CORE

Metadata, citation and similar papers at core.ac.uk

Provided by Biblioteca Digital da Produção Intelectual da Universidade de São Paulo (BDPI/USP)

SISTEMA INTEGRADO DE BIBLIOTECAS
UNIVERSIDADE DE SÃO PAULO

Universidade de São Paulo

Biblioteca Digital da Produção Intelectual - BDPI

Departamento de Ciências de Computação - ICMC/SCC

Artigos e Materiais de Revistas Científicas - ICMC/SCC

2015

# Time series classification with representation ensembles

Lecture Notes in Computer Science, Cham, v.9385, p. 108-119, 2015
http://www.producao.usp.br/handle/BDPI/50099

*Downloaded from: Biblioteca Digital da Produção Intelectual - BDPI, Universidade de São Paulo*

# Time Series Classification with Representation Ensembles

Rafael Giusti(✉), Diego F. Silva, and Gustavo E.A.P.A. Batista

Instituto de Ciências Matemáticas e de Computação, São Carlos, Brazil
{rgiusti,diegofsilva,gbatista}@icmc.usp.br

**Abstract.** Time series has attracted much attention in recent years, with thousands of methods for diverse tasks such as classification, clustering, prediction, and anomaly detection. Among all these tasks, classification is likely the most prominent task, accounting for most of the applications and attention from the research community. However, in spite of the huge number of methods available, there is a significant body of empirical evidence indicating that the 1-nearest neighbor algorithm (1-NN) in the *time domain* is "extremely difficult to beat". In this paper, we evaluate the use of different data representations in time series classification. Our work is motivated by methods used in related areas such as signal processing and music retrieval. In these areas, a change of representation frequently reveals features that are not apparent in the original data representation. Our approach consists of using different representations such as frequency, wavelets, and autocorrelation to transform the time series into alternative decision spaces. A classifier is then used to provide a classification for each test time series in the alternative domain. We investigate how features provided in different domains can help in time series classification. We also experiment with different ensembles to investigate if the data representations are a good source of diversity for time series classification. Our extensive experimental evaluation approaches the issue of combining sets of representations and ensemble strategies, resulting in over 300 ensemble configurations.

## 1 Introduction

Undoubtedly, analysis of time series data has attracted an enormous amount of attention in recent years. Time-oriented data are present in several application domains including medicine (*e.g.*, electrocardiography and electroencephalography), engineering (sensor data), entertainment (motion capture data in video games), meteorology (climate data), etc. The research community has answered to such demand with literally thousands of data analysis methods for diverse tasks such as classification, clustering, prediction, and anomaly detection.

This work partially funded by grant #2012/08923-8, #2013/26151-5, and #2015/07628-0, São Paulo Research Foundation (FAPESP); and CNPq #446330/2014-0 and #303083/2013-1.

© Springer International Publishing Switzerland 2015

E. Fromont et al. (Eds.): IDA 2015, LNCS 9385, pp. 108–119, 2015.
DOI: 10.1007/978-3-319-24465-5_10

Classification is likely the most prominent task in time series mining, accounting for most of the applications and attention from the research community. But in spite of the huge number of methods available, there is a significant body of empirical evidence indicating that the 1-nearest neighbor algorithm (1-NN) in the *time domain* is "extremely difficult to beat" for classification tasks [8,18].

In this paper, we evaluate the use of different data representations in time series classification. Our work is motivated by methods used in related areas such as signal processing and music retrieval. In these areas, a change of representation, for instance, from time to frequency or cepstrum, often reveals features that are not apparent in the original data representation. This approach consists of using different representations to transform the time series into alternative decision spaces. A classifier is then used to provide a classification for each test time series in the alternative domain.

Our goal is to investigate how features provided in different domains can help in time series classification. We perform our experiments using the 1-nearest neighbor classifier (1-NN), due to its simplicity and effectiveness in time series classification. The use of a single classification model helps us to rule out the differences in performance due to different classification algorithms. This way, any difference of performance can be attributed on the change of data representation. We also evaluate different ensembles of 1-NN classifiers to investigate if the data representations are good sources of diversity for time series classification.

Although some recent research has addressed the classification of time series using different representations [4,16], our paper is unique in the sense that we evaluate representations irrespectively of classification models. Moreover, we employ a more diverse set of representations and ensemble strategies. Our extensive experimental evaluation approaches the issue of combining sets of representations and ensemble strategies, resulting in over 300 ensemble configurations.

The paper is organized as follows. In Sect. 2 we present an overview of time series classification, time series (dis)similarity and transformation of time series. In Sect. 3 we present notions of ensembles of classifiers and the ensemble configurations used in this work. In Sect. 4 we present our experimental evaluation and discuss our results. Finally, in Sect. 5 we present our conclusions and future work.

## 2 Time Series Classification and Transformation

Let a time series of length $m$ be an ordered set of values $S = (s_1, s_2, \ldots, s_m)$, $s_t \in \mathbb{R}$ for all $t \in [1, m]$. Each value $s_t$ of $S$ is an observation of the time series at instant $t$ and every pair of consecutive observations $(s_t, s_{t+1})$ is considered equally spaced in time – *i.e.*, the series is uniformly sampled or the sampling rate can be otherwise disregarded.

Time series classification is an important problem that arises in many practical applications. It consists in assigning a class label $C_{\mathbf{x}}$ to a previously unseen example $\boldsymbol{x}$ that is somehow related to the process that produced the time series $\boldsymbol{x}$. The problem of time series classification has attracted great interest from the scientific community. Several approaches are readily available to tackle this problem. One may extract features from the temporal data and use these features to train a classification model, such as the support vector machine [5]. In a different approach, representative segments – shapelets – of temporal data may be extracted from a training data to construct a similarity-based decision tree [19]. However, one of the most popular approaches consists of using the original temporal data as attributes for the $k$-nearest neighbors classification model.

The $k$-nearest neighbors classifier – $k$-NN – is an instance-based classification model. It is built on the nearest-neighbors rule, which roughly translates to the notion that similar instances belong to the same class with high probability. Although simple, the 1-NN classifier with DTW (1-NN-DTW) is repeatedly reported as the best classification model in the average case for time series classification, being considered to be "exceptionally difficult to beat" [8,18].

The concept of (dis)similarity between time series is often estimated with a distance function. The Euclidean distance is a widely used function to estimate the dissimilarity between two time series. A relevant issue of the Euclidean distance is the fact that it is very sensitive to non-linear variations in the time axis known as *warping* [12]. The Dynamic Time Warping (DTW) is a local-warping invariant of the Euclidean distance which minimizes the estimated distance between two time series by finding an optimal alignment between their observations. This alignment promotes the matching of values observed at different relative times under the following constraints: (*i*) the observations must be monotonically ordered with respect to time; (*ii*) the alignment should begin in the first and end in the last observations of both time series; and (*iii*) each value must belong to the optimal alignment – *i.e.*, no value is skipped in any of the time series.

The sampled observations of a time series are a description of how a measurable phenomenon changes with time. Such series is said to be represented in the time domain. "Traditional" classification of time series with the 1-NN classifier is performed in the time domain. However, it is possible to transform a time series to an alternate domain of representation. We define a transformation of time series as a mapping from the time domain to an alternate space of decision. To classify instances with the 1-NN classifier in the transformed space, it suffices to wrap the distance function so that each instance is transformed before being compared. In this work, we transform the time series from the time domain to a different decision space and perform the classification on this new decision space using the 1-NN classifier normally. For certain domains of application, this allows for great classification accuracy. We construct ensembles of the 1-NN classifier for even better results.

There is a huge diversity of time series transformations in the scientific literature [18]. In this work, we attempted to choose transformations that actually provide a distinct decision space that is not a summarization of the original temporal data. Therefore, we have excluded some classical dimensionality reduction techniques, such as Principal Component Analysis [10], SAX [13] and SAX-based transforms [14].

**Power Spectrum:** The Discrete Fourier Transform – DFT – decomposes a time series as an ordered set of sinusoids of decreasing frequency. Each value of the transformed series is a complex number that encodes both amplitude and phase of a periodic component.

The DFT has been used for a long time as a strategy of dimensionality reduction that allows for efficient indexing of time series [1].

We define the power spectrum representation of a time series as the plot of the complex modulus of its Fourier components. The power spectrum gives the "energy" of the time series in the frequency bands associated with the Fourier components. Periodical trends in the time series may be exposed by the power spectrum, even if the original time series contains noise.

For a detailed and the formal definition of Fourier transform, we refer the reader to [2].

**Discrete Wavelet Transform:** The DFT concerns only about frequency, not adding any information about when each frequency component is present in a time series. In order to mitigate this problem, wavelet transform creates a time-frequency representation with different resolutions. This is done by calculating the spectrum with sliding windows of different sizes. Discrete wavelet transform (DWT) is a discrete version of the wavelet transform for numerical signals.

Another relevant difference between DWT and DFT is that the latter represents a composition of sinusoidal waves. In the other hand, DWT may work with an infinite number of functions, called mother wavelets.

There is a plethora of works that use DWT in time series classification. A particularly interesting application of this approach was made for matching stock time series [7], in which the authors used the Haar's mother wavelet [6].

**Autocorrelation Function:** The autocorrelation of a time series measures its predictability at a specific instant from its previous observations. A highly autocorrelated time series is indicative of a very deterministic process while true white noise shows no autocorrelation for pairs of distinct observations. The sample autocorrelation is typically used as an estimate of the autocorrelation of the whole population. In this work, however, we employ the autocorrelation function as a means to transform the time series into a different decision space [4].

## 3 Representation Ensembles

An ensemble of classifiers is a set of base classifiers whose individual decisions are combined to classify new examples [15]. Each classifier is allowed to independently observe the example and provide a tentative classification output, *i.e.*, a vote. The ensemble then combines the individual votes into a single class label.

The simplest sensible ensemble of classifiers is the majority ensemble. In a majority ensemble, each of the base classifiers votes on a class label. The most voted class label is the ensemble's output.

Well-crafted ensembles tend to be more precise than their base classifiers. When designing ensembles, it is important that component classifiers be individually accurate and collectively diverse. Otherwise, if the classifiers are inaccurate, then the composition of incorrect decisions will lead to incorrect decisions, while base classifiers too similar one to another cause the ensemble to make decisions similar to those of the base classifiers, thus failing to improve on them.

In time series classification, variations of the $k$-NN have been used to build ensembles of classifiers. Lines and Bagnall [15] employed different distance measures combined into an ensemble of 1-NN classifiers. Previous work by Oates et al. [16] used the SAX representation of time series to compose ensembles where each base classifier was constructed with different parameters. More recently, Lines et al. [4] proposed a "collective of ensembles" that is essentially an "ensemble of ensembles". The base classifiers include complex classification models such as SVM, Rotation Forests, and variations of the $k$-NN classifier. These include using different distance measures and representation ensembles using the autocorrelation function, the power spectrum, and the shapelet representation.

In this work, we explore how different time series representation may be composed into ensembles of classifiers. Additionally, we are also interested in exploring different strategies of combining base ensembles. We start with the majority ensemble. Then, we explore alternative strategies for weighing base classifiers and composing votes. For clarity sake, we group these strategies into weighted and ranking-based ensembles.

### 3.1 Weighted-Based Ensemble

One straightforward extension to the simple majority ensemble consists of assigning weights to each base classifier. When deciding on a new example, the weighted sum of the votes for each class $C_i$ is considered, and the class label which receives the highest sum of votes is the ensemble output.

This strategy will be referred to as weighted ensemble. In actuality, the weighted ensemble is a family of ensembles that differ on the strategy adopted to assign weights to base classifiers. The majority ensemble is itself a weighted ensemble that uses equal weight to all base classifiers.

In this work, we have considered the following weighing strategies.

- *Accuracy as weights*: the weight of each base classifier is its estimated accuracy. In our experiments, the accuracy is estimated by means of 10-fold cross-validation on the training data. This is the only ensemble strategy that uses the same set of weights for every new example it is presented. Variations of this ensemble may be easily achieved by imposing a "cut-point" on the weights of the base classifiers. A "hard cut-point" of $k$ implies assigning a weight of 0 (zero) to all but the $k$ most accurate base classifiers. A "soft cut-point" of $\delta$ implies assigning a weight of 0 (zero) to all base classifiers that are less accurate than the most accurate base classifier by a value of $\delta$;
- *Distance as weights*: the weight of each base classifier is the normalized distance from the new example $\mathbf{x}$ to its nearest neighbor $\mathrm{z}_{nn}$. Formally, if the

distance function used by the base classifier is $f$, then the normalized distance is $f'(\mathbf{x}, \mathrm{z}_{nn}) = \frac{f(\mathbf{x}, \mathrm{z}_{nn})}{f(z_\alpha, z_\beta)}$, where $z_\alpha$ and $z_\beta$ are instances from the training set such that no other pair of training instances are further apart;

- *Posterior probability as weight*: the weight of each base classifier is the posterior probability of the class given the example. Let $C_j$ be the decision of the $j$-th classifier. The posterior probability $P(C_j|\mathbf{x})$ is the probability that the true class of the new example $\mathbf{x}$ is, in fact, $C_j$. The most straightforward approach to estimate the posterior probability is to count the frequency of $C_j$ in the neighborhood of $\mathbf{x}$. This neighborhood is a parameter of the ensemble, and it is usually larger than the neighborhood of the base classifier.

### 3.2 Ranking-Based Ensembles

While the 1-NN produces a single class label for each new instance $\mathbf{x}$, it may be easily extended to rank classes according to the likelihood that a class $C_i$ is the true class of $\mathbf{x}$. This "extended 1-NN" may be used as the base classifier of an ensemble, provided all other base classifiers also produce rankings as outputs. In this work, this approach is referred to as "ranking ensemble". When a new example is presented to a ranking ensemble, each base classifier is used to produce their own ranking. The ensemble then merges these rankings, much like a single vote is produced from a set of votes in the weighted ensemble. The best-ranked class is chosen as the ensemble decision.

Ranking ensembles differ by the strategy used to construct the rankings. In this work, the merge procedure is the same for all ranking ensembles; namely, it is the mean of ranks. If the classification space has $m$ classes and the ensemble is composed of $n$ base classifiers, then each $j$-th base classifier votes on a ranking $R_j = \{r_{j1}, r_{j2}, \ldots, r_{jm}\}$, where $r_{ji}$ is the rank assigned to the $i$-th class label. The final, merged ranking, is given by the partial order of the set $R_f = \{\overline{r_1}, \overline{r_2}, \ldots, \overline{r_m}\}$, where $\overline{r_i} = \frac{\sum_j r_{ji}}{n}$ is the mean of the ranks assigned to the $i$-th by the base classifiers.

Two different label ranking models were used in this work.

- *Posterior probability*: the posterior class probability of an example $\mathbf{x}$, $P(C_j|\mathbf{x})$, is the probability that the example $\mathbf{x}$ in fact belongs to class $C_j$. It may be considered a decent estimate of the classifier's "confidence" that its decision is correct. It may also be used in label ranking to score class labels.

  One method to estimate the posterior class probability of $\mathbf{x}$, proposed by Atiya [3], associates a set of weights, $\{v_1, v_2, \ldots, v_k\}$ such that $v_i \in [0, 1]$ and $\sum v_i = 1$, to the nearest neighbors of $\mathbf{x}$. Each weight $v_i$ reflects how important is the $i$-th neighbor in estimating the posterior class probability. If $v_1 = 1$ and all other $v_i = 0, i \neq 1$, then only the class of the immediately nearest neighbor should be considered in the estimate. If $v_i = \frac{1}{k}$ for all $k$-nearest neighbors, then every neighbor is equally important. A linear optimization model is used to to find weights that are optimal according to a training data set. The probability that an example $\mathbf{x}$ belongs to the class $C_i$ is the sum of the weights associated to the neighbors of $\mathbf{x}$ that belong to this class. This is better explained with an example. Assume that, for $k = 5$ and a given example $\mathbf{x}$, the classes of its nearest neighbors are $C_1$, $C_2$, $C_1$, $C_3$, and $C_2$. The first and the third-nearest neighbors are of class $C_1$. Therefore, the probability that $\mathbf{x}$ belongs to $C_1$ is given by $P(C_1|\mathbf{x}) = v_1 + v_3$. The values of $P(C_j|\mathbf{x})$ are subjected to a partial order, where higher values are better, and a ranking is constructed. The ranks associated with each class label are the base classifier's vote.

- *Simple-ranking based*: the simple-ranking is an instance-ranking strategy used in the anytime classification with the $k$-nearest neighbors classification model [17]. Basically, the idea of the simple-ranking is to rank the training instances according to their relevance to the classification of a new example. The simple-ranking approach to calculating the importance of each training instance is similar to the leave-one-out validation procedure. Initially, it assigns a score of zero to all training instances. Then, at every round, it removes one training instance from the original data set and use the rest of the training data to predict its class. In the original simple-ranking algorithm, the classification model is the 1-nearest neighbor classifier. The nearest neighbor is then deemed "friendly" or "enemy", depending on whether its classes matches or not the class of the held out instance. If the neighbor is "friendly", then its score is increased by 1. If the neighbor is "enemy", then its score is decreased by $\frac{2}{m-1}$, where $m$ is the number of classes of the decision space. The held out instance is put back into the training data set and the algorithm proceeds to the next round. Finally, a ranking is produced from the scores and the ranks are assigned to the training instances.
  Once the training instances have been ranked, the label ranking of a new example $\mathbf{x}$ is similar to the procedure of the posterior probability-based method. Each nearest neighbor was previously assigned a simple-rank $s_1$, $s_2$, …, $s_k$. For each class $C_j$, a score is computed from the mean of the ranks of neighbors that belong to class $C_j$. For instance, assume that $k = 5$ and the classes of the nearest neighbors of $\mathbf{x}$ are $C_1$, $C_2$, $C_1$, $C_3$, and $C_2$. The first and the third-nearest neighbors belong to class $C_1$. Therefore, the score assigned to this class is $S_1 = \frac{s_1+s_3}{2}$. Similarly, the score assigned to $C_2$ is $S_2 = \frac{s_2+s_5}{2}$ and the score assigned to $C_3$ is $S_3 = s_4$. The scores $S_1$, $S_2$, and $S_3$ are subjected to a partial order, where lower values are better, and a ranking is constructed. The ranks associated with each class label are the base classifier's vote.

## 4 Experimental Evaluation

The main goal of this work is to explore different forms of ensemble composition with representation diversity. To achieve that goal, we designed a set of experiments to evaluate all ensemble strategies presented in Sect. 3 using different subsets of base classifiers. Because some ensemble strategies and some transformations have parameters, we also cross-validated the training data.

The base classifiers employed in our experiments were the 1-NN, the 1-NN-DTW, the 1-NN-DTW with Sakoe-Chiba window, and the 1-NN transformed to the representations discussed in Sect. 2. The width of the Sakoe-Chiba window was individually assigned to each data set. Additionally, one base classifier using only the first iteration of the Haar transform was used, alongside with another base classifier using only the approximation coefficients of the first iteration of the Haar transform. This configuration of base classifiers will be referred to as "full set".

For some configurations, we also experimented with limited sets of base classifiers. That is, instead of using the "full set" as base classifiers, we experimented different combinations. These combinations and the difference of accuracy provided by them will be presented and discussed in the next subsection.

The results from these experiments shed light on how the choice of an ensemble configuration and how adding or removing base classifiers affect the classification accuracy. Based on these results, we then devised a set of 6 base classifiers using 4 time series representations, namely 1-NN, 1-NN-DTW, 1-NN-DTW with Sakoe-Chiba window, and 1-NN transformed to Power Spectrum, Haar coefficients, and autocorrelation coefficients. The same ensemble configurations were used. We refer to this set of base classifiers as the "reduced set".

All experiments were conducted in 45 data sets from the UCR time series repository [11]. The UCR repository is arguably the largest source of time series data for classification and clustering, spanning diverse domains of applications such as agronomy, human movements, medicine, and astronomy. One characteristic of the UCR repository is that data sets are shipped with a predefined partitioning of training and test data. As it is commonplace in the scientific community, we keep that partitioning to promote reproducibility.

### 4.1 Discussion of Results

We evaluated hundreds of ensemble configurations on 45 labeled data sets. This yielded over 17 thousand data points, the totality of which may be found in the accompanying website of this paper [9].

Comparing ensembles with different sets of base classifiers, our results showed that fewer base classifiers tend to provide better results than the "full set". In Table 1, four sets of base classifiers are compared against the "full set". Each column stands for a subset of base classifiers: Time domain, DTW (with Sakoe-Chiba window), Power spectrum, Autocorrelation, and Haar. The values are the frequency of victories and ties assigned to the column ensemble against the respective "full set".

Most configurations produced better results with fewer base classifiers. This led us to repeat the experiments for all configurations with the six base classifiers presented in Sect. 4. In Table 2 we present a summary of ensemble configurations that we deemed representative. The totality of our results may be found in [9].

From Table 2, it is possible to notice that no configuration is better than all other configurations. For instance, though the strategy of weighing base classifiers by their cross-validation accuracy is at least as good as the majority ensemble 86.67 % of the time, it is worse than the strategy of weighing by distance 44.44 % of the time.

**Table 1.** Comparison of ensembles with reduced sets of base classifiers against the "full set". Letters denote base classifiers: (T)ime-domain with Euclidean distance and (D)TW, (A)utocorrelation domain, (P)ower spectrum domain, and (H)AAR wavelets

| | TDPA | TDPH | TDAH | TDPAH |
|---|---|---|---|---|
| Majority | 73.68 % | 34.21 % | 47.37 % | 76.32 % |
| Posterior (NN) | 63.16 % | 31.58 % | 26.32 % | 60.53 % |
| Accuracy | 81.58 % | 76.32 % | 78.95 % | 89.47 % |
| Distance | 94.74 % | 94.74 % | 97.37 % | 97.37 % |

**Table 2.** Comparison of ensemble configurations. Each value is the frequency of victories and ties obtained by the column ensemble against the row ensemble. In the columns, P.P. stands for "posterior probability".

| Against | Reference ensemble | | | | | |
|---|---|---|---|---|---|---|
| | Majority | Accuracy | Distance | P.P. (rank) | P.P. (weight) | SimpleRank |
| Majority | — | 86.67 % | 71.11 % | 55.56 % | 77.78 % | 37.78 % |
| Accuracy | 20.00 % | — | 53.33 % | 26.67 % | 48.89 % | 17.78 % |
| Distance | 35.56 % | 55.56 % | — | 35.56 % | 53.33 % | 35.56 % |
| Posterior (rank) | 48.89 % | 77.78 % | 68.89 % | — | 75.56 % | 37.78 % |
| Posterior (weight) | 28.89 % | 60.00 % | 51.11 % | 28.89 % | — | 26.67 % |
| SimpleRank | 62.22 % | 82.22 % | 64.44 % | 64.44 % | 75.56 % | — |

Surprisingly, more complex strategies based on ranking ensembles, such as the simple-ranking and the posterior probability method, did not yield good results. At first, we suspected this might be caused by overfitting on the training data. However, what we actually observed was an overall superior accuracy of those methods on the test data set, suggesting that these ensembles generalize relatively well from the training data. As an example, Fig. 1 (left) compares the accuracy on the test data against the accuracy on the training set for the posterior ranking ensemble – with neighborhood of size 5. Figure 1 (right) presents a similar analysis for the simple rank ensemble – with neighborhood of size 3.

These results seem to make a good case for composing ensembles of 1-NN on different domains of representation. However, the natural question is: how do these ensembles compare with the state-of-the-art? Considering the overwhelming adoption of the 1-NN-DTW, we did compare our ensemble configurations against it. In Fig. 2 we graphically present two such comparisons. In Fig. 2 (left), the accuracy-weighted ensemble – with "soft cut-point" of 0.1 – is compared against 1-NN-DTW. In Fig. 2 (right), the simple rank ensemble – with neighborhood size 3 – is compared against 1-NN-DTW.

There is a sensible reason for choosing these particular ensembles for comparison against 1-NN-DTW. That particular configuration of the accuracy ensemble was able to beat or tie with 1-NN-DTW in more data sets than most other ensembles. Conversely, that particular configuration of the simple rank ensemble

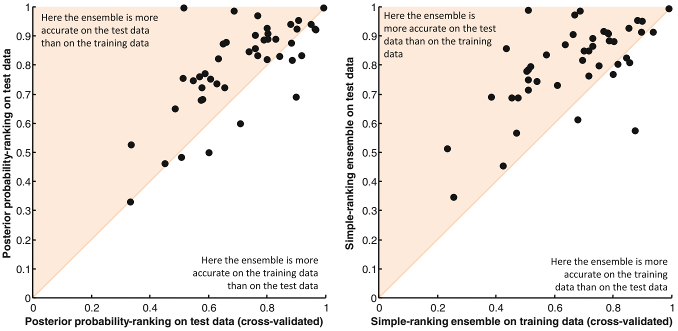

**Fig. 1.** Comparison of ranking ensembles on the test data against training data, suggesting generalization of the classification models.

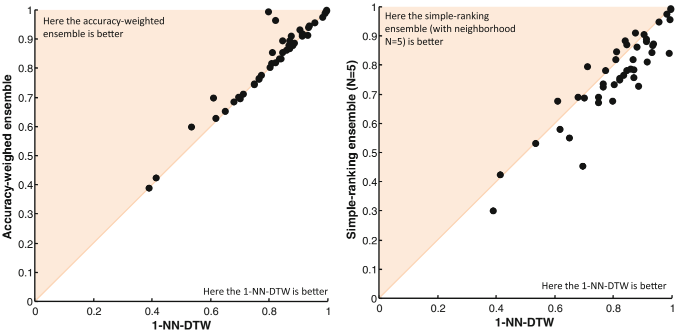

**Fig. 2.** Comparison of two different ensemble strategies against the 1-NN-DTW.

was defeated by the 1-NN-DTW in more data sets than most other ensembles. Figure 2 therefore gives a glimpse of the range of accuracies of our results when compared with the 1-NN-DTW. The accuracy ensemble won against 1-NN-DTW in 28 data sets, tied in 12, and lost in 5.

Finally, we address the issue of reproducibility. Along this section, we presented several results that we considered representative or interesting. The totality of this analysis is based on evaluations performed with the test data, which is usually not available in "real" situations.

We address this question by performing a data-driven selection of ensembles. Every ensemble configuration was evaluated both on the test data and, with cross-validation, on the training data sets. For each data set, the "dynamic ensemble" chooses the ensemble with the highest accuracy on the training data and evaluates it on the test data. When more than one ensemble configuration yields the highest training accuracy, we apply them all on the test data and give as its accuracy the mean of their test accuracies. This equates to finding the expected value of the "dynamic ensemble" accuracy when equally efficient ensemble configurations are randomly selected with uniform distribution.

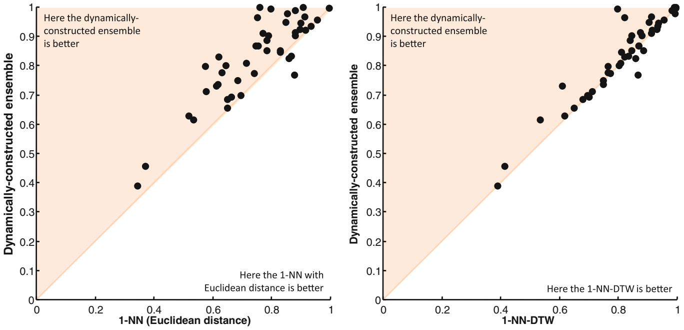

**Fig. 3.** (left) The best ensemble using the "full set" of base classifiers against 1-NN-DTW; (right) accuracy a data-driven selection of ensemble for each data set comparaed with 1-NN-DTW

In Fig. 3 we present the "dynamic ensemble" against the 1-NN (left) and the 1-NN-DTW (right). This particular "dynamic ensemble" was produced using the six base classifiers presented in Sect. 4. This result shows that it is possible to construct ensembles of classifiers that are competitive against the 1-NN-DTW from a purely data-drive approach.

## 5 Conclusion and Future Work

In this paper, we have evaluated the use of different data representations in time series classification and ensemble composition. We employed 5 different time series transformations and 6 ensemble strategies. We performed experiments with over 300 ensemble configurations on 45 data sets. Our extensive experimental analysis makes a strong case for the use of representation diversity in ensemble composition. Some ensemble configurations displayed excellent accuracy performance, being competitive with the 1-NN-DTW. Moreover, because we conducted several experiments with cross-validation on the training data, we have strong evidence that data-driven selection of ensemble configuration is possible, and as shown in Sect. 4.1, capable of yielding good results.

As future work, we intend to analyze more time series representations and ensemble strategies. We suspect that some meta-learning techniques would allow for better selection of base classifiers and ensemble configurations.

## References

1. Agrawal, R., Faloutsos, C., Swami, A.: Efficient similarity search in sequence databases. In: Lomet, David B. (ed.) FODO 1993. LNCS, vol. 730, pp. 69–84. Springer, Heidelberg (1993)
2. Antoniou, A.: Digital Signal Processing. McGraw-Hill, New York (2006)
3. Atiya, A.F.: Estimating the posterior probabilities using the k-nearest neighbor rule. Neural Comput. **17**(3), 731–740 (2005)

4. Bagnall, A., Lines, J., Hills, J., Bostrom, A.: Time-series classification with COTE: the collective of transformation-based ensembles. IEEE Trans. Knowl. Data Eng. **PP(99)**, 1–14 (2015)
5. Bresolin, A.d.A., Neto, A., Alsina, P.: Digit recognition using wavelet and SVM in Brazilian Portuguese. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 1545–1548 (2008)
6. Burrus, C.S., Gopinath, R.A., Guo, H.: Introduction to Wavelets and Wavelet Transforms, vol. 998. Prentice Hall, New Jersey (1998)
7. Chan, K.P., Fu, A.C.: Efficient time series matching by wavelets. In: International Conference on Data Engineering, pp. 126–133 (1999)
8. Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., Keogh, E.: Querying and mining of time series data: experimental comparison of representations and distance measures. VLDB Endowment **1**(2), 1542–1552 (2008)
9. Giusti, R., Silva, D.F., Batista, G.E.A.P.A.: Time series classification with representation ensembles (2015). http://sites.labic.icmc.usp.br/rgiusti/ida15. (URL verified valid as of July 2015)
10. Jolliffe, I.: Principal Component Analysis. Springer, New York (2002)
11. Keogh, E., Xi, X., Wei, L., Ratanamahatana, C.A.: The UCR time series classification/clustering homepage (2006). http://www.cs.ucr.edu/~eamonn/time_series_data/. (URL verified valid as of July 2015)
12. Keogh, E., Ratanamahatana, C.A.: Exact indexing of dynamic time warping. Knowl. Inf. Syst. **7**(3), 358–386 (2005)
13. Lin, J., Keogh, E., Wei, L., Lonardi, S.: Experiencing SAX: a novel symbolic representation of time series. Data Min. Knowl. Discov. **15**(2), 107–144 (2007)
14. Lin, J., Khade, R., Li, Y.: Rotation-invariant similarity in time series using bag-of-patterns representation. J. Intell. Inf. Syst. **39**(2), 287–315 (2012)
15. Lines, J., Bagnall, A.: Time series classification with ensembles of elastic distance measures. Data Min. Knowl. Discovery **29**, 565–592 (2014)
16. Oates, T., Mackenzie, C., Stein, D., Stansbury, L., Dubose, J., Aarabi, B., Hu, P.: Exploiting representational diversity for time series classification. In: International Conference on Machine Learning and Applications, vol. 2, pp. 538–544 (2012)
17. Ueno, K., Xi, X., Keogh, E.J., Lee, D.J.: Anytime classification using the nearest neighbor algorithm with applications to stream mining. In: IEEE International Conference on Data Mining, pp. 623–632 (2006)
18. Wang, X., Mueen, A., Ding, H., Trajcevski, G., Scheuermann, P., Keogh, E.: Experimental comparison of representation methods and distance measures for time series data. Data Min. Knowl. Discovery **26**, 275–309 (2013)
19. Ye, L., Keogh, E.: Time series shapelets: a new primitive for data mining. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 947–956. ACM (2009)