Constructing High Dimensional Feature Space for Time Series Classification

Victor Eruhimov, Vladimir Martyanov, and Eugene Tuv

Analysis & Control Technology, Intel,
5000 W Chandler Blvd, Chandler AZ85226, USA

Abstract. The paper investigates a generic method of time series classification that is invariant to transformations of time axis. The state-of-art methods widely use Dynamic Time Warping (DTW) with One-Nearest-Neighbor (1NN). We use DTW to transform time axis of each signal in order to decrease the Euclidean distance between signals from the same class. The predictive accuracy of an algorithm that learns from a heterogeneous set of features extracted from signals is analyzed. Feature selection is used to filter out irrelevant predictors and a serial ensemble of decision trees is used for classification. We simulate a dataset for providing a better insight into the algorithm. We also compare our method to DTW+1NN on several publicly available datasets.

1 Introduction

The problem of time series classification (TSC) has attracted a lot of attention from the machine learning society in the past decade. Many domains such as computer vision, medicine, biology, manufacturing, and others possess time dependencies as natural problem descriptions, as opposed to individual features extracted from signals. A challenge in working with signals as class predictors is large amount of features and complex dependence of the signal class on these features. Advances in supervised learning methods that allow to work with ultra high dimensional feature space make TSC a very appealing problem. However the Euclidean metric together with One-Nearest-Neighbor (1NN) classifier has proven to be one of the most robust TSC methods. A generalization of this approach that takes into account transformations of time axis has been introduced about a decade ago. [1] suggested a similarity measure called Dynamic Time Warping (DTW) that is based on matching two signals with dynamic programming. Later [2] showed that the complexity $O(n^2)$ of DTW for matching two signals of length n can be reduced to $O(n)$ by constraining the search path without sacrificing accuracy. DTW was proved to be the best state-of-the art technique in multiple domains, "1NN with DTW is exceptionally hard to beat" [3]. We will not provide a full review of TSC methods due to limited space, an extensive survey is available in [4]. A large group of papers is devoted to extracting generic features from signals and transforming a TSC problem into a classical machine learning problem of predicting signal class from a given feature set. A list of features includes Singular Value Decomposition features, Discrete Fourier

J.N. Kok et al. (Eds.): PKDD 2007, LNAI 4702, pp. 414–421, 2007.

Transform, coefficients of the decomposition into Chebyshev Polynomials, Discrete Wavelet Transform, Piecewise Linear Approximation, ARMA (AutoRegression Moving Average) coefficients, various symbolic representations. Each of the methods has its own faults. Euclidean/DTW based methods suffer from the curse of dimensionality – 1NN is known to perform poorly on high-dimensional problems (i.e. long signals) [5]. [6] shows superior performance of a boosted tree ensemble learned on a set of generic features compared to 1NN with Euclidean distance on datasets where time warping is not needed. This paper is devoted to a generalization of this method for the case when time warping is essential for classifying signals.

The essense of the method is to transform time axis of both train and test signals so that the same salient points appear at the same time moments. Then we can apply a generic feature extraction method described in [6]. We sample one signal from each class that we call a base signal. Then we use DTW to warp time axis of every time series to each of base signals, resulting in several time series, one per class. A generic set of features – wavelets, coefficients of the decomposition into Chebyshev polynomials, statistical moments – and several DTW-specific features are extracted from each warped signal. A joint set of features is used as predictors. The number of features could be very high – from hundreds to tens of thousands. Such high dimensional representations are hard to learn from. However if we reduce the feature set we run into a risk of loosing information about the signal and increasing classification error. Recent advances in feature selection methods [7,8] allow us to learn a boosted ensemble of trees with a built-in feature weighting method directly in the original high-dimensional feature space. We show that this method is comparable or superior to DTW on several UCR datasets [9]. We also analyze the performance of the method on simulated data to better understand its pros and cons.

The outline of the paper is as follows: Section 2 is devoted to the warped feature extraction algorithm, Section 3 describes our time series generator and Section 4 discusses experimental results on UCR and simulated data.

2 Warping Time for Feature Extraction

Generic features described in [6] such as wavelets and Chebyshev coefficients are not invariant to time warping that changes position and scale of signal features differently for each signal. Statistical moments do not change much with warping as they take into account only the distribution of signal values but in many cases they are weak predictors. We want to build a generic set of features that would work on signals with arbitrarily (with reasonably low loss of information) warped time.

The general idea of the method that we discuss here is to select a base signal and transform time axis of each signal to minimize DTW distance. Then (when all salient features are aligned) we can extract generic features and learn a classifier. But we cannot select a single base signal because transforming all signals to it could cause deformation of class-dependent signal profiles that are crucial

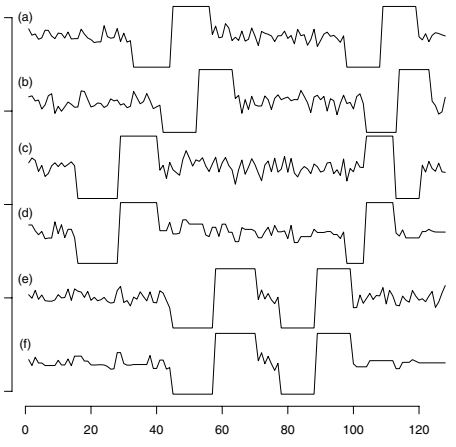

Fig. 1. Example of warped signals from two_patterns [9] dataset: (a) test signal, (b) the DTW-closest training sample, (c) base signal b_2 from class 2, (d) warped test signal wrt b_2, (e) base signal b_4 from class 4, (f) warped test signal wrt b_4

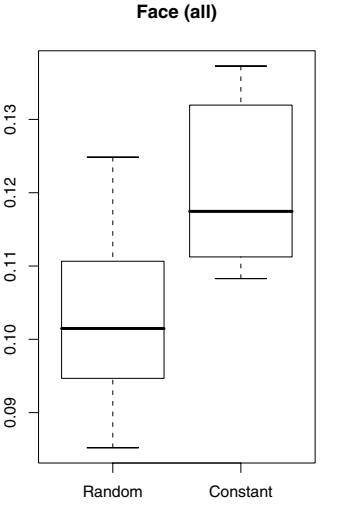

Fig. 2. Distribution of test errors for **Face(all)** dataset for randomly chosen (left boxplot) and fixed (right boxplot) base signals

for classification. So we choose one base signal from each response class. For each signal s and base signal b we find a point-to-point correspondence with DTW and calculate a warped version of s by averaging all values of s corresponding to each point of b. We use DTW algorithm described in [2]. An example of warping is given in Figure 1. We have taken a test signal (a) from the UCR dataset **Two_patterns**, class 4, and warped it to base signals of classes 2 (base signal (c) and warped test signal (d)) and 4 ((e) and (f) correspondingly). The pair of signals (e) and (f) illustrates the alignment of warped signal from the same class. This allows us to extract meaningful features from warped signals characterizing class-dependent signal profiles.

We extract a set of generic features to be used as class predictors from each warped signal. Also, we keep the features from the original unwarped signal in case no time warping is necessary. The essense of the approach is to use as many features that *could be* important as possible, if they are irrelevant, feature selection algorithm will filter them out. The exact description of feature selection method is given in Algorithm 2. We do not use any warping window when transforming signals. We do use a Sakoe-Chiba band [10] when calculating the DTW+1NN (see 2ef of Algorithm 2), the band width is obtained by optimizing DTW+1NN leave-one-out error on the training part of data. Signal warping is described in Algorithm 2. Note that we use the class predicted by DTW+1NN as a feature so the test error can hardly be larger than that of DTW+1NN. We also use DTW distances to base signals as predictors to supply GBT with an

additional information about the features from different base signals. The total number of features that we extract is equal to $C \cdot (W + L + Ch + 1) + W + Ch + 6$, where C is the number of classes, L is the signal length and W is minimum power of 2 greater than L. In order to extract wavelets we add $W - L$ zeros to each signal making Discrete Wavelet Transform applicable. Ch is the number of Chebyshev coefficients – we filter out higher coefficients that proved to be too noisy, we use the value $Ch = 20$ throughout the paper.

Algorithm 1. Warped time feature selection

1. For each class c randomly choose a base signal b_c endfor.
2. For each signal s
   a. feature set $F_s = \{\}$
   b. for each class c
      warp the signal wrt base signal $s_c^{(w)} = Warp(s, b_c)$
      add wavelet D8, Chebyshev coefficients and
        raw features (signal values) of $s_c^{(w)}$ to F_s
      calculate DTW distance from $s_c^{(w)}$ to b_c and add to F_s
   c. endfor
   d. calculate statistical moments (mean, variance, skewness, curtosis,
      and maximum value) and add to F_s
   e. find a signal s_m from the training set D_T such that $s_m = \underset{s_m \in D_T \setminus \{s\}}{\operatorname{argmin}} DTW(s, s_m)$
   f. add the class of s_m as a feature to F_s
   g. add wavelet D8 and Chebyshev coefficients of s to F_s
3. endfor

Algorithm 2. Signal warping $Warp(b, s)$

1. Run DTW for a base signal b and an input signal s.
   Let L_i be the list of elements from s corresponding to the element i from b
2. For each i
      set the i-th value of the warped signal to the average of values in L_i
3. endfor

---

3 Time Series Dataset Generator Description

We used a data generator designed specifically to mimic most of the challenges we face in the real environment (semiconductor manufactuirng signals classification) and to better investigate TSC methods by having insight into signal class nature. Each time series is a trapezoid-like parameterized function (a sample signal is shown in Figure 5). 9 parameters (left node position, horizontal and vertical shifts, right node position, horizontal and vertical shifts, oscillation amplitude,

frequency and phase, left and right slope curvatures) are sampled from predefined distributions for each signal. Curvatures and oscillation amplitude are used to generate a numeric response that is a sum of linear and quadratic functions of parameters:

$$y_n = AV + V^T BV + \varepsilon. \tag{1}$$

Here A is a vector $1 \times N$, B is a matrix $N \times N$, V is a vector of $N = 3$ parameters and ε is Gaussian noise. The values of A and B are taken from a uniform distribution $U(0, 1)$ before we start generating any time series. Categorical response is

$$\begin{aligned} y &= 1(y_n - median(y_n)), where \\ 1(x) &= \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases} \end{aligned} \tag{2}$$

Note that some parameters (such as the phase of oscillation sampled from the $U(0, 2\pi)$) do not participate in the response but have considerable influence on signals. It is a challenge for any predictive engine to recover this functional relationship due to the complex dependence of time series on V and the problem dimensionality.

In order to make things more complex, we add a random (from 0 to 16) amount of zeros to the beginning of each signal. We will refer to the dataset without such random shifts as to **Quad1**, and to the dataset with random shifts as to **Quad1S16**.

4 Experimental Results

We test our TSC method on several UCR datasets and on the simulated datasets in order to better understand how warping works. We have selected a subset of UCR datasets that have more than 30 samples per class on average. Smaller amount of training samples would produce higher noise and would require a more accurate approach to feature selection. Our implementation of GBT is very close to [11] with feature weighting [7] on top of it. All parameters of GBT learning algorithm were fixed: the number of trees $N = 2000$, shrinkage $\nu = 0.02$, subsampling parameters and probability thresholds. Each tree was trained on a randomly chosen 60% portion of the training dataset, the probability threshold was equal to 0.5.

Figure 2 shows the distributions of test errors on one of UCR datasets when we randomly choose base signals (left boxplot) and when we keep base signals fixed (right boxplot), so the variation of test errors is mostly due to GBT, and one can see that the particular choice of base signals is not crucial.

We run the algorithm on each dataset 10 times with randomly chosen base signals and GBT random seed. The results are summarized in Table 1. One can see that we are almost always superior to DTW+1NN or comparable in the case when the problem is easy enough for DTW+1NN and the absolute number of misclassified samples is very low. In order to check how important our warping features are, we run a set of experiments with zero warping window

Table 1. Test errors

Dataset	DTW+1NN test error	Average Test Error	Standard Deviation of Test Error	p-value for warped features
Quad1	0.108	0.0572	0.0018	1
Quad1S16	0.148	0.0855	0.0027	$4.1 \cdot 10^{-8}$
Wafer	0.005	0.00259	0.000453	$2.2 \cdot 10^{-2}$
Yoga	0.155	0.150	0.00219	0.93
Swedish_Leaf	0.157	0.118	0.00394	1
Face(all)	0.192	0.103	0.0122	$1.3 \cdot 10^{-10}$
Synthetic_Control	0.017	0.00233	0.00260	$5.2 \cdot 10^{-5}$
ECG	0.12	0	0	1
OSU_Leaf	0.384	0.379	0.0130	$2.4 \cdot 10^{-2}$
Two_patterns	0.0015	0.00055	0.000384	$1.9 \cdot 10^{-5}$

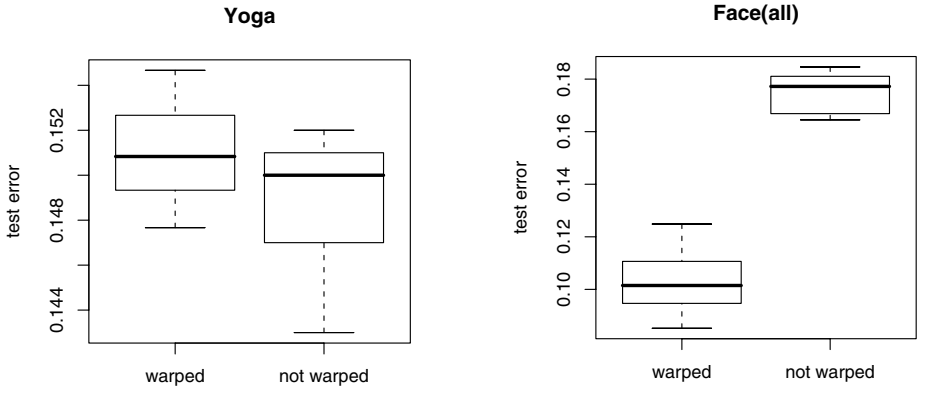

Fig. 3. Distribution of test errors for **Yoga** and **Face(all)** with and without warped features used as predictors

size (keeping the predicted class by DTW+1NN the same) which means that we do not transform signals at all. A one-sided t-test was used to check if test error with warped features is less than test error without warping. The corresponding p-values are given in the last column of Table 1. The improvement in test error is visible for 6 datasets out of 10. There are datasets such as **Quad1** where we did not get any improvement since warping is not important there. Figure 3 illustrates the distributions of test errors with and without warped features for **Yoga** and **Face(all)**. One can see that we get a significant decrease in test error on **Face(all)** when we use warped features. **Yoga**, however, shows a slight increase in test error, most probably due to failure of feature selection to filter out all irrelevant features.

Figure 4 presents the dependence of test error on the width of Sakoe-Chiba band for warping signals (it is important that the feature corresponding to the class predicted by DTW+1NN is kept constant for this experiment corresponding to the optimal width of the band). Note that the dependence of our algorithm

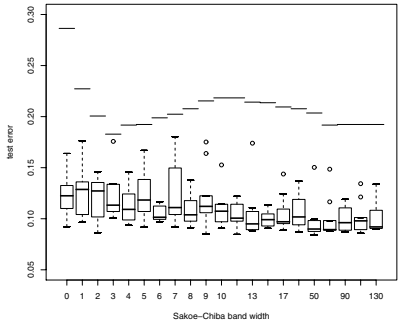

Fig. 4. The dependence of test errors on the Sakoe-Chiba band width used for signal warping. The boxplots correspond to GBT test errors, horizontal lines — to DTW+1NN test errors.

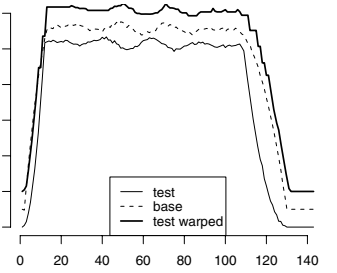

Fig. 5. Example of warped signal from **Quad1S16**. The signals are shifted along the vertical axis 10% from each other due to strong overlapping.

test error on the band width is different from the dependence of DTW+1NN test error. Our interpretation of this effect is that DTW+1NN considers every signal in the training dataset for matching while our approach matches only base signals. So DTW+1NN has higher chances of a correct match with smaller window. This is why we did signal warping without any band restriction. Note that this does not pose computational problems as with DTW+1NN since the latter has to match a test signal with all training time series while we match it with only base signals. The diminishing trend in Figure 4 also shows that the features obtained from warping signals are important for classification. The class predicted by DTW+1NN is also very important – by removing just this one feature from the predictors list of the **Face(all)** dataset we increase the average test error from 0.118 up to 0.179 – a 50% difference!

Quad1 and **Quad1S16** allow us to get an insight of the algorithm weak spots. Going back to Section 3, curvatures of the signal front and back are used to generate response for these datasets. Curvature is not invariant to the transformation that we apply to signals as illustrated by Figure 5. Note the difference in curvatures of test and warped signals in the right part, around time value 120. This means that the information about curvature will be lost in the warped signal and hence will not be reflected in extracted features. Features from the original signal do not help much either since there is a random shift and curvature features are scattered in time corresponding to different wavelet features. The resulting dependence is hard to learn, this is the major reason for 50% difference in test errors for **Quad1** and **Quad1S16**.

5 Conclusion

This work deals with TS classification problems where input signals need to be aligned in time (warped). The proposed approach creates a massive num-

ber of features including original signals, by-class warped signals, wavelet and chebychev decomposition coefficients of warped signals, summary statistical moments of warped signals, and even labels predicted by DTW-1-NN used as input features. Gradient boosting of trees with imbedded dynamic feature selection capable of handling hundreds of thousands predictors is then used for classification. A set of experiments on UCR and artificial datasets show that this combination provides a superior learner relative to the well know state of the art approach. No single subset of features by itself carries enough information to achieve the best performance on different classification tasks. The future work will concentrate on refining this approach for important industrial applications with influential curvature-like features not easily detected currently by any method, and porting the methodolgy to time series regression problems.

References

1. Berndtand, D.J., Clifford, J.: Using dynamic time warping to find patterns in time series. In: Working Notes of the Knowledge Discovery in Databases Workshop, pp. 359–370 (1994)
2. Ratanamahatana, C.A., Keogh, E.: Everything you know about dynamic time warping is wrong. In: Third Workshop on Mining Temporal and Sequential Data, in conjunction with the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, New York (2004)
3. Xi, X., Keogh, E., Shelton, C., Wei, L., Ratanamahatana, C.A.: Fast time series classification using numerosity reduction. In: International Conference on Machine Learning (2006)
4. Keogh, E.: Data mining and machine learning in time series databases (2004)
5. Hastie, T., Tibshirani, R., Friedman, J.: The elemetns of statistical learning: Data mining, inference, prediction. Springer, Heidelberg (2001)
6. Eruhimov, V., Martyanov, V., Tuv, E.: Feature class selection for time series classification. In: Submitted to the Workshop on Time Series Classification, SIGKDD'07
7. Borisov, A., Eruhimov, V., Tuv, E.: Dynamic soft feature selection for tree-based ensembles. In: Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L. (eds.) Feature Extraction, Foundations and Applications, Springer, New York (2006)
8. Borisov, A., Torkkola, K., Tuv, E.: Best subset feature selection for massive mixed-type problems. In: Corchado, E., Yin, H., Botti, V., Fyfe, C. (eds.) IDEAL 2006. LNCS, vol. 4224, pp. 1048–1056. Springer, Heidelberg (2006)
9. Keogh, E., Xi, X., Wei, L., Ratanamahatana, C.A.: The ucr time series classification/clustering homepage (2006)
10. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans. Acoustics, Speech, and Signal Proc. ASSP-26, 43–49 (1978)
11. Friedman, J.H.: Stochastic gradient boosting. Technical report, Dept. of Statistics, Stanford University (1999)