorter time series which denotes the same curve as precisely as possible with fewer elements.

(2) $Projection$: Seek a minimum distance warping path at a coarser resolution, and apply it as an initial assumption for a finer resolution′s minimum distance warping path.

(3) $Refinement$: Refine the warping path projected from a coarser resolution by partially adapting the warping path.

By averaging adjacent pairs of elements, $Coarsening$ shrinks the length of a time series. The resulting time series is two times smaller than the raw time series. To produce different lengths of a time series, coarsening is carried out many times. $Projection$ takes a warping path computed at a coarser resolution and determines what cells it goes through in a finer resolution. Due to the resolution is increased by two times, a single element in the coarser resolution warping path will cast to at least four elements in the finer resolution. During refinement the projected warping path is then utilized as an initial assumption to seek a warping path at the finer resolution. An optimal warping path is found during $Refinement$ in the neighbor of the projected path.

III. The Proposed Method OILFTS

To achieve online and incremental learning of time series, in this paper, we propose a new method based on the self-organizing incremental neural network and fast dynamic time warping, called OILFTS. The proposed OILFTS adopts FastDTW distance as the similarity measure, and achieves online and incremental learning of time series. Moreover, it meets the requirements of most real-time applications. The flowchart of OILFTS is shown in Fig. 1. OILFTS uses node set and edge set to denote density distribution of the input time series data. Firstly, node set is set to contain two nodes with two input time series data. For each online input time series data, the nearest node s_1 (winner) and the second nearest node s_2 (second winner) in the node set is found. With a threshold-based scheme, OILFTS self-adaptively determines whether to insert a new node. With the time series data continually input, the knowledge are learned incrementally. The detailed algorithm is given in Algorithm 1.

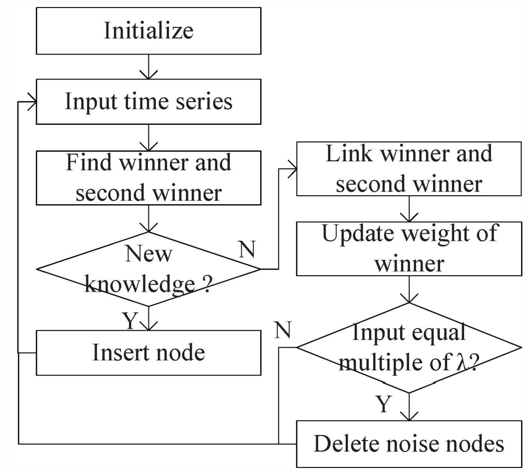

Fig. 1. Flowchart of the proposed OILFTS. Node set A is initialized to contain two nodes with two input time series data. For each online input time series data, OILFTS automatically decides whether to insert a new node. With the time series data continually learned, the knowledge are learned incrementally. If the number of input time series data reaches an integral multiple of parameter λ, noise nodes is removed from node set with a threshold-based scheme.

Most works in the field of time series learning (clustering and classification) are concerned with the definition and utilization of a similarity measure between time series. Euclidean distance is universally acknowledged as the simplest similarity measure between time series. Specifically, SOINN adopts Euclidean distance as the similarity measure [7]. However, this similarity measure does not in accord with the commonly understanding of what time series really is, considering time series is highly sensitive to temporal distortion, phase shifting and might not be of equal length.

Dynamic time warping (DTW) proves a great success in time series matching [8], [9], finding applications in speech recognition and more recently in a variety of time series mining applications. It seeks the optimal alignment between two time series, and obtains flexible distances by aligning the elements inside both time series. However, due to the computational complexity of DTW distance is $O(n^2)$ and time series data are usually of high dimensionality, DTW is too demanding for computation for most real-time applications [10].

Fast dynamic time warping (FastDTW) [11] is an approximate DTW algorithm that provides optimal or near-optimal alignments with a linear time and memory complexity, meeting the requirements of most real-time applications. Thus, in the proposed method OILFTS, FastDTW is adopted as the similarity measure.

The proposed OILFTS includes training phase and testing

Algorithm 1 OILFTS

1: Initialize node set A to contain two nodes c_1 and c_2 with first two input time series data.
2: Set edge set C, storing edge between nodes, to the empty set.
3: Input a new time series data $\xi \in R^n$.
4: Seek the nearest node s_1 (winner) and the second nearest node s_2 (second winner) of the input time series ξ from A by

$$s_1 = \underset{c \in A}{arg\,min}\ dist_{FastDTW}(\xi, W_c). \tag{2}$$

$$s_2 = \underset{c \in A \setminus \{s_1\}}{arg\,min}\ dist_{FastDTW}(\xi, W_c). \tag{3}$$

5: If $dist_{FastDTW}(\xi, s_1) > T_{s_1}$ or $dist_{FastDTW}(\xi, s_2) > T_{s_2}$, insert ξ to node set A. Then go to step 3 to process the next input time series data.
6: If there is no edge between s_1 and s_2, add edge (s_1, s_2) to edge set C:

$$C = C \cup (s_1, s_2). \tag{4}$$

Set the age of edge between s_1 and s_2 to zero:

$$age_{(s_1,s_2)} = 0. \tag{5}$$

7: Increases M_{s_1}, representing local accumulated number of input time series data of node s_1, by 1.

$$M_{s_1} = M_{s_1} + 1. \tag{6}$$

8: For all s_i in the neighbor area of s_1:

$$age_{(s_1,s_i)} = age_{(s_1,s_i)} + 1. \tag{7}$$

9: Transform W_{s_1}, denoting weight of the winner s_1, and the input time series data ξ to equal length.

$$[W_{s_1}^{'}, \xi^{'}] = transform(W_{s_1}, \xi, M_{s_1}). \tag{8}$$

10: Update W_{s_1} with transformed $W_{s_1}^{'}$ and $\xi^{'}$:

$$W_{s_1} = W_{s_1}^{'} + \frac{1}{M_{s_1}}(\xi^{'} - W_{s_1}^{'}). \tag{9}$$

11: Adjust the similarity threshold T_{s_1}/T_{s_2} to the maximum distance between s_1/s_2 and its neighbors.
12: Remove edges with an age larger than a predefined threshold age_{dead}.
13: If the number of input time series data reaches an integral multiple of parameter λ, for all c_i in A, if c_i has no or only one neighbor and M_{c_i} is less than an predefined threshold, remove it from the node set.
14: Go to step 3 to continue the online learning process.

phase. Online time series data are used for training and testing. Training phase learns the density distribution of the input time series data and utilizes nodes and edges to denote that distribution. Then in the testing phase the learned nodes of the networks are used for classifying the test time series.

During the learning process, the time series data is input one by one, renewing the network step by step and the latter renewal is independent of the learned data, which achieves the online learning of time series and significantly reduces the storage needs.

Incremental learning means the network grows dynamically and nodes are sequentially inserted properly. Nevertheless, permanent insertion might lead to perpetual growth of network. Thus, we need to determine when to insert a new node and when to remove an old node.

We apply a threshold-based method to determine when to insert a new node. For the input time series data ξ, the first step is to seek the nearest node s_1 (winner) and the second nearest node s_2 (second winner) of the input time series. Then if the distance between ξ and s_1 greater than the threshold parameter of s_1, or the distance between ξ and s_2 greater than the threshold parameter of s_2, the input time series data will be inserted into the network. This method ensures the proposed OILFTS to generate nodes for each class properly.

Besides, the proposed OILFTS is capable of denoising [13], determining when to remove an old node. Noise generally exists in real life data. We need to take measures to remove those nodes in the network that are probably to be noise. Constructing proper topology structure is helpful for detecting probable noise in the network. We apply the competitive Hebbian rule [14] to link two nodes in case of they become the winner and the second winner of an input time series data. In this paper, we take the following strategy [7] for denoising: if the number of input time series data up to an integer multiple of a parameter, we delete nodes with no or only one neighbor.

IV. Experiment

In this section, we assess the effectiveness of proposed method with the UCR time series datasets [12]. Firstly, a brief introduction of the UCR time series datasets and experimental setup is given. Then, we assess the classification error rate of proposed method with the state-of-the-art similarity measure approaches and widely investigated kernel-based SVMs, respectively.

A. UCR time series datasets and experimental setup

The UCR time series datasets include 20 two-class or multi-class problems for various applications, such as biomedical data, motion tracking in video monitoring, electromagnetic measurements of lightning activity and so on. The length of the time series ranges from 60 to 637. For each of the 20 datasets, a training subset and a test subset are predefined. The essential information of each dataset is briefly described in $Table$ 1.

Model selection is done by executing 5-folder cross-validation method on the training subset to determine the hyper-parameter values [15]. After training a classifier using the optimized hyper-parameter values, we get the final classification error rate on each dataset with the test subset.

In order to obtain a comprehensive assessment, we compare the performance of the proposed OILFTS with two state-of-the-art similarity measure approaches, and three widely investigated kernel-based SVMs. Specifically, we compare average ranks of multiple approaches, and then analyze the statistical difference of multiple approaches with the nonparametric Bonferroni-Dunn test [16], which is appropriate for comparing classifiers over multiple datasets.

TABLE I. DESCRIPTION OF THE UCR TIME SERIES DATASETS.

Dataset	Class	Length	Training	Test
Synthetic Control	6	60	300	300
Gun-Point	2	150	50	150
CBF	3	128	30	900
Face (all)	14	131	560	1,690
OSU Leaf	6	427	200	242
Swedish Leaf	15	128	500	625
50Words	50	270	450	455
Trace	4	275	100	100
Two Patterns	4	128	1,000	4,000
Wafer	2	152	1,000	6,174
Face (four)	4	350	24	88
Lightning-2	2	637	60	61
Lightning-7	7	319	70	73
ECG200	2	96	100	100
Adiac	37	176	390	391
Yoga	2	426	300	3,000
Fish	7	463	175	175
Beef	5	470	30	30
Coffee	2	286	28	28
Olive Oil	4	570	30	30

B. Comparison with similarity measure approaches

With the classification error rate as the assessing criteria, we compare the performance of the proposed OILFTS with two state-of-the-art similarity measure approaches, including 1-nearest-neighbor classifier with Euclidean distance (1NN-ED) and 1-nearest-neighbor classifier with DTW (1NN-DTW). $Table$ 2 shows the experimental results of these approaches on each dataset. These approaches are ranked on each dataset on the basis of their classification error rate and the average ranks are obtained over all datasets (lower ranking represents better performance).

Average ranks provide a fair comparison of these approaches. On average, the proposed OILFTS ranks the first, with the 1NN-DTW trailing slightly behind, and 1NN-ED ranks the last. In terms of average ranks, OILFTS is much better than 1NN-ED. Compared with 1NN-DTW, the proposed OILFTS is slightly better in average ranks, but OILFTS takes much less time than 1NN-DTW.

With the Bonferroni-Dunn test, we analyze the performance difference between these approaches. At the significance level $\alpha = 0.05$, the critical value of the test is 1.52. The proposed method OILFTS is statistically better than 1NN-ED. Specifically, over all the 20 datasets, the number of OILFTS performs better than 1NN-ED is 13, the number of two methods perform the same is 1, while the number of OILFTS performs worse than 1NN-ED is 6.

C. Comparison with kernel-based SVMs

In this part, we compare the performance of the proposed OILFTS with three widely investigated kernel-based SVMs, including SVM with Gaussian RBF kernel (GRBF-SVM), SVM with Negated DTW kernel (NDTW-SVM) and SVM with Gaussian DTW kernel (GDTW-SVM). $Table$ 2 shows the experimental results of these approaches on each dataset.

On average, the proposed OILFTS ranks the first, GRBF-SVM and GDTW-SVM rank the second and the third, and NDTW-SVM ranks the last. Apparently, in terms of average ranks, OILFTS is much superior to other three approaches.

With the Bonferroni-Dunn test, we analyze the performance difference between these approaches. At the significance level $\alpha = 0.05$, the proposed method OILFTS is statistically better than NDTW-SVM, and GDTW-SVM is just below significant difference. Specifically, over all the 20 datasets, the number of OILFTS performs better than NDTW-SVM is 18, the number of two methods perform the same is 1, while OILFTS performs worse than NDTW-SVM only on one data set "Trace". Compared with GRBF-SVM, the proposed OILFTS is more effective in improving the classification accuracy of time series. For example, over all the 20 datasets, OILFTS outperforms GRBF-SVM on 13 datasets, while GRBF-SVM outperforms OILFTS on 7 datasets.

V. CONCLUSION

In this paper, we propose an online incremental learning algorithm for time series based on the self-organizing incremental neural network (SOINN) and fast dynamic time warping (FastDTW), called OILFTS. The proposed OILFTS adopts FastDTW as the similarity measure, which provides optimal or near-optimal alignments between two time series with a linear time and memory complexity, meeting the requirements of most real-time applications. By learning the time series data one by one, OILFTS achieves online learning of time series. By applying a threshold-based method to determine when to insert a new node and a denoising scheme to determine when to remove an old node, OILFTS achieves incremental learning of time series.

We test our method with UCR time series datasets, and experimental results show that, in terms of classification accuracy, the proposed OILFTS is much better than the state-of-the-art similarity measure approaches and widely investigated kernel-based SVMs.

ACKNOWLEDGEMENT

This work is supported in part by the National Science Foundation of China under Grant Nos. (61375064, 61373001) and Jiangsu NSF grant (BK20131279).

REFERENCES

[1] Hautamaki V, Nykanen P, Franti P. Time-series clustering by approximate prototypes[C]//Pattern Recognition, 2008. ICPR 2008. 19th International Conference on. IEEE, 2008: 1-4.

[2] Nanopoulos A, Alcock R, Manolopoulos Y. Feature-based classification of time-series data[J]. International Journal of Computer Research, 2001, 10(3).

[3] Rodrłguez J J, Alonso C J. Interval and dynamic time warping-based decision trees[C]//Proceedings of the 2004 ACM symposium on Applied computing. ACM, 2004: 548-552.

[4] Bahlmann C, Haasdonk B, Burkhardt H. Online handwriting recognition with support vector machines-a kernel approach[C]//Frontiers in Handwriting Recognition, 2002. Proceedings. Eighth International Workshop on. IEEE, 2002: 49-54.

[5] Noma H S K. Dynamic time-alignment kernel in support vector machine[J]. Advances in neural information processing systems, 2002, 14: 921.

[6] Cuturi, Marco, et al. A kernel for time series based on global alignments[C]//Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on. IEEE, 2007, 2: II-413-II-416.

[7] Furao S, Hasegawa O. An incremental network for on-line unsupervised classification and topology learning[J]. Neural Networks, 2006, 19(1): 90-106.

[8] Piyush Shanker A, Rajagopalan A N. Off-line signature verification using DTW[J]. Pattern Recognition Letters, 2007, 28(12): 1407-1414.

TABLE II. CLASSIFICATION ERROR RATE (%) FOR THE UCR TIME SERIES DATASETS.

Dataset	1NN-ED	1NN-DTW	GRBF-SVM	NDTW-SVM	GDTW-SVM	OILFTS
Synthetic Control	12.00	**0.67**	2.33	1.33	1.00	1.00
Gun-Point	8.67	9.33	**4.00**	38.67	7.33	8.00
CBF	14.78	0.33	10.55	0.33	1.22	**0.00**
Face (all)	18.05	11.83	16.68	17.04	23.96	**1.89**
OSU Leaf	48.35	40.91	42.97	65.70	36.77	**30.99**
Swedish Leaf	14.72	13.28	**8.64**	33.44	18.08	17.76
50Words	27.47	**22.42**	29.89	41.69	30.76	27.03
Trace	24.00	**0.00**	15.00	**0.00**	**0.00**	1.00
Two Patterns	9.33	**0.00**	7.52	0.67	0.02	**0.00**
Wafer	**0.00**	**0.00**	0.43	18.12	2.77	0.92
Face (four)	21.59	17.05	19.31	10.22	10.22	**5.68**
Lightning-2	**0.00**	**0.00**	29.50	44.26	13.11	13.11
Lightning-7	42.47	27.4	45.07	34.24	26.02	**21.91**
ECG200	**0.00**	**0.00**	14.00	36.00	16.00	14.00
Adiac	27.88	28.9	**24.55**	68.79	37.59	36.82
Yoga	16.97	16.37	14.56	54.30	19.50	**8.40**
Fish	21.71	16.57	**14.28**	78.28	20.00	16.57
Beef	**46.67**	50.00	56.67	53.33	56.67	**46.67**
Coffee	25.00	17.85	**7.14**	10.71	17.85	10.71
Olive Oil	**13.33**	**13.33**	**13.33**	60.00	23.33	16.67
Average rank	3.95	2.43	3.35	4.70	3.85	**2.38**

[9] Rath T M, Manmatha R. Word image matching using dynamic time warping[C]//Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on. IEEE, 2003, 2: II-521-II-527 vol. 2.

[10] Xi X, Keogh E, Shelton C, et al. Fast time series classification using numerosity reduction[C]//Proceedings of the 23rd international conference on Machine learning. ACM, 2006: 1033-1040.

[11] Salvador S, Chan P. Toward accurate dynamic time warping in linear time and space[J]. Intelligent Data Analysis, 2007, 11(5): 561-580.

[12] Keogh E, Xi X, Wei L, et al. The UCR time series classification/clustering homepage[J]. URL=http://www.cs.ucr.edu/~eamonn/time_series_data, 2006.

[13] Furao S, Ogura T, Hasegawa O. An enhanced self-organizing incremental neural network for online unsupervised learning[J]. Neural Networks, 2007, 20(8): 893-903.

[14] Martinetz T. Competitive Hebbian learning rule forms perfectly topology preserving maps[M]//ICANN93. Springer London, 1993: 427-434.

[15] Salzberg S L. On comparing classifiers: Pitfalls to avoid and a recommended approach[J]. Data mining and knowledge discovery, 1997, 1(3): 317-328.

[16] Demsar J. Statistical comparisons of classifiers over multiple data sets[J]. The Journal of Machine Learning Research, 2006, 7: 1-30.