A Trend Based Similarity Calculation Approach for Mining Time Series Data

Yuhang Yang, Yingju Xia, Fujiang Ge, Yao Meng, and Hao Yu

***Abstract*—In this study, a Trend Based Similarity Calculation (TBSC) approach is proposed for mining time series data. Trend information, which are ignored in existing techniques, are extracted for similarity calculation. The proposed approach involves two steps. The first step is trend based transform. The second step is similarity calculation measured by the proposed weighted edit distance. Experiments are conducted for time series classification on twenty datasets from different areas. Results show that the proposed approach achieves significant improvements on some datasets. Another set of experiment figure out it is possible to select the right approach for a particular dataset. That guarantees the overall performance can be enhanced. This paper also proposes a set of measures in order to describe the characteristics of different datasets. Correlations between performance and the proposed measures are analyzed.**

***Index Terms*—Similarity Calculation, Trend Based Transform, Weighted Edit Distance, Classification, Time Series**

I. Introduction

A time series is an ordered set of real values which can be defined as $X = [x_1, \ldots, x_N]$ of N values, where x_i denotes the value corresponding to the time slot i $i \in T = \{t_1,...,t_N\}$ and T is the domain of time. Techniques of mining time series has been applied in countless applications, like sensor measurements [1], mobile object tracking [2], data center monitoring [3], motion capture sequences [4, 5], environmental monitoring [6] and many more.

Similarity calculation is a fundamental technique which is irreplaceable in many tasks, such as classification, clustering and anomaly detection. Generally speaking, similarity calculation involves two steps. The first step is representation which aims to represent a time series in an appropriate way. The second step is to select a similarity measurement to measure the similarity.

Representation methods can be mainly divided into four categories: continuous time domain representation, transform based representation, discretization based representation and model based representation. Since time series data deals with raw format which is expensive in terms of processing and storage, some researches focus on time series data format to solve the above problem. For analysis of such data, the ability to process the data in a single pass, or a small number of passes, while using little memory, is crucial. Feature extraction and feature selection are commonly used for dimensionality reduction, e.g. unsupervised feature extraction algorithm [7] and underlying uniqueness based global feature extraction [8]. Clipped data [9, 10], based on a bit level approximation of the data, is another representative method which aims to improve the efficiency of time series processing.

Similarity measures include lock-step measure, elastic measure and some others. L_p-norms are the most popular and simple measure. The L^p spaces are function spaces defined using natural generalizations of p-norms for finite-dimensional vector spaces. When $p = 2$, the space L^2 is the only Hilbert space of this class. $L_2(\vec{x},\vec{y})$ is the most commonly used Euclidean distance [11]. Autocorrelation has been proposed [12], along with a variety of other measures in recent years, such as cosine wavelets [13] and piecewise probabilistic measures [14]. However, the empirical comparison in [15] revealed that the Euclidean distance metric still performs favorably compared to others when tested on the same datasets. Elastic measures, such as Dynamic Time Warping Distance [16], can deal with similarity calculation of time series with variable lengths.

Existing techniques suffer from several problems. The first problem is information loss. In some situations, the crucial information are not notable in the original time series. In other situations, significant information are lost in the transform process. The second problem is over fitting. Many existing methods use complicated parameter tuning algorithms. When the test data and training data are not so similar, the performance drop significantly. The third problem is space consuming and time consuming. Some approaches, such as Polynomial and Probabilistic, can lead to good offline prediction accuracy but not suitable for online stream environment. Because online processing requires low prediction and training costs.

In this study, we propose a Trend Based Similarity Calculation (TBSC) approach. The main idea of the proposed approach is to extract trend information to characterize a time series. It involves two steps. The first step is trend based transform. The second step is similarity calculation based on the proposed weighted edit distance. Experiments are conducted on time series classification by using twenty datasets from different areas in order to verify the efficiency of the approach. Another set of experiments are conducted to figure out if it is possible to select an appropriate approach for a particular dataset in advance. A set of measures are also proposed in order to describe the characteristics of different datasets. Correlations between performance and different measures are analyzed. The satisfactory performance can be guaranteed by choosing appropriate approach for a particular dataset.

The rest of the paper is organized as follows. Section 2

Manuscript received January 4, 2012; revised January 20, 2012.

Y. Yang, Y. Xia, F. Ge, Y. Meng, and H. Yu are with the Fujitsu Research & Development Center Co., LTD. 15/F, Tower A, Ocean International Center,No.56 Dong Si Huan Zhong Rd, Chaoyang District, Beijing, 100025, P.R. China (phone: +86-10-59691000 ext 5752; fax: +86-10-59691504; e-mail: yyh@cn.fujitsu.com).

describes the proposed algorithms. Section 3 explains the experiments and the performance evaluation. Section 4 is the conclusion.

II. Algorithm Design

The proposed Trend Based Similarity Calculation approach extracts the trend information, instead of actual values used in existing techniques, for similarity calculation. First, the trend information are extracted based on trend based transform. Second, the similarity is calculated by using the weighted edit distance.

A. Trend Based Transform

The main idea of the proposed trend based transform is to compare the values of the current node with the last one. Trend information represent the changes compared to the last node. The trend information, which are always ignored in the existing techniques, are extracted for similarity calculation. Compared to most existing techniques based on absolute values, similarities are obtained by using the comparison results. A original time series $X = [x_1, \ldots, x_N]$ can be transferred to a trend based series $X' = [x'_{1,\ldots,}x'_{N-1}]$. A typical equation for trend based transform is shown as follows.

$$x_i' \begin{cases} 0, x_{i+1} < x_i - \varepsilon \\ 1, x_i - \varepsilon \le x_{i+1} \le x_i + \varepsilon \\ 2, x_{i+1} > x_i + \varepsilon \end{cases} \quad (1)$$

Where $\varepsilon (0 \le \varepsilon \le \max |x_{i+1} - x_i|)$ is a parameter which can be predefined or selected by using a parameter tuning technique. In this equation, three types of trends are identified: down (0), smooth (1) and up (2). By using this equation, absolute values are replaced by trend information.

Of course, the transform equation can be changed to satisfy different requirements. Another instance is shown below.

$$x_i' \begin{cases} 0, x_{i+1} \le \gamma \times x_i \\ 1, x_{i+1} > \gamma \times x_i \end{cases} \quad (2)$$

Where γ is threshold of abnormal which indicates multiples of the last node value. In equation (2), trend is divided into two subcategories: normal and abnormal. x'_i is set to 1 if x_{i+1} is much higher than x_i which means abnormal. Otherwise, x'_i is set to 0 which means normal.

B. Weighted Edit Distance

Edit distance was first proposed for distance calculation between different character strings. Edit distance has been widely used in many tasks. The distance is measured by the number of operations required to change one string into the other. In the traditional edit distance, operations are treated equally. In the proposed weighted edit distance, operations are treated differently. Giving two transferred time series $X' = [x'_{1,\ldots,}x'_{N-1}]$ and $Y' = [y'_{1,\ldots,}y'_{M-1}]$, the details of weighted edit distance are shown as follows:

$$\begin{aligned} &\forall i, j > 0 \\ &D(i,j) = \min[D(i-1,j) + c_d, D(i,j-1) + c_i, D(i-1,j-1) + c_r] \\ &D(i,0) = D(i-1,0) + c_d \\ &D(0,j) = D(0,j-1) + c_i \\ &D(0,0) = 0 \end{aligned} \quad (3)$$

$$D(X', Y') = D(N-1, M-1) \quad (4)$$

$$c_r(i,j) = \begin{cases} 0 & \text{if } x_i' = y_j' \\ |x_i' - y_j'| / \max|x_i' - y_j'| & \text{if } x_i' \ne y_j' \end{cases} \quad (5)$$

Generally speaking, there are three types of operations including insert, delete and replace. The costs of different operations are set to 1 in traditional edit distance. In this case, insert and delete operations ($c_i = 1$; $c_d = 1$) are still set to 1. The main difference is that the cost of replace operation (c_r) is dependent on particular change. If different trend types are computable, the cost of replace operation ($c_{replace}(i, j)$) is measured by the distance from x_i to y_j and normalized by dividing the max distance. Taking equation (1) as an example, there are three types of trends which denote down (0), smooth (1) and up (2), respectively. Down and smooth are more similar than down and up. By using equation (5), the cost of replacing down to smooth is 0.5 while the cost of replacing down to up is 1. Thus minor differences can be captured.

III. Performance Evaluation

Similarity calculation is an irreplaceable step in many tasks. In this study, experiments are conducted on time series classification in order to evaluate the proposed approach. In this case, equation (1) introduced in Section 2 is used for trend based transform. First, a set of experiments are conducted on time series classification to compare the proposed approach with the state-of-art algorithms. Second, another set of experiments are conducted to figure out if it is possible to select an appropriate approach for a particular dataset in advance. Third, some measures are proposed to describe the characteristics of different datasets. The correlation between the performance of the proposed approach and different measures are analyzed.

Twenty time series datasets [17] from different areas and within different sizes are used for experiments. The datasets cover different research areas, such as image (OSU Leaf, Swedish Leaf), handwritten word spotting (50Words), and sensor data (Wafer). Numbers of classes also cover a long rage (from 2 to 50).

For comparison, two state-of-art algorithms, Euclidean distance and DTW (Dynamic Time Wrapping) which are wildly used and achieve good performance in literature, are taken as baseline methods. Different distance calculation methods are used in the same classification model in order to verify the efficiency of the proposed approach. In order to focus on the similarity calculation techniques, a straight forward classification model 1-NN (1-Nearest Neighbor) is applied. For each time series in the testing data, 1-NN classification finds the time series having the nearest distance with the target from the training set. The target time series is assigned to the class of the nearest neighbor. Performance are evaluated in terms of classification error rate. Table 1 shows the experimental results.

As shown in Table 1, the trend based similarity calculation outperforms baseline methods in 8 of 20 datasets. In some datasets, the improvements are significant. For an instance, the classification error has been reduced more than 50% on the OSU Leaf dataset. It should be pointed out that ε in equation (1) is a parameter having direct impact on the algorithm performance. In the experiments, training process is very straightforward. Training data are divided into two

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

datasets with the same size for pre-training and pre-testing. From 0 to the max distance between each pair of neighbor nodes, there are only 11 values with the same step length are tested. The best parameter is selected for real testing. Thus there is still room for performance improvement by considering the simple parameter turning process.

Table 1. Performance of Differernt Algorithms

Dataset	Euclidean Distance	Best Warping Window DTW (r)	DTW, no Warping Window	TBSC
Synthetic Control	0.12	0.017 (6)	0.007	0.74
Gun-Point	0.087	0.087 (0)	0.093	**0.06**
CBF	0.148	0.004 (11)	0.003	0.521
Face (all)	0.286	0.192 (3)	0.192	0.326
OSU Leaf	0.483	0.384 (7)	0.409	**0.182**
Swedish Leaf	0.213	0.157 (2)	0.210	0.208
50Words	0.369	0.242 (6)	0.310	0.440
Trace	0.24	0.01 (3)	0.0	0.3
Two Patterns	0.09	0.0015 (4)	0.0	0.522
Wafer	0.005	0.005 (1)	0.020	**0.0**
Face (four)	0.216	0.114 (2)	0.170	**0.102**
Lightning-2	0.246	0.131 (6)	0.131	0.508
Lightning-7	0.425	0.288 (5)	0.274	0.671
ECG	0.12	0.12 (0)	0.23	0.22
Adiac	0.389	0.391 (3)	0.396	0.558
Yoga	0.170	0.155 (2)	0.164	**0.151**
Fish	0.217	0.160(4)	0.167	**0.063**
Beef	0.467	0.467(0)	0.5	**0.2**
Coffee	0.25	0.179(3)	0.179	**0.0357**
OliveOil	0.133	0.167(1)	0.133	0.267

It is perfect that a proposed method outperforms other algorithms in most situations. However, it is really difficult or almost impossible. It is meaningful to select a more appropriate approach for each dataset. Thus another set of experiments are conducted to verify the possibility. The training data is split into two subsets which are taken as pre-training set and pre-testing set, respectively. The similarity calculation approach which achieves the best performance in the pre-testing dataset is selected for the real testing. The proposed TBSC approach is compared with Euclidean distance in the experiments. Fig. 1 shows the performance comparison between TBSC and Euclidean distance in both training and testing datasets. X-axis is performance improvement of TBSC compared with Euclidean distance in the pre-testing set. Similarly, Y-axis is performance improvement in the real testing set. As shown is Fig.4, there are four districts named PP (positive performance in both pre-test and real test), PN (positive performance in pre-test but negative performance in real test), NP (negative performance in pre-test but positive performance in real test) and NN (negative performance in both pre-test and real test). TBSC outperforms Euclidean distance in pre-test on 9 datasets. In such situations, TBSC is selected for real test. As shown in Fig. 1, there are 8 points in PP field and 1 point in PN field. The possibility of performance improvement by using TBSC is 88.9% (8 datasets out of 9), while the possibility of accuracy loss caused by TBSC is only 11.1% (1 datasets out of 9). There is still room to further avoid such precision loss by considering the naive training process.

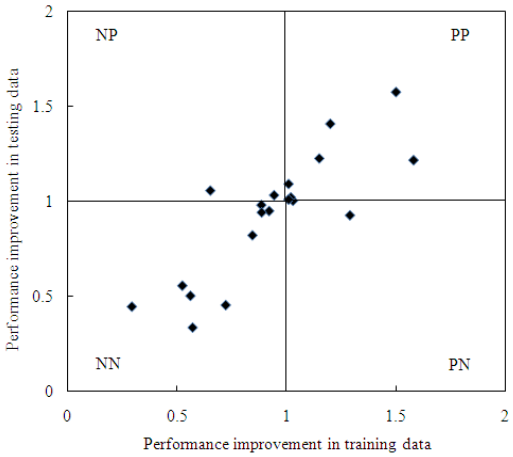


Fig. 1. Comparison of TBSC and Euclidean Distance in Both Training and Testing Data

In order to further analyze the characteristics of data which the proposed approach can achieve satisfied results, a set of measures are proposed to distinguish different datasets.

Complexity: the total distance of each pair of adjacent nodes is defined as complexity (*Com*(*X*)) which is calculated in terms of the following equation:

$$Com(X) = \frac{1}{N-1}\sum_{i=1}^{N-1}(nor(x_{i+1}) - nor(x_i))^2 \quad (6)$$

Each time series is normalized in order to eliminate the gaps of datasets with different dimensions. $nor(x_i)$ is normalized value which can be calculated as:

$$nor(x_i) = \frac{x_i - \min x}{\max x - \min x} \quad (7)$$

Where max x and min x represent the maximum value and minimum value of the time series X, respectively.

Fluctuation: the total distance of each node to the average value (ave *nor*(*x*)) is defined as fluctuation (*Flu*(*X*)) which can be calculated as follows:

$$Flu(X) = \sum_{i=1}^{N}(nor(x_i) - \text{ave } nor(x))^2 \quad (8)$$

Similarly, each time series is normalized at first.

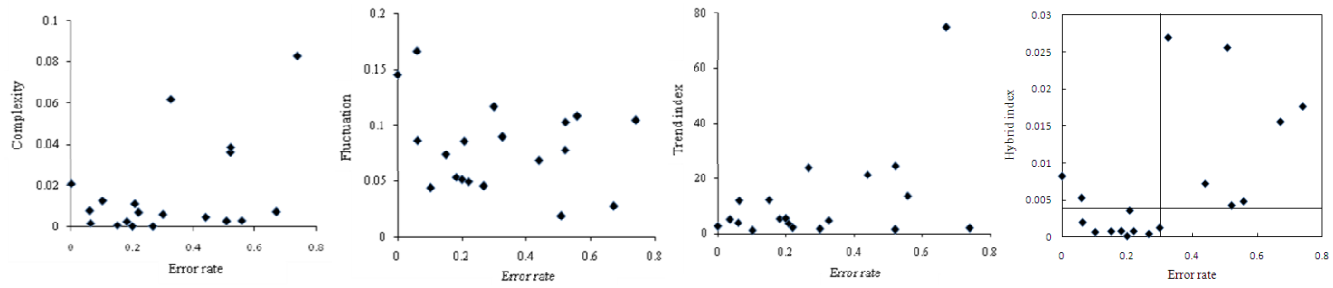
Trend index: the average length of continuous trend pieces is defined as trend index (*TI*). The parameter ε is simply set to 0 when calculating the trend index for every dataset. Giving a transferred time series 00000011111222221111111, *TI*=(6+5+5+7)/4=5.75.

Hybrid index: the result of multiplying three indexes introduced before is defined as hybrid index (*HI*(*X*)) which can be calculated as:

$$HI(X) = Com(X) \times Flu(X) \times HI(X) \quad (9)$$

The performance evaluated by classification error rate is compared with complexity, fluctuation, trend index and hybrid index, respectively. A few datasets with extremely high values are not shown in the figures in order to demonstrate the trend of most datasets. The dataset Coffee (Complexity = 0.54256, Fluctuation = 0.59758, Hybrid index = 1.71838, Error rate = 0.036) is not shown in Fig. 2(a), Fig. 2(b) and Fig. 2(d). The dataset Lightning-2 (Trend index = 439.58, Error rate = 0.508) is not shown in Fig. 2(c). Fig. 2(a) demonstrates that TBSC achieves satisfactory results on most datasets with lower complexity values. Fig. 2(c) shows TBSC can achieve good performance on most datasets with lower trend index. As shown in Fig. 2(d), TBSC achieves good

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)



(a) (b) (c) (d)

Fig. 2. Correlations of TBSC performance and different index. (a) Correlation of performance and complexity (b) Correlation of performance and fluctuation(c) Correlation of performance and trend index (d) Correlation of performance and hybrid index

performance on datasets with lower trend index (lower than 0.004). The performance are not satisfied on datasets with higher trend index (higher than 0.01). The performance are not stable on datasets with middle trend index (from 0.004 to 0.01). That demonstrates a strong correlation between TBSC performance and hybrid index which means hybrid index is helpful for approach selection.

IV. Conclusion

In this paper, we propose a novel similarity calculation approach based on trend information. Experiments are conducted on time series classification by using 20 datasets from different areas. Experimental results show that the proposed TBSC approach outperforms the state-of-art algorithms on 8 of 20 datasets. Another set of experiments demonstrates that appropriate approach can be correctly selected for a particular dataset. The possibility of accuracy improvement by using TBSC is 88.9%, while the possibility of accuracy loss caused by TBSC is only 11.1%. Thus the overall performance can be improved significantly. It should be pointed out that a set of measures are proposed to distinguish different datasets. Correlation between performance of TBSC and hybrid index is relatively strong which reveals another possible way for selecting the right approach.

References

[1] A. Jain, E. Y. Chang, and Y.-F. Wang. 2004. Adaptive stream resource management using kalman filters. In SIGMOD, pages 11–22.

[2] G. Kollios, D. Gunopulos, and V. J. Tsotras. 1999. On indexing mobile objects. PODS, pages 261–272, 1999.

[3] G. Reeves, J. Liu, S. Nath, and F. Zhao. 2009. Managing massive time series streams with multiscale compressed trickles. PVLDB, 2(1):97–108.

[4] E. Keogh, T. Palpanas, V. B. Zordan, D. Gunopulos, and M. Cardle. 2004. Indexing large human-motion databases. In VLDB2004, pages 780–791.

[5] L. Li, J. McCann, N. Pollard, and C. Faloutsos. 2009. Dynammo: Mining and summarization of coevolving sequences with missing values. In KDD, New York, NY, USA, 2009. ACM.

[6] S. Papadimitriou, J. Sun, and C. Faloutsos. 2005. Streaming pattern discovery in multiple time-series. VLDB, 2005.

[7] Hui Zhang, Tu Bao Ho, Yang Zhang and Mao Song Lin. 2005. Unsupervised Feature Extraction for Time Series Clustering Using Orthogonal Wavelet Transform. Journal on Informatics, vol. 30, pp. 305-319.

[8] Xiaozhe Wang, Kate Smith and Rob Hyndman. 2006. Characteristic-Based Clustering for Time Series Data. Journal on Data Mining and Knowledge Discovery, vol. 13, no. 3, pp. 335-364.

[9] A. J. Bagnall, and G. J. Janacek. 2004. Clustering time series from ARMA models with Clipped data. ACM Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 49-58.

[10] Bagnall, Anthony, Janacek, and Gareth, 2005. Clustering Time Series with Clipped Data. Journal on Machine Learning, vol. 58, no. 3, pp. 151-178.

[11] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. Fast Subsequence Matching in Time-Series Databases. In SIGMOD Conference, 1994.

[12] C. Wang and X.S. Wang. 2000. Supporting content-based searches on time series via approximation. In Proc. of the 12th International Conference on Scientific and Statistical Database Management, Berlin, Germany, pp. 69–81.

[13] Y. Huntala, J. Karkkainen, and H. Toivonen. 1999. Mining for similarities in aligned time series using wavelets. In Proc. of the Data Mining and Knowledge Discovery: Theory, Tools, and Technology, Orlando, FL, pp. 150–160.

[14] E. Keogh and P. Smyth. 1997. A probabilistic approach to fast pattern matching in time series databases. In Proc. of the 3rd International Conference on Knowledge Discovery and Data Mining, Newport Beach, CA, USA, pp. 20–24.

[15] E. Keogh and S. Kasetty. 2002. On the need for time series data mining benchmarks: A survey and empirical demonstration. In Proc. of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada, pp. 102–111.

[16] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In KDD Workshop, 1994.

[17] Keogh, E., Xi, X., Wei, L. & Ratanamahatana, C. A. (2006). The UCR Time Series Classification/Clustering Homepage: www.cs.ucr.edu/~eamonn/time_series_data/

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)