Fusion of Similarity Measures for Time Series Classification

Krisztian Buza, Alexandros Nanopoulos, and Lars Schmidt-Thieme

Information Systems and Machine Learning Lab (ISMLL)
University of Hildesheim, Germany
{buza,nanopoulos,schmidt-thieme}@ismll.de

Abstract. Time series classification, due to its applications in various domains, is one of the most important data-driven decision tasks of artificial intelligence. Recent results show that the simple nearest neighbor method with an appropriate distance measure performs surprisingly well, outperforming many state-of-the art methods. This suggests that the choice of distance measure is crucial for time series classification. In this paper we shortly review the most important distance measures of the literature, and, as major contribution, we propose a framework that allows fusion of these different similarity measures in a principled way. Within this framework, we develop a hybrid similarity measure. We evaluate it in context of time series classification on a large, publicly available collection of 35 real-world datasets and we show that our method achieves significant improvements in terms of classification accuracy.

Keywords: time series, classification, fusion, hybrid similarity measure

1 Introduction

One of the most prominent research topics in artificial intelligence, in particular in data-driven decision tasks, is time series classification. Given a series of measured values, like the blood pressure of a patient every hour, the position coordinates of a ballpoint pen in consecutive moments, acoustic or electrocardiograph signals, etc., the task is to recognize which pre-defined group the signal belongs to. In the previous applications these groups could correspond, for example, to words written or said by a person or to the health status of a patient (normal, high or low blood pressure; regular or irregular heart rhythm). In general, besides speech recognition [15], time series classification finds applications in various domains such as finance, medicine, biometrics, chemistry, astronomy, robotics, networking and industry [8].

Because of the increasing interest in time-series classification, various approaches have been introduced ranging from neural and Bayesian networks to genetic algorithms, support vector machines and frequent pattern mining [2]. One of the most surprising recent results is, however, that the simple 1-nearest neighbor (1-NN) classifier using dynamic time warping (DTW) distance [15] has been shown to be competitive or superior to many state-of-the art time-series classification methods [6], [9], [13]. These results inspired intensive research of DTW in the last decade: this method has been examined in depth (for a thorough summary of results see [11]), while the improvements in its accuracy [2], [12] and efficiency [10] allowed to apply it to large, real-word recognition problems.

This success of DTW suggests that, in time series classification, what really matters is the distance measure, i.e. when and why two time series are considered to be similar. DTW allows shifting and elongations in time series, i.e., when comparing two time series t_1 and t_2, the i-th position of t_1 is not necessarily matched to the i-th position in t_2, but it can be matched to some other positions too (that are usually close to the i-th position). By allowing for shifting and elongations, DTW captures the global similarity of the shape of two time series very well. In general, however, many other characteristic properties might be crucial in a particular application, such as similar global or local behavior in the frequency domain, that can be captured by the Fourier or Cosine-spectrum or the Wavelet Transform of the signal [3], [7].

In this paper, we examine this phenomenon in more detail. We consider a set of state-of-the art time series similarity measures and discuss what kind of similarity they capture. As major contribution, we propose a framework that allows fusion of these different similarity measures in a principled way. Within this framework, we develop a hybrid similarity measure. We evaluate our findings in context of time series classification on a large, publicly available collection of 35 real-world datasets and show that our method achieves substantial (statistically significant) improvements in terms of classification accuracy.

2 Related Work

We focus on related works that are most relevant w.r.t. the major contribution, i.e. fusion of similarity measures. For a (short) review of time series similarity measures we refer to Section 3.

There were many attempts to fuse several classifiers by combining their outputs. This resulting structure is often called as an ensemble of classifiers. Besides the simple schemes of majority and weighted voting, more sophisticated methods were introduced such as bagging, boosting [1], [5] and stacking [18]. Ensembles of classifiers have been designed and applied for time series classification in e.g. [16], [19]. In contrast to these works, we aim at fusing the *similarity measure*, instead of working at the level of classifiers' outputs.

One of the core components of our framework is a model for pairwise decisions about whether two time series belong to the same class. Similar models were applied in context of web page clustering [14] and de-duplication [4]. Both of these works, however, aimed at finding equivalent items (whereas the concept of "equivalence" is understood in a broad sense by defining e.g. two web pages as equivalent if they write about the same person). In contrast to them, we work with time series, and, more importantly, we focus on classification.

Fusion of similarity measures is also related to multiple kernel learning [17]. As opposed to [17], we consider time series in a simple and generic framework.

3 Time Series Similarity Measures

In this section we review the most important time series similarity measures. Please note, that throughout this paper we use *similarity measure* and *distance measure* as synonyms. Denoting the i-th position of the time series t by $t(i)$ we can define the *Euclidean Distance* of two time series t_1 and t_2 of length k as: $d_E(t_1, t_2) = \sqrt{\sum_{i=1}^{k}(t_1(i) - t_2(i))^2}$.

The intuition behind *Dynamic Time Warping* (DTW) is that we can not expect an event to happen (or a characteristic pattern to appear respectively) at *exactly* the same time position and its duration can also (slightly) vary. DTW captures global similarity of two time series' shapes in a way that it allows for shifting and elongations. DTW is an edit distance: the distance of two time series t_1 and t_2 of length k, denoted as $d_{DTW}(t_1, t_2)$, is the cost of transforming t_1 into t_2. This can be calculated by filling entries of a $k \times k$ matrix. See also [11], [12].

The *Discrete Fourier Transformation* (DFT) maps the time series t to a set of (complex) coefficients $\{c_f\}_{f=1}^{k}$ that are defined as

$$c_f(t) = \frac{1}{\sqrt{k}} \sum_{i=1}^{k} t(i) e^{-\frac{2\pi j f i}{k}}$$

where $j = \sqrt{-1}$. The Fourier-coefficients $\{c_f\}_{f=1}^{k}$ of t can efficiently be calculated in $\mathcal{O}(k \log k)$ time with the Fast Fourier Transform (FFT) algorithm. DFT captures the signal's periodic behavior by transforming the time series into the frequency domain. If different periodic behavior characterize the time series classes of the underlying application, it is worth to calculate e.g. the Euclidean distance of the Fourier-coefficients $\{c_f(t_1)\}_{f=1}^{k}$ and $\{c_f(t_2)\}_{f=1}^{k}$ of two time series t_1 and t_2: $d_{FE}(t_1, t_2) = \sqrt{\sum_{f=1}^{k}(c_f(t_1) - c_f(t_2))^2}$.

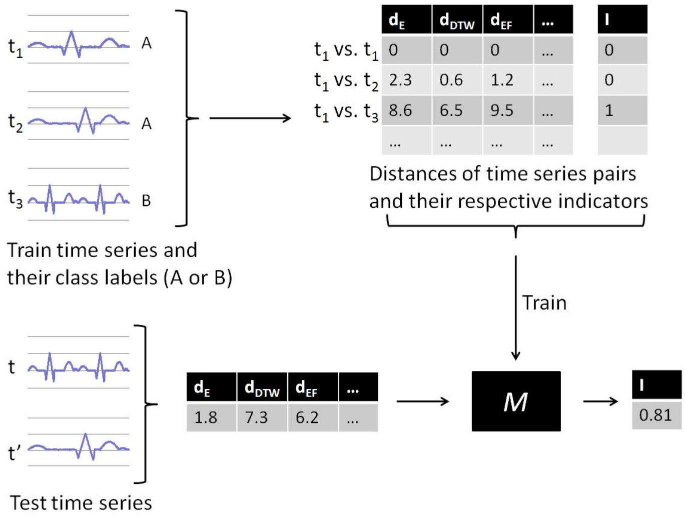
While DFT captures *global periodic* behavior, *wavelets* reflect both *local* and *global* character of a time series [7]. We use the recursive Haar Wavelet decomposition[1] of a time series t that results in a set of Wavelet-coefficients $\{w_i(t)\}_{i=1}^{k}$. Similarly to d_{FE}, we can calculate the Euclidean distance of these Wavelet-coefficients, denoted as d_{WE}.

In order to be able to capture further aspects of similarity, we use the following similarity measures (see [6] and the references therein for a more detailed description): (a) DISSIM that computes the similarity of time series with different sampling rates, (b) distance based on longest common subsequences (LCSS), (c) edit distance on real sequences (EDR), (d) edit distance with real penalty (EPR) that combines DTW and EDR.

4 Fusion of Similarity Measures

In the recent work of Ding et al. [6], none of the examined similarity measures could outperform DTW in general. However, in some specific tasks, one or the

[1] See http://www.ismll.uni-hildesheim.de/lehre/ip-08w/script/imageanalysis-2up-05-wavelets.pdf for an example


t1
A
t2
A
t3
B
Train time series and their class labels (A or B)
Distances of time series pairs and their respective indicators
Train
t
t'
Test time series
M

	d_E	d_{DTW}	d_{EF}	...	I
t_1 vs. t_1	0	0	0	...	0
t_1 vs. t_2	2.3	0.6	1.2	...	0
t_1 vs. t_3	8.6	6.5	9.5	...	1
	...	...	...	...	

d_E	d_{DTW}	d_{EF}	...
1.8	7.3	6.2	...

I
0.81

Fig. 1. Example: fusion of time series similarity measures. A regression model M is trained and its output is used as similarity measure.

other similarity measure worked better than DTW, which is likely to be explained by the fact that different aspects of similarity are relevant in different domains. In the case of simple tasks, one of the similarity measures may capture the relevant aspects of similarity entirely. This best similarity measure can be found based on domain knowledge or by measuring e.g. the leave-one-out classification error on the *train* data for the candidate similarity measures. In more complex cases, however, a single similarity measure may not be sufficient alone. Thus, we need to combine several ones. Such hybridization is often achieved in an ad hoc manner. In contrast, we develop a fusion schema for time series similarity measures that allows to combine similarity measures in a principled way.

In order to distinguish between the similarity measures that we want to combine and the resulting similarity measure, we refer to the former ones as *elementary similarity measures* whereas to the later one as *fused similarity measure*. Our approach for fusion of similarity measures[2] consists of the following steps, see also Figure 1 for an example:

1. For all the *pairs* of time series in the train data, we calculate the similarity values using all the considered elementary similarity measures.
2. In some of the above pairs, both time series belong to the same class, in others they belong to different classes. We define the indicator $\mathcal{I}(t_1, t_2)$ of a pair of time series (t_1, t_2), as follows: $\mathcal{I}(t_1, t_2) = 0$ if t_1 and t_2 belong to the same class, $\mathcal{I}(t_1, t_2) = 1$ otherwise.

[2] Note that we do *not* assume the elementary similarity measures to fulfill specific properties (such as triangular inequality).

3. We train a regression model $\mathcal{M}$. We use the similarity values (see first step) as training data along with the corresponding indicators as labels.
4. We propose to use the output of $\mathcal{M}$ as the fused similarity measure. For a pair of time series (t', t), where either or both of them can be unlabeled (test) time series, we calculate the similarity values using all the considered elementary similarity measures. Then use $\mathcal{M}$ to predict (based on these similarity values) the likelihood that t and t' belong to the different classes. Finally, we use this prediction as the distance of t and t'.

Note that our approach is generic, as this framework allows the fusion of arbitrary similarity measures using various regression models as $\mathcal{M}$. Furthermore, this fused similarity measure can be used by various classification algorithms.

Also note that the above description is just the conceptual description of our approach. While implementing it, one would not separately calculate the similarity of the pair (t_1, t_2) and (t_2, t_1) if the used elementary similarity measure is symmetric. Furthermore, one can pre-calculate and store the similarities of many pairs in case if the classification algorithm (which uses this fused similarity measure) queries the similarity of the same pair several times.

While fusing elementary similarity measures according to the above description, we consider all the pairs of time series. Therefore, if the training data contains n time series, the elementary similarity values are required to be calculated $\mathcal{O}(n^2)$ times and the data used to train $\mathcal{M}$ contains $\mathcal{O}(n^2)$ records. In case of small data sets, this is not a problem. For large datasets, we propose to sample the pairs, and calculate the elementary similarity values only for the sample. In this case, a large enough sample is sufficient for training $\mathcal{M}$.

As mentioned before, in simple domains, one single similarity measure might be sufficient to capture all the relevant aspects of similarity. In such cases, fusion of similarity measures is not necessary and could introduce noise. In order to avoid it, we propose to select the best similarity measure out of some *fused* similarity measures (with different regression models $\mathcal{M}$) *and* all the elementary similarity measures. In order to allow for this selection, we can judge the quality of each similarity measure by its leave-one-out nearest neighbor classification error on the train data.

5 Experiments

Datasets. We examined 35 out of all the 38 datasets used in [6]. We excluded 3 of them (Coffee, Beef, OliveOil) due to their tiny size (less than 100 time series).
Considered similarity measures. We used all the elementary similarity measures described in Section 3. We use two versions of DTW with warping window sizes constrained at 5% and 10% around the matrix diagonal [11] [12].
Comparison protocol. As discussed in Section 1, 1-NN has been shown to be competitive and often even superior to many state-of-the art time series classification algorithms. Therefore, we compare time series similarity measures in context of 1-NN classification. We measure classification error as the misclassification ratio. We perform 10-fold cross validation. For each dataset we test

whether the differences between the performance of our approach and its competitors is statistically significant (t-test at significance level of 0.05).[3]

Baselines. We use two state-of-the art time series classifiers as baselines. The first one is the 1-NN using DTW with window size constrained at 5%. We denote it as DTW. For our second baseline, ELEM, we select the best *elementary* similarity measure based on the leave-one-out classification error on the train data and we use that similarity measure in the 1-NN classifier.

Fusion of Similarity Measures. We produce two fused similarity measures: as $\mathcal{M}_1$ and $\mathcal{M}_2$ we use (i) linear regression and (ii) multilayer perceptron[4] from the Weka machine learning library (http://www.cs.waikato.ac.nz/ml/weka/). After training $\mathcal{M}_1$ and $\mathcal{M}_2$, we select the best similarity measure out of the *fused* and *elementary* similarity measures based on the leave-one-out nearest neighbor classification error on the train data. Finally, we use the selected similarity measure in the 1-NN classifier. This approach is denoted as FUSION.

Results. For many of the examined datasets, the classification task is simple: DTW's error rates are less than 10 %. In these cases, all methods worked equally well, we did not observe statistically significant differences. For the remaining 22 non-trivial datasets, our results are shown in Tab. 1 and summarized in Tab. 2. In Tab. 1 bold font denotes the winner, in case of ties we use italic fonts. Whenever FUSION outperformed any of the baselines, we provide a symbol in form of $\pm/\pm$ where $+$ denotes significance and $-$ its absence against DTW and ELEM respectively. The baselines never outperformed FUSION significantly.[5] Note that we are not concerned with binary classification problems as the number of classes is more than two in most of the cases. In fact, this is one of the reasons why these datasets are challenging and this explains the relatively high error rates.

Discussion. As mentioned before, in simple domains an appropriately chosen elementary similarity measure can lead to very good classification accuracy, whereas a hybrid similarity measure, like the one we introduced, is necessary in more complex cases. Our experimental results show that based on the leave-one-out nearest neighbor classification error of the train data, FUSION could successfully identify those cases where the fusion of similarity measures is beneficial. Therefore, FUSION significantly outperformed DTW in 15 cases and ELEM in 5 cases, while FUSION never lost significantly against the baselines.

[3] In order to save computational time, as discussed in Section 4, for some large datasets we randomly sample the pairs: we calculate similarities in case of Faces and Motes for 10 %, ChlorineConcentration for 5%, Mallat and TwoPatterns for 2%, Yoga and Wafer for 1 %, CinC and StarLightCurves for 0.5 % of all the pairs. In order to ensure fair comparison, we used the same sample of pairs both in our approach and for the baselines.

[4] We used Weka's standard parameter-settings, i.e. learning rate: 0.3, momentum: 0.2, number of train epochs: 500.

[5] We note that in 15 out of the 22 non-trivial datasets $\mathcal{M}_2$ (the similarity measure fused by multilayer perceptron) outperformed $\mathcal{M}_1$ (the similarity measure fused by linear regression). These datasets are: 50words, Adiac, Car, FacesUCR, Haptics, Lighting2, Lighting7, ChlorineConcentration, CinC, InlineSkate, Mallat, StarLightCurves, SwedishLeaf, WordsSynonyms, Yoga.

Table 1. Examined non-trivial datasets, their sizes and classification errors (in %)

Dataset	Size	DTW	ELEM	FUSION	Dataset	Size	DTW	ELEM	FUSION
Haptics	463	**55.9**	57.4	58.7	Star[d]	9236	23.2	20.8	**15.0**+/+
InlineSkate	650	51.4	**45.2**	46.2+/-	Lighting2	121	23.1	*20.6*	*20.6*
Chlorine[a]	4307	50.1	*47.3*	*47.3*+/-	Lighting7	143	**23.1**	25.1	25.1
Yoga	3300	42.5	41.8	**40.1**+/+	Words[e]	905	21.8	22.5	**21.4**-/-
Adiac	781	41.0	40.7	**35.2**+/+	ECG200	200	20.0	*14.5*	*14.5*+/-
Car	120	37.5	32.5	**28.3**+/-	Swedish[f]	1125	17.5	*11.5*	*11.5*+/-
OSULeaf	442	33.9	**21.7**	22.8+/-	FaceFour	112	16.0	**9.8**	11.6
TwoP[b]	5000	32.0	*0.2*	*0.2*+/-	CinC	1420	15.0	7.4	**4.3**+/+
FISH	350	26.6	**16.9**	17.4+/-	Motes	1272	14.0	*7.0*	*7.0*+/-
Medical[c]	1141	24.2	*23.4*	*23.4*-/-	Mallat	2400	11.6	11.6	**10.0**+/+
50words	905	23.4	24.2	**22.4**-/-	FacesUCR	2250	11.6	*7.7*	*7.7*+/-

[a] ChlorineConcentration, [b] TwoPatterns, [c] MedicalImages, [d] StarLightCurves, [e] WordsSynonyms, [f] SwedishLeaf

Table 2. Number of FUSION's wins/loses and ties against DTW and ELEM

	against DTW		**against ELEM**	
	total	significant	total	significant
Wins	20	15	8	5
Ties	0	-	9	-
Loses	2	0	5	0

6 Conclusions and Outlook

Motivated by recent results, in this paper we focused on similarity measures for time series classification. We discussed what aspects of similarity they capture. As in complex applications several of these similarity aspects may be relevant simultaneously, we developed a generic framework which allowed fusion of various similarity measures. In our experiments over a large collection of real-world datasets, we showed that such complex applications exist and our approach achieved statistically significant improvements in those cases.

Our method for the fusion of similarity measures is not limited to time series classification. As future work, we would like to examine fusion of similarity measures in other contexts such as vector data or more complex, structured data. As our approach works on data instance pairs, for large datasets, we aim at exploring sampling strategies with special focus on the possibly imbalanced nature of the pair indicators. We would also like to examine in more depth, if *all* the similarity measures are worth to be fused or one should rather select a subset of them, because many of them could do better (and hopefully faster) than all [20].

References

1. Bauer, E., Kohavi, R.: An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine Learning 36(1), 105–139 (1999)
2. Buza, K., Nanopoulos, A., Schmidt-Thieme, L.: Time-Series Classification based on Individualised Error Prediction. In: International Conference on Computational Science and Engineering. IEEE (2010)
3. Chan, K., Fu, A.: Efficient time series matching by wavelets. In: 15th International Conference on Data Engineering. pp. 126–133. IEEE (1999)
4. Christen, P.: Automatic record linkage using seeded nearest neighbour and support vector machine classification. In: Proc. of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 151–159. ACM (2008)
5. Dietterich, T.: An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. Machine Learning 40(2), 139–157 (2000)
6. Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., Keogh, E.: Querying and mining of time series data: experimental comparison of representations and distance measures. Proceedings of the VLDB Endowment 1(2), 1542–1552 (2008)
7. Hastie, T., Tibshirani, R., Friedman, J.: The elements of statistical learning: data mining, inference, and prediction, 5th Chapter. Springer Verlag (2009)
8. Keogh, E., Kasetty, S.: On the need for time series data mining benchmarks: A survey and empirical demonstration. Data Mining and Knowledge Discovery 7(4), 349–371 (2003)
9. Keogh, E., Shelton, C., Moerchen, F.: Workshop and challenge on time series classification (2007), `http://www.cs.ucr.edu/~eamonn/SIGKDD2007TimeSeries.html`
10. Keogh, E., Pazzani, M.: Scaling up dynamic time warping for datamining applications. In: 6th ACM SIGKDD Int'l. Conf. on Knowledge Discovery and Data Mining. pp. 285–289. ACM (2000)
11. Ratanamahatana, C., Keogh, E.: Everything you know about dynamic time warping is wrong. In: SIGKDD Int'l. Wshp. on Mining Temporal and Seq. Data (2004)
12. Ratanamahatana, C., Keogh, E.: Making time-series classification more accurate using learned constraints. In: SIAM Int'l. Conf. on Data Mining. pp. 11–22 (2004)
13. Rath, T., Manmatha, R.: Word image matching using dynamic time warping. In: Conference on Computer Vision and Pattern Recognition. vol. 2. IEEE (2003)
14. Romano, L., Buza, K., Giuliano, C., Schmidt-Thieme, L.: XMedia: Web People Search by Clustering with Machinely Learned Similarity Measures. In: 2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference (2009)
15. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. Acoustics, Speech and Signal Processing 26(1), 43–49 (1978)
16. Schuller, B., Reiter, S., Muller, R., Al-Hames, M., Lang, M., Rigoll, G.: Speaker independent speech emotion recognition by ensemble classification (2005)
17. Sonnenburg, S., Ratsch, G., Schafer, C., Scholkopf, B.: Large scale multiple kernel learning. The Journal of Machine Learning Research 7, 1531–1565 (2006)
18. Ting, K., Witten, I.: Stacked generalization: when does it work? In: 15th Int'l. Joint Conf. on Artifical Intelligence-Vol. 2. pp. 866–871. Morgan Kaufmann (1997)
19. Zhang, G., Berardi, V.: Time series forecasting with neural network ensembles: an application for exchange rate prediction. Journal of the Operational Research Society 52(6), 652–664 (2001)
20. Zhou, Z., Wu, J., Tang, W.: Ensembling neural networks: Many could be better than all. Artificial intelligence 137(1-2), 239–263 (2002)