Convolutional Nonlinear Neighbourhood Components Analysis for Time Series Classification

Yi Zheng[1,2], Qi Liu[1], Enhong Chen[1(✉)], J. Leon Zhao[2], Liang He[1], and Guangyi Lv[1]

[1] School of Computer Science and Technology, University of Science and Technology of China, Hefei, China
{xiaoe,hshl05,gylv}@mail.ustc.edu.cn, {qiliuql,cheneh}@ustc.edu.cn

[2] Department of Information Systems, City University of Hong Kong, Hong Kong, China
jlzhao@cityu.edu.hk

Abstract. During last decade, tremendous efforts have been devoted to the research of time series classification. Indeed, many previous works suggested that the simple nearest-neighbor classification is effective and difficult to beat. However, we usually need to determine the distance metric (e.g., Euclidean distance and Dynamic Time Warping) for different domains, and current evidence shows that there is no distance metric that is best for all time series data. Thus, the choice of distance metric has to be done empirically, which is time expensive and not always effective. To automatically determine the distance metric, in this paper, we investigate the distance metric learning and propose a novel Convolutional Nonlinear Neighbourhood Components Analysis model for time series classification. Specifically, our model performs supervised learning to project original time series into a transformed space. When classifying, nearest neighbor classifier is then performed in this transformed space. Finally, comprehensive experimental results demonstrate that our model can improve the classification accuracy to some extent, which indicates that it can learn a good distance metric.

1 Introduction

Among time series data mining tasks, the classification has attracted amount of interest during last decade. Actually, many studies on time series classification methods have been proposed and it is suggested that Nearest Neighbor classifier (especially, 1-NN) is difficult to beat [1,3]. Since the performance of 1-NN algorithm depends critically on the distance metric given for specific tasks, the subsequent question then becomes how to determine the distance metric for so many applications.

A number of different distance metrics have been proposed. Among them, two of the most widely used are Euclidean distance and Dynamic Time Warping

T. Cao et al. (Eds.): PAKDD 2015, Part II, LNAI 9078, pp. 534–546, 2015.
DOI: 10.1007/978-3-319-18032-8_42

(DTW) [1,3,22]. Euclidean distance is simple and efficient, and it could achieve a good performance for certain applications. In contrast, DTW introduces the alignment of two sequences and allows the points of different time stamps to match, which leads to even better performance than Euclidean distance for some scenarios. However, one of the deficiencies of DTW is that it needs more time cost when calculating the distance. Also, even though 1-NN with DTW can achieve best performance in many domains, for some other applications, it performs no better than other distance metrics. In summary, current evidence shows that there is no distance metric that is best for all time series data [3]. Typically, the choice of distance metric has to be determined empirically, which is time expensive and not always effective. Hence, we believe that it's a challenge to choose a suitable distance metric for the specific data set automatically.

Inspired by the learning perspective, we investigate to use distance metric learning to obtain better distance metric and further to improve the classification performance for time series data. Indeed, many distance metric learning methods have been proposed. For instance, [4] provided a linear transformation model named Neighbourhood Components Analysis (NCA) to optimize the performance of k-NN in the learnt low-dimensional space. As [19] noted, the linear transformation has a limitation that "it cannot model higher-order correlations between the original data dimensions". Hence, [19] proposed a nonlinear distance metric learning model named Nonlinear NCA (NNCA). The discovered low-dimensional representations could work better than previous linear NCA. Unfortunately, both Linear NCA (LNCA) and NNCA models cannot capture the intrinsic property of the time series data, i.e., time shift.

To capture the time shift property, in this paper, we consider the merit of Convolutional Neural Network (CNN), e.g., invariance of spatial-temporal, and propose a novel distance metric learning method for time series. Specifically, we follow NNCA model [19] and propose a novel Convolutional Nonlinear Neighbourhood Components Analysis (CNNCA) model, which could not only learn a nonlinear transformation from the data but also naturally capture the time shift of sequences. Based on the learnt distance metric, 1-NN classifier would be used to perform the classification. Moreover, we conduct comprehensive experiments on the data sets from UCR Time Series repository [7]. By comparing to conventional Euclidean distance, DTW and window constraint DTW, the experimental results reveal the classification performance is improved for many data sets, especially for the data sets that have sufficient training samples for each class. On the other hand, we also evaluate the efficiency of each method. It reveals that CNNCA is more efficient for larger data set and long time series. We summarize the contributions of this paper in these parts:

- Though there are several studies that have explored the distance metric learning for time series data [12,15], to the best of our knowledge, we are the first to consider the time shift property when learning distance metric for the time series classification task.
- Along this line, we propose a novel distance metric learning method CNNCA for time series data, which can obtain combined feature representation by

concatenating CNN and Multiple Perceptron (MLP), and then learn a distance metric based on the scheme of stochastic neighbour assignments.
– We conduct comprehensive experiments on amount of public data sets, then compare the performance of CNNCA with other distance metrics, including not only three conventional distance metrics, but also two learnt by LNCA and NNCA. The results prove that CNNCA can improve classification accuracy to some extent, especially for the relatively large scale data sets.

The rest of this paper is organized as follows. Section 2 shows the related studies. Definitions of time series and relevant distance metric learning methods are given in section 3. In section 4, the CNNCA is introduced and comprehensive experiments are presented in section 5. Finally, we conclude the paper and give the future work in section 6.

2 Related Work

We group the related studies into two categories. In the first category, researchers focus on improving the performance of time series classification by choosing distance metrics combined with 1-NN classifier. As [1,16,22] claimed, the Nearest Neighbour (NN) classification algorithm (especially 1-NN) has been empirically proven as the current state-of-the-art [1,16,22]. Then the challenge of 1-NN is how to determine the distance metric for specific data sets. Extensive experiments have been conducted by [3] on amount of time series data sets and many distance metrics have been evaluated, i.e., Manhattan distance, Euclidean distance, L_∞-norm, DISSIM, DTW, LCSS, EDR, Swale, ERP, TQuEST, SpADe [3]. According to the experimental results, they concluded that there is no clear evidence that there exists one similarity measure that is superior to others for most of data sets. Hence, for specific data set, it is challenging to determine a suitable distance metric for better performance.

In the second category, researchers concentrate on the distance metric learning (or manifold learning). Essentially, the aim of distance metric learning is to learn either a linear or nonlinear transformation based on the original data for further tasks (e.g., classification, clustering or visualization) [4,12,15,19]. For instance, [4] proposed a method by optimizing the expected leave-one-out error of a stochastic nearest neighbor classifier in the projection space, which can learn a linear distance metric to be used for data visualization and fast classification. [19] said that the linear transformation cannot capture the higher-order correlations between original data dimensions and proposed a nonlinear NCA model, which stacks multiple neural networks to learn the nonlinear transformation for handwritten digit recognition task. To the best of our knowledge, there are only several existing studies using distance metric learning on time series classification. For instance, [15] considered to learn a variation of Mahalanobis distance and performed the time series classification with 1-NN algorithm. They concluded that such a kind of distance is inferior to DTW in accuracy but it is more efficient. Recently, [12] proposed two novel models to learn a task-specific similarity measure for time series data, however, the transformation is still linear.

In general, existing distance metric learning methods either linear or nonlinear cannot capture the time shift property well, thus the performance of time series classification are suffered. Motivated from the nonlinear distance metric learning and utilizing the merit of CNN, we will propose a convolutional nonlinear NCA model to learn a better distance metric for time series, and further improve the performance of classification.

3 Preliminaries

In this section, we provide preliminaries for our work. Specifically, we first give the definitions of time series and subsequence. Then, two related distance metric learning models are explained.

3.1 Definitions of Time Series and Subsequence

Definition 1 *A time series (denoted as T) is a sequence of data points, measured typically at successive points in time spaced at uniform time intervals. A time series can be denoted as $T = t_1, t_2, ..., t_n$, and n is the length of T.*

Following the previous works [5], we first extract some subsequences from the long time series instead of classifying time series with the whole sequence. Then, we proceed the classification with these subsequences, since the pattern or shape in the subsequences of time series could be a key feature to distinguish different classes of time series. The subsequence is defined as follows.

Definition 2 *Subsequence is a series of consecutive points which are extracted from a long time series T and could be denoted as $s = t_i, t_{i+1}, ..., t_{i+k-1}$, where k is the length of subsequence, and we have $1 \leq i \leq n$, $1 \leq k \leq n$ and $i+k-1 \leq n$.*

Three conventional distance metrics are most widely used: Euclidean distance, DTW and window constraint DTW. Due to space limitations, we skip the details of these distance metrics (which could be found in [3,22]).

3.2 Distance Metric Learning

Many distance metric learning methods have been proposed during last decade [4,19]. In this paper, we concentrate on two preliminary methods on Neighborhood Components Analysis (NCA), i.e., linear and nonlinear NCAs.

Linear Neighbourhood Components Analysis (LNCA). Based on stochastic neighbour assignments in the transformed space, [4] introduced a differentiable cost function for learning neighbour components analysis. Specifically, for each point x_i, it selects another point x_j as its neighbour with the probability p_{ij}, and furthermore, x_i would be classified as the label of point x_j with the same probability. In the softmax scheme, the definition of p_{ij} with Euclidean distance is shown in Equation 1.

$$p_{ij} = \frac{exp(-\|Ax_i - Ax_j\|^2)}{\sum_{k \neq i} exp(-\|Ax_i - Ax_k\|^2)}, \quad p_{ii} = 0, \tag{1}$$

where A is the matrix that needs to be learnt for transforming the input data linearly. Based on such a stochastic neighbour assignments scheme, the probability that point x_i would be classified correctly is computed as follows.

$$p_i = \sum_{j \in C_i} p_{ij}, \tag{2}$$

where C_i represents the set of points that have same class label as point x_i, and c_i denotes the class label of point x_i then we define this set as $C_i = \{j | c_i = c_j\}$. The objective function of LNCA is shown in Equation 3, which is also the expected number of points that is correctly classified.

$$\mathcal{L} = \sum_i \sum_{j \in C_i} p_{ij} = \sum_i p_i. \tag{3}$$

To maximize the objective function, the common method is to use a gradient based optimizer according to the derivative of $\mathcal{L}$. When we denote that $x_{ij} = x_i - x_j$, then the derivative of $\mathcal{L}$ with respect to A is derived in Equation 4.

$$\frac{\partial \mathcal{L}}{\partial A} = -2A \sum_i \left(p_i \sum_k p_{ik} x_{ik} x_{ik}^\top - \sum_{j \in C_i} p_{ij} x_{ij} x_{ij}^\top \right). \tag{4}$$

Nonlinear Neighborhood Components Analysis (NNCA). The limitation of linear transformation is that it cannot capture the higher-order correlations between original data dimensions [19]. Based on LNCA and by introducing a multilayer neural network, [19] proposed a Nonlinear Neighborhood Components Analysis (NNCA) model.

In contrast to Equation 1, for NNCA model, the probability that point x_i selects one of its neighbours x_j and inherits the class label of x_j is defined in Equation 5.

$$p_{ij} = \frac{exp(-\|f(x_i) - f(x_j)\|^2)}{\sum_{k \neq i} exp(-\|f(x_i) - f(x_k)\|^2)}, \quad p_{ii} = 0, \tag{5}$$

where $f(\cdot)$ is the nonlinear transformation learnt by a multilayer neural network, which is different from the linear transformation of LNCA in Equation 1 (i.e., Ax_i). Besides that, the subsequent process of NNCA model is similar to LCNA as shown in Equation 2 and 3, which includes the probability that point x_i belongs to a certain class z and the objective function. The optimization of the objective function is performed with gradient ascent method. Denote $x_{ij} = x_i - x_j$, the derivative of $\mathcal{L}$ with respect to $f(x_i)$ is derived as:

$$\frac{\partial \mathcal{L}}{\partial f(x_i)} = 2\Big(\sum_{l:c^l=c^i} p_{li} d_{li} - \sum_{l \neq i} \Big(\sum_{q:c^l=c^q} p_{lq} \Big) p_{li} d_{li} \Big)$$
$$-2\Big(\sum_{j:c^i=c^j} p_{ij} d_{ij} - \sum_{j:c^i=c^j} p_{ij} \Big(\sum_{z \neq i} p_{iz} d_{iz} \Big) \Big). \tag{6}$$

Through computing gradient and iterating to update the parameters, then we could obtain the nonlinear transformation when the model convergent.

Even though NNCA can learn a nonlinear transformation of the input space, it does not consider the time shift and still cannot capture the intrinsic property of time series. Therefore, for time series classification, both of LNCA and NNCA cannot achieve good performance. We will verify this in the experiments.

4 Convolutional Nonlinear NCA (CNNCA)

In this section, we show the novel distance metric learning model CNNCA, including the architecture and the learning procedure. Meanwhile, we explain how to perform classification with CNNCA at the end of this section.

4.1 Architecture

We follow the scheme of NCA model and extend the nonlinear NCA model for subsequent classification. Specifically, we propose a novel Convolutional Nonlinear Neighborhood Components Analysis (CNNCA) model to learn a better distance metric for time series. By consideration of time shift property of time series, the motivation of introducing CNN into distance metric learning is that convolutional and pooling operations can preserve the spatial and temporal locality, i.e.,

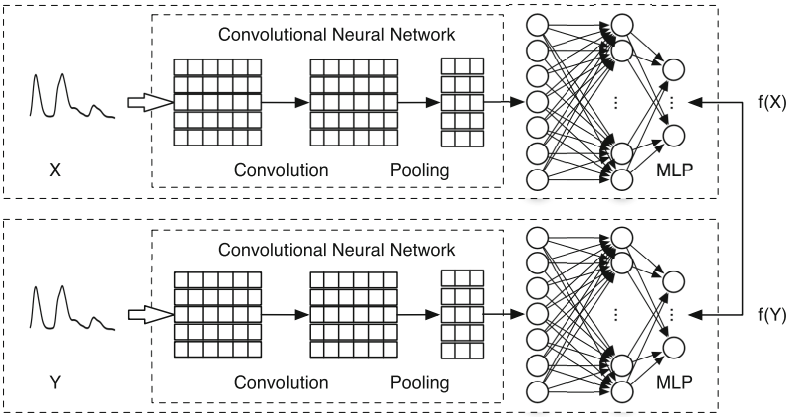

Fig. 1. Architecture of Convolutional Nonlinear Neighborhood Components Analysis. X and Y represent two time series that have identical class label. $f(X)$ and $f(Y)$ denote the nonlinear transformation.

CNN has the advantage of time shift invariance to some extent [9], which may improve the performance of subsequent classification. Furthermore, MLP can combine the feature representations learnt by CNN and perform nonlinear transformation for better classification. Hence, CNNCA extends LNCA by combining CNN and MLP, in other words, the distance d_{ij} between two projected points with respect to x_i and x_j, is calculated in this form: $d_{ij} = \|f(x_i) - f(x_j)\|^2$, where $f(\cdot)$ defines a nonlinear transformation through convolutional neural networks and multilayer perceptron. We illustrate the architecture of CNNCA model in Fig. 1. The probability that point i belongs to class z depends on the relative proximity for all other points that belongs to class z, which is the same as NNCA that was shown in Equation 2. Moreover, similarly, the distribution of distance p_{ij} is formalized and was shown in Equation 5. The objective function of CNNCA is identical to that of linear and nonlinear NCA in Equation 3. Our aim is to maximize this function, from another perspective, $\mathcal{L}$ is the expected number of correctly classified points for the training data.

4.2 Optimization

Based on conventional backpropagation algorithm, to update the parameters iteratively, feedforward computation and backpropagation need to be performed alternatively until the model converges.

Feedforward Pass. The feedforward pass aims to perform the nonlinear transformation from the input time series to the final low-dimensional space. Concretely, we use CNN to learn the features and then feed the output feature maps into a MLP, the purpose of which is to combine of the learnt features and obtain a good distance metric at the final layer. For the traditional CNN, it could consist of multiple stages and each stage contains three cascaded layers [8,9,11,21,24]. We briefly recall the process of these three layers, i.e., filter (convolutional), activation and pooling layers.

$$\mathbf{z}_j^l = \sum_i \mathbf{x}_i^{l-1} * \mathbf{k}_{ij}^l + b_j^l, \quad \mathbf{x}_j^l = \phi(\mathbf{z}_j^l), \quad \mathbf{x}_j^{l+1} = pool(\mathbf{x}_j^l),$$

where $*$ denotes the convolutional operation, $pool(\cdot)$ represents the function used in pooling layer, and $\phi(\cdot)$ represents the activation function. Besides, $\mathbf{x}_i^{l-1}$ and $\mathbf{z}_j^l$ denote the input and output of filter layer and the superscript l represents which layer they involve. $\mathbf{z}_j^l$ and $\mathbf{x}_j^l$ denote the input and output of activation layer, $\mathbf{x}_j^l$ and $\mathbf{x}_j^{l+1}$ denote the input and output of pooling layer. For pooling layer, *average* and *max* pooling strategies are most widely used [13,20]. While the activation function could be considered as $sigmoid(\cdot)$, $tanh(\cdot)$ and ReLU [14,23]. We adopt *max* pooling and ReLU function in this paper due to their good generality and fast convergence [13,14,20,23].

After CNN, we also use a 2-layers fully-connected MLP to combine the learnt features, since the feedforward pass of MLP is standard and the space consumption is limited. More details of MLP can be referred to [10].

Backpropagation Pass. In this paper, we utilize the backpropagation algorithm to train the CNNCA model. Specifically, once the loss function $\mathcal{L}$ is acquired, then based on the chain-rule of derivatives, the error can be propagated back from layer to layer reversely. Here, the derivative of $\mathcal{L}$ with respect to $f(x_i)$ is the same as that of NNCA model, which is already shown in Equation 6. Then the error could be propagated back to the conventional MLP based on $\frac{\partial \mathcal{L}}{\partial f(x_i)}$ layer-wise. After that, the backpropagation of conventional CNN is performed layer by layer reversely [2,24].

4.3 Classification with Distance Metric Learning

We adopt an objective and widely used evaluation method in this work [6], which uses 1-NN classifier on labeled training data to evaluate the classification accuracy of the distance metric used. Each time series has been labeled with correct class in both of training and test sets. 1-NN classifier tries to find the nearest neighbour of input and predict its class label as that of nearest neighbour. For distance metric learning framework, once we have learnt the transformations, according to specific models (LNCA, NNCA, CNNCA), we first transform the test data and training data. Then, 1-NN classifier would be applied on the transformed training and test data for further classification. In this way, the better the distance metric the lower the classification error should be observed.

5 Experiments

In this section, we conduct experiments on a bunch of public time series data sets, and we demonstrate: 1) the classification accuracies/errors with respect to different distance metrics i.e., CNNCA and other existing distance metrics; 2) the comparison of classification performance on the largest 9 data sets with more training samples; 3) the efficiency analysis and discussion.

5.1 Experimental Setup

We conduct comprehensive experiments on 39 diverse time series data sets, provided by UCR Time Series repository [7], which is shown in the first column of Table 1. As claimed by [3], these 39 diverse data sets could make up approximately more than 90% of all publicly available, labeled time series data sets. Besides, the preprocessing was also applied, e.g., standard normalization was formed for each data set and the maximum scale of each time series is 1.0.

Previous studies observed that Euclidean distance (ED), Dynamic Time Warping (DTW) and window constraint DTW (denoted as DTW(r), r represents

the percentage of time series length) are competitive distance metrics for time series classification [3,17,18,22]. Following them, we consider five distance metrics as baseline methods of our CNNCA, and they include ED, DTW, DTW(r), and two of the related distance metric learning models LNCA and NNCA. All the six distance metrics combine 1-NN to perform classification.

5.2 Experimental Results

Overall effectiveness. The experimental results are shown in Table 1. Six rightmost columns of this table exhibit the classification error with respect to these different distance metrics. Bold number accompanied with star symbol of each row indicates the best result for the corresponding data set. For all 39 data sets, our CNNCA model achieves the best results on 13 out of them, which is more than that of ED (3), DTW (9), LNCA (2), NNCA (6) and equals to that of DTW(r). It reveals that our CNNCA model is competitive not only to conventional ED, DTW and DTW(r) but also to LNCA and NNCA models. Especially, it is superior to ED, LNCA and NNCA for most of the data sets.

To illustrate the performance of these different distance metrics more intuitively compared to Table 1, we also provide some scatter plots in Fig. 2 to depict the pair-wise comparisons between CNNCA and the baseline distance metrics. For each of the scatter plots, the vertical (y) and horizontal (x) axes represent CNNCA and the compared distance metrics, which are denoted as "C" and "O"

Table 1. Classification Error of Different Distance Metrics on 1-NN Classifier

	classes	Size	Ratio	DTW	DTW(r)	ED	LNCA	NNCA	CNNCA
wafer	2	1,000	500	0.020	0.005	0.005	0.007	0.005	**0.004**(∗)
StarLight	3	1,000	333.3	0.093	0.095	0.151	0.155	0.091	**0.090**(∗)
Two-Patterns	4	1,000	250	**0**(∗)	0.002	0.090	0.359	0.085	0.048
Chlorine	3	467	155.6	0.352	0.350	0.350	0.471	0.436	**0.250**(∗)
yoga	2	300	150	0.164	0.155	0.170	0.227	0.232	**0.151**(∗)
ECG200	2	100	50	0.230	0.120	0.120	0.130	0.100	**0.070**(∗)
synthetic-control	6	300	50	0.007	0.017	0.120	0.033	0.047	**0.003**(∗)
Thorax1	42	1,800	42.8	0.171	0.185	0.209	0.297	0.295	**0.131**(∗)
Thorax2	42	1,800	42.8	0.120	0.129	0.135	0.166	0.264	**0.101**(∗)
FaceAll	14	560	40	**0.192**(∗)	**0.192**(∗)	0.286	0.410	0.301	0.231
MedicalImages	10	381	38.1	0.263	**0.253**(∗)	0.316	0.379	0.317	0.321
ItalyPowerDemand	2	67	33.5	0.050	0.045	0.045	0.038	**0.031**(∗)	0.044
OSULeaf	6	200	33.3	0.409	**0.384**(∗)	0.483	0.579	0.533	0.463
SwedishLeaf	15	500	33.3	0.210	**0.157**(∗)	0.213	0.320	0.166	0.168
Haptics	5	155	31	0.623	0.588	0.630	0.653	**0.558**(∗)	0.617
Lighting2	2	60	30	**0.131**(∗)	**0.131**(∗)	0.246	0.197	0.279	0.213
FISH	7	175	25	0.167	**0.160**(∗)	0.217	0.520	0.229	0.166
Gun-Point	2	50	25	0.093	0.087	0.087	0.047	0.100	**0.033**(∗)
Trace	4	100	25	**0**(∗)	0.010	0.240	0.350	0.280	0.130
FacesUCR	14	200	14.2	0.095	**0.088**(∗)	0.231	0.363	0.241	0.191
InlineSkate	7	100	14.2	0.616	**0.613**(∗)	0.658	0.769	0.707	0.660
Coffee	2	28	14	0.179	0.179	0.250	**0**(∗)	**0**(∗)	0.036
SonyAIBORobotSurfaceII	2	27	13.5	0.169	**0.141**(∗)	**0.141**(∗)	0.163	0.172	0.142
ECGFiveDays	2	23	11.5	0.232	0.203	0.203	0.273	**0.046**(∗)	0.056
TwoLeadECG	2	23	11.5	0.096	0.132	0.253	0.368	**0.068**(∗)	0.223
WordsSynonyms	25	267	10.6	0.351	**0.252**(∗)	0.382	0.633	0.541	0.401
Adiac	37	390	10.5	0.396	0.391	0.389	0.575	0.783	**0.340**(∗)
CBF	3	30	10	**0.003**(∗)	0.004	0.148	0.141	0.050	0.141
CinC-ECG-torso	4	40	10	0.349	**0.070**(∗)	0.103	0.469	0.251	0.164
Lighting7	7	70	10	**0.274**(∗)	0.288	0.425	0.521	0.425	0.301
MoteStrain	2	20	10	0.165	0.134	**0.121**(∗)	0.141	0.137	0.160
SonyAIBORobotSurface	2	20	10	0.275	0.305	0.305	**0.186**(∗)	0.236	0.195
50words	50	450	9	0.310	**0.242**(∗)	0.369	0.629	0.409	0.338
OliveOil	4	30	7.5	**0.133**(∗)	0.167	**0.133**(∗)	0.300	0.167	**0.133**(∗)
MALLAT	8	55	6.8	0.066	0.086	0.086	0.107	**0.061**(∗)	0.157
Beef	5	30	6	0.500	0.467	0.467	0.333	0.367	**0.267**(∗)
FaceFour	4	24	6	0.170	**0.114**(∗)	0.216	0.261	0.227	**0.114**(∗)
Symbols	6	25	4.1	**0.050**(∗)	0.062	0.100	0.228	0.139	0.122
DiatomSizeReduction	4	16	4	**0.033**(∗)	0.065	0.065	0.176	0.046	0.082

(e.g., ED), respectively. The classification error ratio of two distance metrics under comparison for certain data set is a point that locates at certain coordinates (x, y). Considering that we use classification error but not accuracy to compare the performance, if the classification error ratio (i.e., the point (x, y)) locates above the diagonal line (red line in 2), then it indicates that "O" is more accurate than "C", i.e., $x < y$. Moreover, the further point (x, y) is away from the diagonal line, the greater the margin of classification accuracy being improved. Otherwise, when "C" is more accurate than "O", and point (x, y) would locate below the diagonal line, i.e., $x > y$. All the points that locate at diagonal line indicate that they achieve identical classification error on these data sets, i.e., $x = y$. Besides, more points on one side of the diagonal line indicates that one distance metric is more superior to the other.

Through comparing the classification accuracy between CNNCA and conventional ED, DTW, DTW(r), the results in Fig. 2 reveal that CNNCA is superior to ED on most of the data sets, which demonstrates that such a learnt distance metric can improve the classification accuracy to some extent. However, on total 39 data sets, there is no evidence that either DTW (or DTW(r)) is better or worse than CNNCA, even though window constraints DTW is a little better than DTW. Moreover, by comparing the classification accuracy between CNNCA and LNCA, NNCA, the results show that CNNCA outperforms both of LNCA and NNCA on most of the data sets, which provides the evidence that CNNCA is more effective than previous NCAs just as our expectation.

Effectiveness on Large Data Sets. We provide both the number of classes and the size of training set in Table 1 (in the second and the third columns) and furthermore the average number of training samples per class is calculated as shown in the forth column. As is well known, if the training samples of each class are too few then neural networks cannot capture the features well and may obtain a poor performance. Motivated by this, we filter out the data sets that has few training samples per class, i.e., eliminating the ratio that is no larger than 40 as shown in Table 1. Finally, we obtain 9 data sets (the top 9 rows of Table 1). Likewise, we provide Fig. 3 to depict the comparisons between CNNCA and the baseline distance metrics on these 9 data sets. From both the top rows of Table 1 and the results in Fig. 3, we could observe that our CNNCA model is

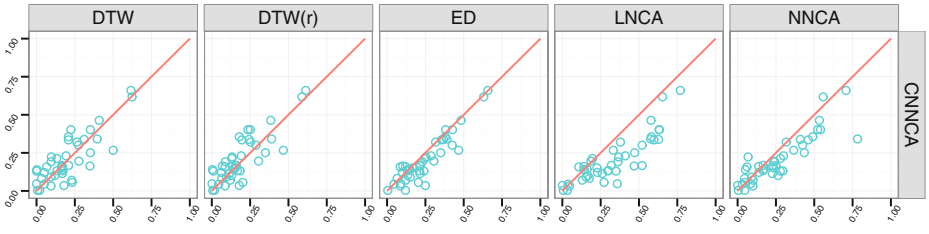

Fig. 2. Comparison of classification accuracy between CNNCA and other baselines

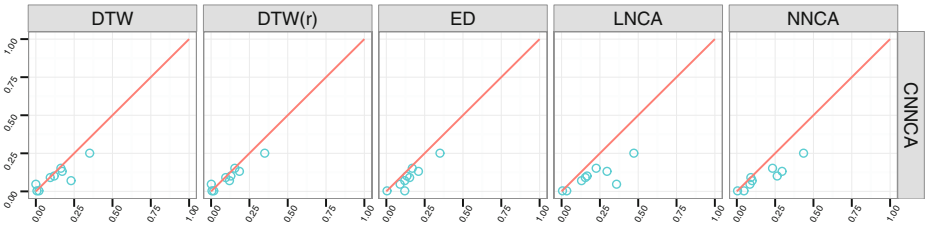

Fig. 3. Comparison of classification accuracy between CNNCA and other baselines. The average number of training samples per class is more than 40.

superior to all other methods on 8 out of these 9 data sets, which demonstrates that if the training samples are sufficient then our CNNCA model could achieve good performance and outperform the baseline methods.

Efficiency Analysis. Supposed that the size of training set is $\mathcal{N}$, and given two time series of length $\mathcal{D}$, then the time complexity of ED, DTW and DTW(r) are $O(\mathcal{N}\mathcal{D})$, $O(\mathcal{N}\mathcal{D}^2)$ and $O(\mathcal{N}\mathcal{D}^2 r)$, respectively, when we apply dynamic programming to compute DTW. Usually, r is no larger than 10% for most applications. Before analyzing the time complexity for LNCA, NNCA and CNNCA, we should note that we only focus on analyzing the online classification of them and skip the offline training process due to the limited space, and it is necessary to define some notations. One hidden layer NNCA is considered for convenience and the number of its hidden neurons sets to $\tilde{n}_h$. For CNNCA, the number of kernels in filer layer and hidden neurons in MLP are denoted as n_k and n_h, respectively. Moreover, the size of kernel and the pooling factor are usually set to 5 and 2. We use d to represent the dimensions of transformed space. To classify each test case, the time complexity of LNCA, NNCA and CNNCA are $O(\mathcal{D}d + \mathcal{N}d)$, $O(\mathcal{D}\tilde{n}_h + \tilde{n}_h d + \mathcal{N}d)$ and $O(5n_k\mathcal{D} + \frac{1}{2}(\mathcal{D}-5+1)n_k n_h + n_h d + \mathcal{N}d)$, respectively, where $O(\mathcal{N}d)$ is the time cost of 1-NN on the transformed data and the remainder is the transformational cost. After reduction, the time cost of CNNCA is $O(n_k\mathcal{D} + \mathcal{D}n_k n_h + n_h d + \mathcal{N}d)$. When $d \ll \mathcal{D}$ and $\mathcal{N}$ is large enough, it is efficient for LNCA, NNCA and CNNCA compared to conventional ED, DTW and DTW(r) if we fix $\tilde{n}_h$, n_h, n_k to constants. We also provide the real time cost of classification on the top 9 data sets in Fig. 4, the data sets in which are ordered by the product of $\mathcal{D}$ and $\mathcal{N}$ increasingly. It reveals that CNNCA is more efficient for larger data set and long time series, i.e., either $\mathcal{N}$ or $\mathcal{D}$ becomes large enough.

Discussion. In summary, the overall experimental results demonstrate that CNNCA is competitive to not only conventional ED, DTW and DTW(r) but also LNCA and NNCA. Especially, after filtering out the relatively small data sets, our CNNCA is superior to all the other distance metrics, which verifies the motivation that CNNCA can capture the intrinsic features and improve the classification performance if the training samples per class are sufficient. By comparison with both of LNCA and NNCA, we also demonstrate that CNNCA

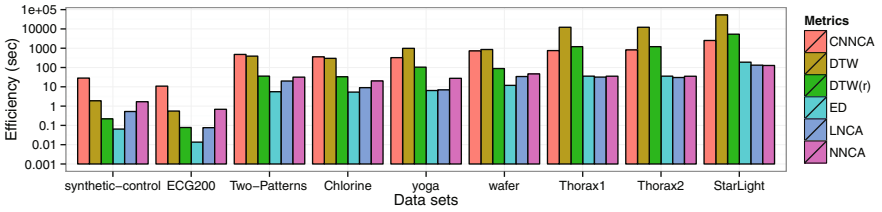

Fig. 4. Time cost of each distance metric combined with 1-NN classifier

is more effective than them to some extent, which is benefited from the capability of capturing time shift property. On the other hand, CNNCA is more efficient than DTW and DTW(r) when the data set grows large enough and time series is long, e.g., for three large data sets, Thorax1, Thorax2 and StarLight in Fig. 4.

6 Conclusions and Future Work

In this paper, we proposed a novel CNNCA model for time series classification. Specifically, we extended the NCA model with CNN and MLP to learn distance metric and then combined 1-NN to classify time series. The benefit of introducing CNN into NCA is to get good feature representations for further classification, and MLP is used to combine these learnt features and obtain nonlinear transformation for better distance metric. For evaluation, we conducted experiments on a bunch of public time series data sets, and observed encouraging results. In particular, CNNCA is superior to current state-of-the-art methods when the training samples are sufficient. We hope this work could lead to many future studies. Actually, we plan to investigate better methods based on CNNCA and further improve the performance (e.g., efficiency) of time series classification.

Acknowledgments. This research was partially supported by grants from the National Science Foundation for Distinguished Young Scholars of China (Grant No. 61325010), the Natural Science Foundation of China (Grant No. 61403358), the Fundamental Research Funds for the Central Universities of China (Grant No. WK0110000042) and the Anhui Provincial Natural Science Foundation (Grant No. 1408085QF110). Qi Liu gratefully acknowledges the support of the Youth Innovation Promotion Association, CAS.

References

1. Batista, G., Wang, X., Keogh, E.J.: A complexity-invariant distance measure for time series. In: SIAM Conf. Data Min. (2011)
2. Bouvrie, J.: Notes on convolutional neural networks (2006)
3. Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., Keogh, E.: Querying and mining of time series data: experimental comparison of representations and distance measures. Proc. VLDB Endow. **1**(2), 1542–1552 (2008)

4. Goldberger, J., Roweis, S., Hinton, G., Salakhutdinov, R.: Neighbourhood components analysis. In: NIPS, pp. 513–520 (2005)
5. Hu, B., Chen, Y., Keogh, E.: Time series classification under more realistic assumptions. In: SIAM Int. Conf. Data Min., p. 578 (2013)
6. Keogh, E., Kasetty, S.: On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration. DMKD **7**(4), 349–371 (2003)
7. Keogh, E., Zhu, Q., Hu, B., Hao, Y., Xi, X., Wei, L., Ratanamahatana, C.A.: The UCR Time Series Classification/Clustering (2011). http://www.cs.ucr.edu/~eamonn/time_series_data/
8. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. In: NIPS, vol. 25, pp. 1106–1114 (2012)
9. LeCun, Y., Bengio, Y.: Convolutional networks for images, speech, and time series. Handb. Brain Theory Neural Networks **3361** (1995)
10. LeCun, Y., Bottou, L., Orr, G.B., Müller, K.-R.: Efficient backprop. In: Orr, G.B., Müller, K.-R. (eds.) NIPS-WS 1996. LNCS, vol. 1524, pp. 9–50. Springer, Heidelberg (1998)
11. LeCun, Y., Kavukcuoglu, K., Farabet, C.: Convolutional networks and applications in vision. In: IEEE ISCS, pp. 253–256, May 2010
12. Lu, Y., Zhao, W.X., Yan, H., Li, X.: A metric learning based approach to evaluate task-specific time series similarity. In: Wang, J., Xiong, H., Ishikawa, Y., Xu, J., Zhou, J. (eds.) WAIM 2013. LNCS, vol. 7923, pp. 314–325. Springer, Heidelberg (2013)
13. Nagi, J., Ducatelle, F., et al.: Max-pooling convolutional neural networks for vision-based hand gesture recognition. In: IEEE ICSIPA, pp. 342–347 (2011)
14. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: ICML, vol. (3), pp. 807–814. Omnipress Madison, WI (2010)
15. Prekopcsák, Z., Lemire, D.: Time series classification by class-specific Mahalanobis distance measures. ADAC **6**(3), 185–200 (2012)
16. Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., Zakaria, J., Keogh, E.: Searching and mining trillions of time series subsequences under dynamic time warping. In: ACM SIGKDD, p. 262 (2012)
17. Ratanamahatana, C., Keogh, E.: Making time-series classification more accurate using learned constraints. In: SIAM Int. Conf. Data Min., p. 11 (2004)
18. Ratanamahatana, C.A.: Three Myths about Dynamic Time Warping Data Mining. In: SIAM Int. Conf. Data Min., pp. 506–510 (2005)
19. Salakhutdinov, R., Hinton, G.E.: Learning a nonlinear embedding by preserving class neighbourhood structure. In: ICAIS, pp. 412–419 (2007)
20. Scherer, D., Müller, A., Behnke, S.: Evaluation of pooling operations in convolutional architectures for object recognition. In: Diamantaras, K., Duch, W., Iliadis, L.S. (eds.) ICANN 2010, Part III. LNCS, vol. 6354, pp. 92–101. Springer, Heidelberg (2010)
21. Sutskever, I., Martens, J., Dahl, G., Hinton, G.: On the importance of initialization and momentum in deep learning. In: ICML, p. 28 (2013)
22. Xi, X., Keogh, E.J., Shelton, C.R., Wei, L., Ratanamahatana, C.A.: Fast time series classification using numerosity reduction. In: ICML, pp. 1033–1040 (2006)
23. Zeiler, M.D., Ranzato, M., Monga, R., et al.: On rectified linear units for speech processing. In: IEEE ICASSP, pp. 3517–3521 (2013)
24. Zheng, Y., Liu, Q., Chen, E., Ge, Y., Zhao, J.L.: Time series classification using multi-channels deep convolutional neural networks. In: Li, F., Li, G., Hwang, S., Yao, B., Zhang, Z. (eds.) WAIM 2014. LNCS, vol. 8485, pp. 298–310. Springer, Heidelberg (2014)