Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 54 (2015) 343 – 352

Procedia Computer Science

Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015)

Classification of Time Series Data by One Class Classifier using DTW-D

Kumar Vasimalla[a,*], C. Narasimham[b] and B. Sujith[a]

[a] *Department of Computer Science, CUK, Kasaragod, Kerala 671 316, India*
[b] *Department of CSE, SR Engg. College, Hanumakonda, Telangana 506 371, India*

Abstract

Time series data classification is an important problem and it have number of applications in scientific environment, activity and gesture recognition, anthropology, entomology, sports etc. The most of the research community working on time series classification typically testing their algorithms using datasets available in UCR and other data repositories, which contain labeled training data and test data. But in reality getting labeled time series data is often very difficult and requires some expert help on that domain. But it is possible to get real time series data with one class label. The possible approach to solve this problem is semi-supervised learning algorithm with a special distance measure DTW-D, and compared this approach with semi-supervised learning with Euclidean Distance, semi-supervised learning with Dynamic Time Warping. We showed that our approach is better one compared to other two approaches, and also explained why other approaches have less accuracy. We demonstrate our ideas on diverse real world problems.

Peer-review under responsibility of organizing committee of the Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015)

Keywords: Anthropology; DTW-D; Dynamic time warpping; Time series classification; Semi-supervised-learning.

1. Introduction

Much amount of research is going on in Dynamic Time Warping from last two decades[1,2]. Two related conclusions have been emerged in the community. One, there are number of classification algorithms in the literature. The NN algorithm is particularly suited for unique structure of time series, and particularly all time series classifications use it. Second there are number of distance measures for time series available like, Longest Common Sub-Sequence (LCSS), Dynamic Time Wrapping (DTW). DTW is a technique from the dawn of computing, and is exceptionally complex to beat. And recent research papers conclude that to prove a small improvement in accuracy over DTW require very powerful statistical tests.

From the last decade almost all data mining, pattern recognition and Machine Learning communities using only UCR data sets for their research[3], to check for similarity, clustering and other functions, because most of the datasets available with labelled. Where as in real world obtaining labelled data is very difficult, and it requires some domain expert's assistance. Example in medical[4] field to label ECG data we require doctor help, similarly finance, astronomy and other fields.

*Corresponding author. Tel.: +919567435951/+919848419972.
E-mail address: kumar.sanvi@gmail.com

Peer-review under responsibility of organizing committee of the Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015)
doi:10.1016/j.procs.2015.06.040

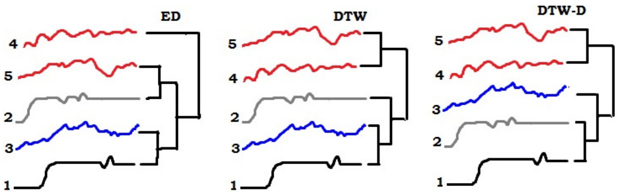

Fig. 1. Clustering of two items from trace dataset with three random walks from left to right ED, DTW, DTW Delta proposed technique.

The solution to this problem is semi-supervised learning. However the direct applications of semi-supervised learning do not work well for time series. In this work we addressed the problem of why semi-supervised algorithms do not work for time – series problems directly, and also we introduce the how we fix this problem so that how well the semi-supervised algorithm works for what type of time series data.

Under certain assumptions unlabeled members of a positive class may be closure to some unlabeled members of diverse negative class than to the labeled positive data. This is true even under DTW. Unlabeled positive data tend to benefit more from using DTW than unlabeled negative examples. The quantity of benefit from using DTW over ED is a meta – feature that can be exploited. We illustrate this in Fig. 1 where we show the hierarchical clustering of five objects under various measures. Two of the five objects are randomly chosen examples (red/bold) from class 3 of the trace dataset. The other three objects are simply random walk time series (blue/light).

Figure 1 shows that Euclidean Distance does poorly here. This is not surprising, since the trace dataset is known to have classes that contain exemplars that are time wrapped versions of prototypical shape. We see that DTW does manage to do better, reducing the distance between trace-1 and trace 2. However this reduction is not enough, random – walk-3 is not close to trace-2, still close to trace-1 only.

Our key observation is that moving from ED to DTW seems to help the true class data more than the random unstructured data. We can encode this difference/delta that DTW makes with DTWD, the ratio of DTW over ED. As we see in figure 1 right, this does produce the correct clustering. Imagine that we had been doing semi-supervised learning in this dataset using just trace-1 as our sole positive example. For both ED and DTW, the very first item we added to the positive set would have been a false positive, and it would be very difficult for any algorithm to recover from this. In contrast, DTW-D would have correctly suggested trace-2 as the net item to add and assuming only that we have good stopping criterion, we would have done very well.

The remaining part of this work is organized as follows, section 2 discuss about related work done in the field of Time series classification and semi-supervised learning, definitions and notations used in this work explained in section 3, section 4 is about distance measure DTW-D and section 5 contains semi-supervised learning algorithm and unlabeled nearest neighbor algorithm details and section 6 is experimental evaluation on projectile pointers and activity recognition data sets, section 7 conclusion and future work.

2. Related Work

From the last two decades an enormous amount of work is one on time series classification. Most of the work/researchers feel that a large amount of labelled training data is available[5]. In reality the high cost of labeling may render such an assumption invalid. For example it requires the time and expertize of cardiologist to annotate individual heart beats in an ECG data trace[6], but a single sleep study test my produce 40,000 such heartbeats. Getting unlabelled data is somewhat easy for example PhysioBank contains over 36000 recordings of digitized polysomnograph and other physiologic signals only a tiny fraction of which are labelled. Likewise tens of thousands of millions books, images and maps and historical manuscripts available in the internet, many of which could be mined in time series space, if only we had more labelled data.

Semi-supervised learning is a learning paradigm useful in application domains in which labelled data are limited and unlabelled data are fully available[7,8]. The literature offers a plenty of SSL methods, among which self – training is the most commonly used. In self – training, a classifier is first trained with a small number of labelled data. It is then used to classify the unlabelled data and adds the most confidently classified objects into labelled dataset the classifier re-trains itself using new labelled set and the procedure repeated until adding the new objects to the labelled set does not increase the accuracy of the classifier or some other stopping criteria is met.

Some of the SSL techniques specially designed for time series are[7–9] Iteratively expand the labelled set by adding the closest object that is classified as positive to the labelled set. The classifier considers all unlabelled as negative and use the Euclidian Distance (ED) has been noted in elsewhere[3]. Proposed to build a SSL classifier using DTW distance, although moving from ED to DTW helps to improve the accuracy of the classifier, the algorithm is not enough accurate still, in most of real time applications. The authors of[4] introduced a SSL technique that interleaves exemplar selection with feature selection, using the work of[2] as a starting point. The method of[4] improves the SSL algorithm, but still uses standard distance measures (ED). As such it is orthogonal to our contribution, which demonstrates that a subtle change in the distance measure dwarfs all possible changes in the algorithms.

In retrospect, only three research efforts on SSL for time series is a surprisingly small number, given that both SSL and time series classification are very popular research topics. In this work we venture to clime that we understand why progress in this area has been so slow. In brief, we affirm that there is little utility in tweaking the architecture of the SSL algorithm for time series, as they are all condemned to perform poorly if they use DTW or ED. The contribution of this work is to show a simple but effective fix that will allow the existing SSL methods to work very well for time series. It is important to recognize that we are not claiming a contribution to SSL algorithms per se. Rather we will show that changing the distance function used in SSL algorithms can produce a remarkable improvement for time series.

3. Definitions and Notations

Here we define all definitions and notations used in this work, for ease of understanding we present notations for Positive Unlabeled learning (PU Learning), which is the collection of SSL methods that trains a classifier based on the positive (labelled) dataset and unlabeled dataset only.

Definition 1. *P is a set of training data including all positive labelled objects.*

P initially contains a small number of labelled objects from a positive class, as learning proceeds the size of P increases by some of the previously unlabeled objects in U are labelled as positive and moved to P. Thus P contains some of the previously labelled objects, as well as the objects chosen by the classifier from the unlabeled dataset.

Definition 2. *U is the set of unlabeled data.*

The dataset U contains objects from both positive as well as negative class. Generally we will expect most of the data is from negative class[10]. The goal of Semi-Supervised method is to map all objects in U to the correct class so that classifier is accurately trained with the classified objects. The classifying process of selecting one object from set U and moving to set P is done iteratively. So we need to stop this classifying based on some stopping criteria, the criteria need to predict whether the algorithm added all unlabeled positive objects to set P or not. The problem of finding stopping criteria is an open problem, with iterative solution based on MDL, Bayesian information criterion, bootstrapping[10,11] etc. As this issue is somewhat orthogonal to our contribution, we simply glass over it here.

However, note that as we shall show in the empirical section, the difference our algorithm makes completely dwarfs any consideration of the optimal stopping criteria. That is to say, even if we did a post – hoc discovery of the optimal stopping criteria for the state – for – art rival, our method would have much higher accuracy for a huge range of sub-optimal stopping values. For brevity, we do not explicitly define time series, ED, DTW, which in any case are rather well known. Instead we use the notation from[2], a heavily cited survey paper on these topics. We do note however the following useful fact, that the ED is an upper bound to the DTW. That is to say, for all x, y we have $DTW(x, y) <= ED(x, y)$.

4. DTW-D

To explain our observations and our key insight, we taken an example. Let us imagine that we have target class of objects that are taken as three time series, the instances may be corrupted by wrapping, different damping rates, noise

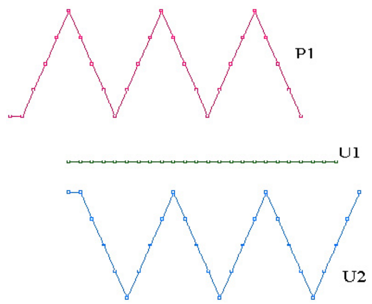

Fig. 2. A labeled dataset P consists of single object $P1$, unlabeled dataset consists of single true negative $U1$, single true positive $U2$.

Table 1. The distance measures of three objects $P1$, $U1$, $U2$ in $[P, U]$ under ED, DTW, and DTW-D.

	ED		
	P1	U1	U2
P1	0	11.48	22.97
U1		0	11.48
U2			0

	DTW		
	P1	U1	U2
P1	0	48	96
U1		0	48
U2			0

	DTW-D=DTW/ED		
	P1	U1	U2
P1	0	4.18	4.17
U1		0	4.18
U2			0

and minor changes as shown in the Fig. 2, they are unambiguously recognizable to the human eye. In this example the negative class consists of just a constant line with the same mean as positive class, corrupted by some noise. Suppose if we ask any SSL algorithm to choose one object from U to add to P using ED as we can see in the Fig. 2 $U1$ is much closer to $P1$ than $U2$ is, thus our SSL algorithm would do poorly here.

$$P1 = 5, 5, 6, 7, 8, 9, 8, 7, 6, 5, 6, 7, 8, 9, 8, 7, 6, 5, 6, 7, 8, 9, 8, 7, 6, 5.$$
$$U1 = 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5.$$
$$U2 = 5, 5, 4, 3, 2, 1, 2, 3, 4, 5, 4, 3, 2, 1, 2, 3, 4, 5, 4, 3, 2, 1, 2, 3, 4, 5.$$

This is not surprising because of the brittleness of ED to even small amounts of wrapping is well known and explains the ubiquity of DTW in most research efforts[12,13]. By finding the optimal peak-to-peak/valley-to-valley alignment between two time series before calculating the distance, DTW is largely invariant to warping, as shown in Fig. 2.

DTW helps significantly as shown in table 1 but it is not enough, U1 is still close to $P1$ than $U2$ is, the SSL algorithm still picks the wrong object to move from U to P. Why did DTW still not solve the problem? While DTW is invariant to wrapping, there are other differences between P1 and U2, including the fact that the first and last peaks have different heights.

DTW cannot mitigate this. However, an examination of distance matrices shown in Table 1 does reveal an interesting fact. Moving from ED to DTW barely changed the distance between $P1$ and $U1$, but it did greatly affect the distance between $P1$ and $U2$. We can codify this with the following observation, by examining the distance matrices shown in Table 1 revel an interesting fact. Moving from ED to DTW barely changed the distance $P1$ and $U1$, but it did greatly affect the distance between $P1$ and $U2$, we can implement with the following observation.

Observation 1: If a class is characterized by the existence of inter – class wrapping (possibly among other distortions), then we should expect that moving from ED to DTW reduces distances more for intra – class comparisons than interclass comparisons. To see this more clearly, we can consider the ratio of distance under DTW and ED as shown in Table 1.

If we consider DTW/ED ratios, we finally have $P1$ and $U2$ appear closer than $P1$ and $U1$. Table 1 visualizes all three distance matrices with complete linkage clustering. We need to consider one miner special case. If the ED is zero then DTW-D would give a divide-by-zero error. Generally we never observe perfect duplicates for real – values objects. It is natural to ask when this phenomenon actually occurs in the real world. In next section we will discuss our assumptions about when our ideas can help.

4.1 Two key assumptions

We do not clime that our ideas will help for all time series problems. In particular, we are making two explicit assumptions, which we will enumerate and discuss below. We will later show that these assumptions are very often true in real world domains.

Assumption 1. The positive class (target class) contain time wrapped version of some platonic ideal (some proto typical shape) possibly with other type of noise/distortions.

Note that this assumption was true of our toy example in Fig. 1. While all members of the trace dataset have some noise, as shown in Fig. 1, the most obvious variability between instances is in the timing of onset of the "ramp-up" and the "oscillation" patterns. DTW is able to compensate for and remove this variability. This ability of DTW to compensate for the inherent warping in this class can produce a dramatic difference in classification accuracy. In the UCR archive dataset the four class trace data set is provided with a 100/100 train test split. The ED error rate on this data set is 0.24, where as DTW has an error – rate of 0.0. Since the exact same splits and classification algorithm (1 NN) were used, and zero parameters are adapted for either approach, all of this difference can be attributed to the superiority of DTW over ED.

Assumption 2. The negative class may be very distinct, and occasionally by chance produces objects close to a member of the positive class, even under DTW.

Empirically, negative classes do tend to be diverse. For example, there are only a limited number of ways an audio snippet can sound similar to a mosquito, but there are infinite ways a sound can be a non-mosquito. Once again this belief was illustrated by our toy example in Fig. 1. The random walk class is naturally very diverse, and it can produce an instance that is closer to trace-1 than the other member of the positive class (trace-2). It is our central clime that if the two assumptions are true for a given problem, our different scoring function DTW-D will be better than either ED or DTW. As there are basic assumptions, we will next consider when we might expect them to be true.

5. Algorithm Details

Our ideas can be applicable to any time series SSL learning framework, by replacing the DTW or ED calculations with DTW-D. Here we define SSL algorithm used in this work. The method is simple but effective fix to problem. My focus in this work is to demonstrate the effectiveness of our ideas.

5.1 DTW-D algorithm

The proposed distance measure DTW-D[12] represented the equation $DTW - D(y) = DTW - D(y)/ED(y) + \varepsilon$. Where ε is extremely small positive quantity used to avoid zero-by-divide error. We re-iterate that ε is not a parameter for our system, it is a device to enable terser definition. The computation of $DTW - D(x, y)$ can be achieved on two series x and y using one line of code.

$$\text{Function distance} = DTW - D(x, y)$$
$$\text{Distance} = DTW(x, y)/ED(x, y) + \varepsilon$$

5.2 Training the classifier

The classifier used is a one-class classifier[14]. The training dataset consists of only one object from positive class; the goal of the classifier is to accurately extract all positive class objects from the unlabeled dataset[15]. The data used to train a classifier is labeled dataset P, unlabeled dataset U. In the beginning there is as few as one labeled object in P. The classifier trains itself through the following steps.

```
ALGORITHM  Train_One_Class_Classifier(P, U, dist, N)
{//P    Initial training dataset with a single training object
// U    The unlabeled dataset
// dist distance function
// N    The number of objects to be moved from U to P
        For i= 1 to n
        {       NNObj = FindUnLNN(P, U, dist);
                P = [P, NNObj];         // update P
                U = [U-NNObj];          //update U

        }
Return p;
}
ALGORITHM FindULNN (P, U, Dist.)
{
        Dist=zeros(1, |U|);
        For i= 1 to |U|
        Dist(i) =min_{j=1,...,|P|}distance(U_i, P_j);
        [~, NN_index]=min(dist);        // closest to P
        NNObj=U_{NN_index};
Return NNObj;
}
```

Algorithm 1.

Step 1: The classifier is trained on initial labeled dataset, which contains as few as only one object from the positive class

Step 2: For each object in the unlabelled dataset U, we compute its distance to the labeled dataset using $DTW - D$.

Step 3: Among all the unlabelled objects, the one we can mostly confidently classify as positive is the object that is closest to the labeled dataset. The object is added to the labeled dataset and removed from the unlabelled dataset. The labeled dataset is changed, so we retrain the classifier using updated labeled dataset. The procedure is repeated until some stopping criteria are met.

The classification process is quite similar to the semi-supervised classification algorithm. However a more careful observation revels that they are fundamentally different. The classifier used in[16] semi-supervised classification is a binary classifier, with all unlabeled objects regarded as training examples from the negative class, where as our classifier is a one class classifier with no training examples from negative class. The advantage of this approach is it makes much more realistic assumptions about how SSL works in practice. In algorithm 1 the training process stops when an unlabeled data set U is exhausted of true positives by DTW-D.

5.3 Evaluating the classifier

To evaluate the accuracy of the classifier we test the classifier using data that is hidden during the training stage. The test dataset contains some positive class objects and many other objects. The intention of the classifier is to exactly extract the all the positive objects from the test dataset. If an instance in the test data set is top k closest to the labeled data set the instance is classified as positive otherwise it is negative. K is the count of positive objects in the test dataset. Thus we can count the number of true positives out of k classification. Precision[17] is calculated with the following equation where N_{Positive} denotes the number of true positives among the top k closest instances.

$$\text{Precision} = N_{\text{Positive}}/K.$$

6. Experimental Evaluation

For all experiments we divide the data into two mutually exclusive datasets. The learning dataset and holdout dataset

- **Learning dataset:** The learning dataset is used in the SSL process to train the classifier. It is divided into labelled dataset P and unlabelled dataset U. The labelled dataset include a single positive example, which is randomly selected true positive object from the learning dataset. The rest of the objects in the learning dataset are regarded as unlabelled objects and are included in U.
- **Holdout dataset:** The holdout dataset are used to test the accuracy of the learned classifier. Objects in the holdout dataset are hidden from the SSL process.

The performance of the trained classifier can be sensitive to the initial training example. To alleviate this sensitivity for each experiment, we iterate the training process by every time starting from a different training example. In particular we allow each positive object in the learning dataset to be used as the starting training instance once, and average the accuracy of the classifier from all runs. To show the changes in the performance of the classifier as the labelled dataset P is gradually augmented, we show the average accuracy for each size of P. That is we evaluated classifier using holdout dataset each time an unlabelled object is added to P. Thus all figures shown below shows holdout accuracy.

For each experiment we compared the performance of three different classifiers[18], the classifier using ED, the classifier using DTW, the classifier using DTW-D. All three classifiers are trained using the same SSL algorithm, the only difference among them is distance function used. As we shall show by simply changing distance function we can improve performance of SSL algorithm for time series by the significant amount.

6.1 Projectile points dataset

Anthropology offers many interesting challenges to data mining, particularly mining of shapes. Examples of shapes which anthropologists may be interested in mining include projectile points (arrow heads) and bones, pottery, petroglyphs. Projectile point's classification is an important topic in anthropology, however labelling of projectile points type is expansive. In this experiment we first convert the shapes of projectile points into time series data using angle – based methodology. We then train the projectile point's classifier with a single training example using our proposed SSL frame work. The Fig. 3 below shows some examples of projectile points used in these experiment two examples from positive class and two examples from negative class.

For this experiment we randomly select 544 projectile points and divide the dataset into two datasets: learning dataset with 136 projectile points, out of which 38 are positive objects, 98 are negative objects. And the hold out dataset with 408 projectile points out of which 114 positive objects, 294 negative objects. We repeated SSL process 40 times each time starting from a different training example for each run we trained three classifiers. The NN classifier using ED, the NN classifier using DTW and NN classifier using DTW-D, the average performance of the three classifiers over 40 runs for each size of positive objects shown in the following Fig. 4.

The Fig. 4 below shows the comparison results of average accuracy of three distinct classifiers, the Nearest Neighbor (NN) classifier using DTW-D distance, the NN classifier using ED distance, the NN classifier using DTW distance, the results are spikey, the labelling of positive and negative is somehow subjective. However the superiority of DTWD classifier over DTW classifier and ED classifier is quite impressive.

The Fig. 5 and 6 above shows the results of the combinatorial experiment. The results shows that DTWD is better than ED and DTW in both selection and evaluation phases.

6.2 Pamap dataset for activity recognition

This dataset contains data of 18 different activities such as *running, rope – jumping, ironing, and vacuum – cleaning* Performed by 9 subjects wearing three Inertial Measurement Units (IMU)[19] we choose the activity of *ascending stairs* as positive class and the remaining as negative class, besides instead of using three sensors for activity recognition, we use only the data against the sensor located on the shoe. The Fig. 7 below shows the average accuracy

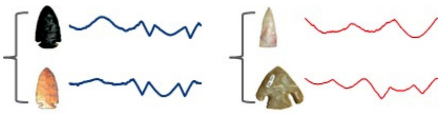

Fig. 3. Example projectile points used in this experiment, and corresponding time series. Left two examples from positive class and right two examples from negative class.

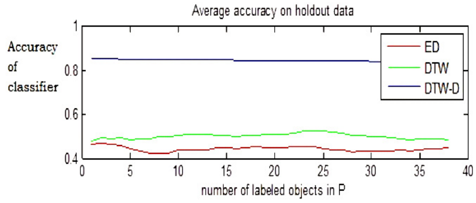

Fig. 4. The average accuracy of three classifiers for different sizes of P, evaluated using holdout dataset.

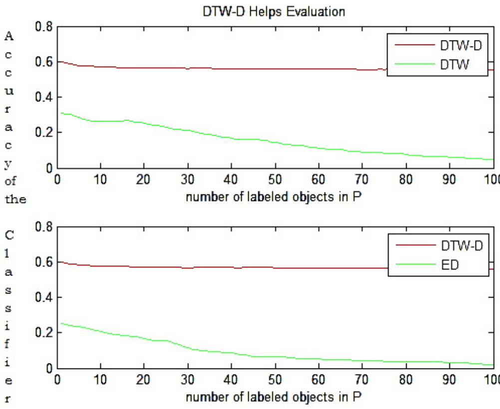

Fig. 5. Comparison of DTW-D with DTW and DTW-D With ED, DTW-D helps the evaluating process by selecting Better NN.

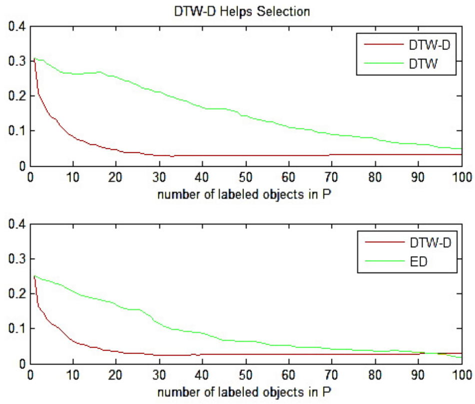

Fig. 6. Comparison of DTW-D with DTW/DTW-D with ED; DTW-D helps the training process by selecting better examples.

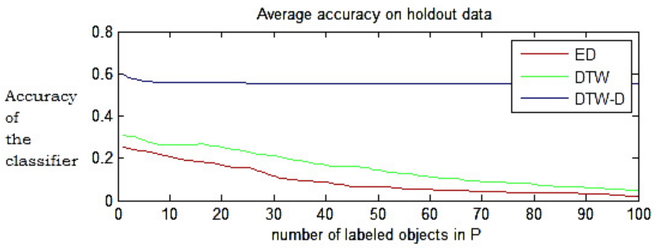

Fig. 7. The average accuracy of three classifiers for different sizes of P, evaluated using holdout dataset.

of three different classifiers when taking ascending – stairs as a positive activity We randomly select 214 positive segments and 694 negative once from the dataset and divide the selected data into two datasets the learning dataset with 100 positive objects and 400 negative objects, and holdout dataset with 114 positive objects and 294 negative objects the SSL process repeated 100 times each time starting from a different training seed. The result shows that the DTW-D classifier starts form higher base – line and continues to improve over the entire range of values. In contrast both ED and DTW start from a lower base – line and eventually get worse.

We remind the reader that as with all experiments in this work the three lines in the Fig. 7 based on identical data, identical conditions and identical algorithms. The only difference is that the distance measure used thus we can safely attribute all improvement attribute to DTWD. Figure 8 and 9 shows comparison of DTW-D with ED and DTW in the selection nearest neighbor and selection of better example.

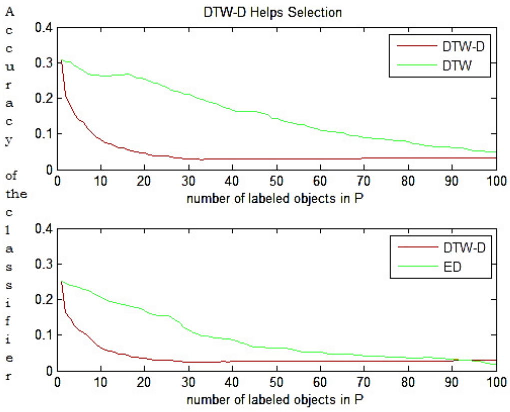

Fig. 8. Comparison of DTW-D with DTW/DTW-D With ED, DTW-D helps the evaluating process by selecting Better NN.

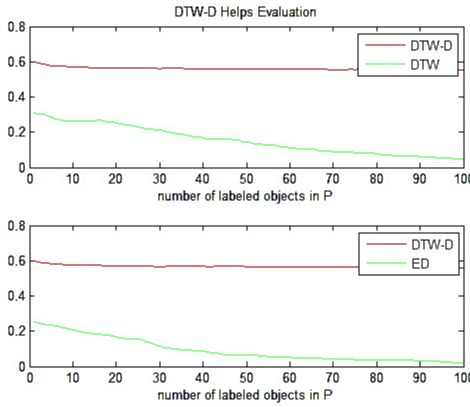

Fig. 9. Comparison of DTW-D with DTW/DTW-D with ED; DTW-D helps the training process by selecting better examples.

7. Conclusion and Future Work

We have introduced a simple idea that dramatically improves the quality of SSL in time series domains. We have conducted our experiments such that all improvements observed can be only attributed to the use of DTWD-D. Our work has the following advantages: It is completely parameter free, thus requires no tuning/tweaking. It allows the use of state of the art indexing methods and fast similarity search methods. The time and space overhead are inconsequential; the coding effort requires only a single line of code to be changed. While we choose a simplest SSL method to demonstrate our ideas, they can trivially be used with any SSL algorithm. Future work includes revisiting the stopping criteria issue in light of DTW-D, and seeing other avenues where DTW-D may be applied.

Acknowledgements

We would like to thank Dr. Professor Eamonn Keogh and Yanping Chen of Department of Computer Science, University of California Riverside for their support of datasets.

References

[1] G. Batista, X. Wang and E. J. Keogh, A Complexity – Invariant Distance Measure for Time Series, *SDM'11*, pp. 699–710, (2011).
[2] 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, *PVLDB*, vol. 1(2), pp. 1542–1552, (2008).
[3] E. J. Keogh, Q. Zhu, B. Hu, Y. Hao, X. Xi, L. Wei and C. A. Ratanamahatana, The UCR Time Series Classification/Clustering Homepage, (2011).
[4] D. Preston, P. Protopapas and C. E. Brodley, Discovering Arbitrary Event Types in Time Series, *Statistical Analysis and Data Mining*, vol. 2(5–6), pp. 396–411, (2009).
[5] M. Raptis, K. Wnuk and S. Soatto, Flexible Dictionaries for Action Recognition, *MLVMA/ECCV*, (2008).
[6] P. Chazal, M. O. Dwyer and R. B. Reilly, Automatic Classification of Heartbeats using ECG Morphology and Heartbeat Interval Features, *IEEE Trans. Biomed. Eng.*, vol. 51, pp. 1196–1206, (2004).
[7] M. A. Ranzato and M. Szummer, Semi-Supervised Learning of Compact Document Representations with Deep Networks, *ICML*, pp. 792–799, (2008).
[8] M. Guillaumin, J. Verbeek and C. Schmid, Multimodal Semi-Supervised Learning for Image Classification, *CVPR*, pp. 902–909, (2010).
[9] S. Cheng, Y. Shi and Q. Qin, Particle Swarm Optimization Based Semi-Supervised Learning on Chinese Text Categorization, *CEC*, pp. 1–8, (2012).
[10] X. Zhu, Semi-Supervised Learning Literature Survey, Technical Report, no. 1530, *Computer Sciences*, University of Wisconsin-Madison, (2005).
[11] A. Sun and R. Grishman, Semi-Supervised Semantic Pattern Discovery with Guidance from Unsupervised Pattern Clusters, COLING (Posters), pp. 1194–1202, (2010).
[12] E. J. Keogh, W. Li, X. Xi, S. Lee, M. Vlachos and LB_Keogh, Supports Exact Indexing of Shapes under Rotation Invariance with Arbitrary Representations and Distance Measures, *VLDB*, pp. 882–893, (2006).

[13] T. M. Rath and R. Manmatha, Word Image Matching Using Dynamic Time Warping, *CVPR*, vol. (2), pp. 521–527, (2003).
[14] D. M. J. Tax. One-Class Classification: Concept-Learning in the Absence of Counter-Examples, (2001).
[15] X. Li and B. Liu, Learning to Classify Text Using Positive and Unlabelled Data, *IJCAI*, pp. 587–594, (2003).
[16] W. Li and E. Keogh, Semi-Supervised Time Series Classification, *ACM SIGKDD*, (2006).
[17] C. J. Van Rijsbergen, Information Retrieval, 2nd Edition, London, England: Butterworths, (1979).
[18] C. A. Ratanamahatana and D. Manichean, Stopping Criterion Selection for Efficient Semi-Supervised Time Series Classification, *SNPD 2012*, vol. 149, pp. 1–14, (2008).
[19] PAMAP, Physical Activity Monitoring for Aging People, www.pamap.org/demo.html, Retrieved 2012-05-12.