2010 International Conference on Pattern Recognition

A study on combining sets of differently measured dissimilarities

Alessandro Ibba, Robert P.W. Duin, Wan-Jui Lee
Pattern Recognition Laboratory,
Delft University of Technology
Mekelweg 4, 2628 CD Delft, The Netherlands
{*a.ibba, r.p.w.duin, w.j.lee*}*@tudelft.nl*

***Abstract*—The ways distances are computed or measured enable us to have different representations of the same objects. In this paper we want to discuss possible ways of merging different sources of information given by differently measured dissimilarity representations. We compare here a simple averaging scheme [1] with dissimilarity forward selection and other techniques based on the learning of weights of linear and quadratic forms.**
Our general conclusion is that, although the more advanced forms of combination cannot always lead to better classification accuracies, combining given distance matrices prior to training is always worthwhile. We can thereby suggest which combination schemes are preferable with respect to the problem data.

I. Introduction

One of possible representations of data that differs from a feature based description is based on pair-wise comparisons of objects namely distances or dissimilarities. In many cases, distances are obtained directly from raw or pre-processed measurements. Dissimilarities may be chosen when feature representations cannot be helpful in discriminating different classes of objects, in case the experts are not able to define proper features, or if the data lies in high-dimensional spaces (too many features). But also the intrinsic nature of the problem at hand is quite relevant: for instance measures of curves and shapes are good examples of cases in which a dissimilarity representation might be more suitable than classic feature representations in recognition tasks. Although dissimilarity representations can already be seen as a form of classifier combination namely a combination of nearest neighbour (NN) classifiers, in this paper we want to focus our study on possible (feasible) techniques designed in order to gain from the combination of different dissimilarities.
Combining dissimilarity representations (and kernels) has already received some attention in the literature as researchers realized that different dissimilarity measures may emphasize different types of information of objects and classes to be distinguished. [2] and [3] studied combination of kernels for use by support vector machines. [4] and [5] studied the optimization of distance measures in feature space. A combination of differently measured (or computed) dissimilarities can occur at different stages of a pattern recognition system. For example, using the outputs of classifiers built on each dissimilarity separately, but also by combining the various dissimilarities directly. In this paper however we focus on given dissimilarity matrices, as they may arise in practical applications, and study combinations of them judged by the performance of the linear SVM [6] in dissimilarity space.
In a previous study on this topic [1] we compared different ways of combining dissimilarities obtained from different measurements of the same underlying data. The method that did show the best classification accuracy (with respect to the linear normal density based classifier) was based on the sum of normalized matrices. This is the equivalent of a weighted sum where the weights are the normalization factors. The experimental results of [1] triggered the following questions:

- Is averaging dissimilarity matrices always helpful, and if not, is it possible to define conditions (on the measured data, on the distance metrics involved, on the combining weights) to be fulfilled in order to increase the accuracy of our designed classification system?
- Is it possible to define a general optimization procedure in order to select sets of weights that maximize our performance measure?

This work sets out to address these issues.

II. Combining dissimilarities

In a weighted sum of different dissimilarities as:

$$D_{sum} = \Sigma_{i=1}^{K} \omega_i D^{(i)} \tag{1}$$

where K is the total number of available matrices $D^{(i)}$ and ω_i the related weights. In the simplest approach these weights are the inverse of the maximum distance of the corresponding matrix. This scaling procedure has been applied to avoid that the combining method used might be biased by representations with larger distances. This simple averaging scheme (NS) has been compared with other methods used to determine the weights of a linear combination of dissimilarity matrices as eq. (1).

A. Optimization procedures

In this section we will give an overview of the techniques used to find the weights of a combination of distance matrices.

Forward selection (FS)
The dissimilarity forward selection approach can be seen as a greedy combinatorial optimization scheme that gives binary

1051-4651/10 $26.00 © 2010 IEEE
DOI 10.1109/ICPR.2010.820

weights as output. All the given matrices are normalized beforehand, dividing each one by the mean of its distances. The first matrix is selected with respect to the leave-one-out NN error computed on the training set (the entire square matrix), the following matrix is the one that summed to the first minimizes the criterion (NN error). The procedure stops when the criterion on the obtained summed matrix starts rising.

Fisher

This procedure makes use of a kind of Fisher criterion in the dissimilarity space that resembles a method that is often used in kernel combination: the kernel alignment [7]. The objective function to be minimized is given by the following expression:

$$F(\omega_1, \omega_2, ..., \omega_K) = \log\left(\frac{\Sigma_{(x_i,x_j)\in\mathcal{S}}\Sigma_k \omega_k d_k^2(x_i,x_j)}{\Sigma_{(x_i,x_j)\in\mathcal{D}}\Sigma_k \omega_k d_k^2(x_i,x_j)}\right) \quad (2)$$

where $\mathcal{S} = \{(x_i, x_j)|c_i = c_j\}$ and $\mathcal{D} = \{(x_i, x_j)|c_i \neq c_j\}$, c_i is the class to which the object x_i belongs, and K is total number of available matrices and therefore weights to be found. From equation (2) it is possible to see how this criterion resembles the Fisher criterion in a dissimilarity space where the objective function we want to minimize is the log of the ratio between the sum of distances "within" class and the sum "between" class. This method emphasizes the compactness of within class distributions and therefore tends to suffer from multimodal data distributions.

MCML and NCA

In the optimization procedures MCML (Maximally Collapsing Metric Learning)[8] and NCA (Neighbourhood Component Analysis) [9] the elements of the matrix of a Mahalanobis distance between the given ones are determined. The approaches used in this work are instead based on the computation of the weights of a linear combination of squared distances, therefore these are the diagonal versions of the mentioned methods. This variation leads to a much lighter computational load and it has also been proven to provide sufficiently good results [10].

Both methods make use of a conditional distribution such that the probability of selecting an object x_j as a neighbour of the given x_i (with c_i turning to be c_j) is $p(j|i)$. This distribution $p(j|i)$ is computed as the following function of the weighted sum of squared distances:

$$p(j|i) = \frac{\exp\left(-\Sigma_k \omega_k d_k^2(x_i,x_j)\right)}{\Sigma_{t\neq i}\exp\left(-\Sigma_k \omega_k d_k^2(x_i,x_t)\right)}, p(i|i) = 0 \quad (3)$$

Since $p_0(j|i) = 1$ if $(x_i, x_j) \in \mathcal{S}$ and $p_0(j|i) = 0$ if $(x_i, x_j) \in \mathcal{D}$ represents the ideal distribution, the MCML algorithm minimizes the KullbackLeibler divergence [11] between these two distributions ($p(j|i)$ and $p_0(j|i)$) given the semi-positive definiteness of weight matrix (in our setting: weights larger or equal to zero).

$$F(\omega_1, \omega_2, ..., \omega_K) = \Sigma_i KL[p_0(j|i)|p(j|i)] \quad (4)$$

NCA is based on the maximization of the following function:

$$F(\omega_1, \omega_2, ..., \omega_K) = \Sigma_i \log\left(p_i\right) \quad (5)$$

where $p_i = \Sigma_{j\in c_i} p(j|i)$. This method optimizes a continuous version of leave one out kNN error (on the training set), and as MCML is non parametric. But it is not convex as MCML and therefore there is no guarantee that a gradient method (like the conjugate gradient) will converge to a global solution. In order to solve the last three optimization problems (Fisher, MCML and NCA) the conjugate gradient method [12] has been employed.

III. Data and experiments

We have conducted our experiments using the following four datasets:

Chicken pieces silhouettes dataset (446 objects belonging to five classes: 76, 96, 96, 61, 117) [13] (chicken_pieces_44); reduced sets of 11 distance matrices (chicken_pieces); **Biological data** (Bio_data_lkc) set of 5 matrices each constituted of 2400 objects belonging to two classes [14]; **Flowcytometry** 833 (three classes of: 335, 131, 146) histograms described by 252 features, measured with 4 tubes (set of 4 matrices) [15] ; **M-feat**: this dataset consists of features of handwritten digits ('0'–'9', 200 per digit). Six different feature sets are extracted [16], therefore 6 euclidean distances have been computed.

We have applied five different combination techniques (the four mentioned before: FS, NCA, MCML, Fisher and the simple NS approach) compared to the best performing individual ones (BIO). The performance measure used in our experiments has been the classification error of a linear support vector machine [6] in the obtained dissimilarity space. For each one of the four datasets used we have applied a two fold crossvalidation repeated 40 times, in each run of this process we have splitted our data in a training and a test set, the weights have been determined using the optimization procedures (and the binary weights of the Forward selection approach) on the training set. It is important to underline that in the case of the optimization procedures (NCA, MCML and Fisher) the weights have been internally (in the routines) normalized with the Froebenious norm. The mentioned partitioning of the datasets has been carried out consistently for all the used methods, this means that in each run of our procedure the same data has been used as train set and the remaining for testing for each of the six settings. It is very important to underline that the best individual ones have not been selected on the basis of the test set error but on a 40 times 2-fold crossvalidation employed on the training set used also for the other described settings. This gives a less optimistic but definititely more realistic error estimation with respect to the individual matrices.

IV. Results

Our experimental results are provided in table I. They show the classification errors using a linear support vector

Table I
CLASSIFICATION ERRORS (STANDARD DEVIATIONS) FOR THE FOUR DATASETS USING SIX METHODS, THE FIRST ONE BIO (BEST INDIVIDUAL ONES) IS ONLY MEANT TO SHOW THE PROPERTY OF THE ANALYZED DATASETS.

	Combining methods					
Datasets	BIO	FS	NS	NCA	MCML	Fisher
m_feat	3.6 (0.6)	2.3 (0.4)	2.1 (0.5)	**1.9 (0.5)**	**1.9 (0.5)**	3.2 (0.6)
flow_cyto	31.3 (2.2)	13.9 (2.4)	12.3 (1.5)	**11.8 (1.4)**	12.0 (1.4)	16.7 (2.1)
chicken_pieces	8.1 (2.1)	5.5 (2.2)	**5.3 (1.8)**	5.5 (1.8)	5.5 (1.8)	5.8 (2.2)
chicken_pieces_44	8.3 (2.1)	**5.7 (1.9)**	5.8 (2.1)	**5.7 (2.1)**	5.8 (2.2)	7.1 (2.8)
Bio_data_lkc	7.9 (1.7)	7.2 (1.4)	**5.9 (1.1)**	7.0 (1.2)	7.0 (1.1)	6.8 (1.3)

machine (libsvm [6] with default parameters), the given values are the means (and standard deviations) of the classification errors computed as described previously making use of six different procedures. These results show that for all the studied datasets the five methods that involve combinations of the given matrices outperform the BIO error for the best dissimilarity matrix (with respect to the test set).

The binary weights computed by the forward selection lead to classification accuracies very close to the best ones. In the case of the chicken pieces (table I) and in particular for the full collection of 44 chicken pieces matrices [13] this procedure scores even not particularly different from the best one. For the cases of the mfeat and the flowcytometer datasets the NCA and MCML optimization methods are outperforming the NS while this does not happen for the chicken pieces and bio datasets. For these two cases the Fisher method scores equivalent to NCA and MCML. This suggests that the data distributions suffer less from multimodality. For the first two datasets the accuracy of the Fisher technique appears to be much worse than for the other methods. These results might therefore suggest that for multimodal data distributions the NS approach can be a better (and faster) choice than more sophisticated (and computationally expensive) optimization tasks.

In order to test further the performances of the studied optimization techniques we have added a magnified (with a factor of 200) random distance matrix to the previous ones for each dataset and run our experiments with the same settings as before (due to the space constraint the results table has not been provided). The NS performances are (as expected) in this case heavily deteriorated. It is also clear that the NCA and MCML techniques are always better than the other approaches (with the sole exception of BIO); at the same time we can see that the Fisher method is always the worse. In this noisy setting the simple Forward selection based on the leave one out NN error is leading to results characterized by a very high variance.

V. DISCUSSION AND CONCLUSION

Previous works in the field of combining dissimilarity representations ([1], [17], [18]) suggest that a simple averaging of the matrices can lead to classification performances that outperform the results of the individual ones. This was reported with respect to linear and quadratic classifiers (in some cases also regularized) on dissimilarity representations obtained with different prototype selection methods.

In this paper we presented a further analysis, considering weighted averages of dissimilarity matrices. A SVM was used so that regularization and dimension reduction effects could be avoided. It was found that the original conclusions are still valid: averaging of different dissimilarity representations of the same objects may show considerable improvements of the classification performances. Optimizing the weights may improve the results further. A fast and simple procedure to select the most significant dissimilarity matrices hardly ever outperforms averaging all matrices.

The main aim in this work was to compare the simple procedure of averaging matrices with other more sophisticated techniques based on the learning of weights in linear and quadratic forms using optimization algorithms. We have seen that a normalized sum of given matrices can be outperformed by optimization techniques like NCA and MCML. From the experimental results it appears that this might in particular hold for multi-modal data distributions, or at least that lead to worse classification accuracy with respect to the Fisher approach.

The learning of kernel [19] or dissimilarity weights (or metric learning [10], [8], [5], [4]) have been widely studied, but with this paper we wanted to focus in particular on combining different sources of information. These are namely dissimilarity matrices mainly originating from different measurements of the problem data. We have shown that combining before the training stage generally helps and that using techniques previously used for metric learning [10] these linear combinations can lead to even better results. In the future we will further investigate the influence of data with a multi-modal distribution, and in particular procedures to determine a priori, on the basis of the given distances, which might be the most suitable way to combine.

ACKNOWLEDGMENT

We acknowledge financial support from the FET programme within the EU FP7, under the SIMBAD project (contract 213250).

REFERENCES

[1] A. Ibba and R. Duin, "A Multiscale Approach in Combining Classifiers in Dissimilarity Representations," in *Proc. 15th ASCI*, 2009.

[2] I. M. de Diego, J. M. Moguerza, and A. Muñoz, "Combining kernel information for support vector classification," in *Proc. Multiple classifier systems workshop*, 2004, pp. 102–111.

[3] W. J. Lee, S. Verzakov, and R. P. W. Duin, "Kernel Combination Versus Classifier Combination," in *Proc. Multiple classifier systems workshop*, 2007, pp. 22–31.

[4] K. Q. Weinberger and L. K. Saul, "Distance Metric Learning for Large Margin Nearest Neighbor Classification," in *JMLR*, 2009, pp. 207–244.

[5] L. Yang, R. Jin, R. Sukthankar, and Y. Liu, "An efficient algorithm for local distance metric learning," in *AAAI*. AAAI Press, 2006.

[6] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[7] N. Shawe-Taylor and A. Kandola, "On kernel target alignment," *NIPS*, p. 367, 2002.

[8] A. Globerson and S. Roweis, "Metric learning by collapsing classes," *NIPS*, vol. 18, p. 451, 2006.

[9] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighbourhood components analysis," *NIPS*, vol. 17, pp. 513–520, 2005.

[10] A. Woznica, A. Kalousis, and M. Hilario, "Learning to combine distances for complex representations," in *Proc. 24th ICML*. ACM, 2007, p. 1038.

[11] S. Kullback and R. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[12] M. Avriel, *Nonlinear programming: analysis and methods*. Dover Pubns, 2003.

[13] H. Bunke and U. Buhler, "Applications of approximate string matching to 2D shape recognition," *Pattern recognition*, vol. 26, no. 12, pp. 1797–1812, 1993.

[14] M. Hulsman, M. Reinders, and D. de Ridder, "Evolutionary Optimization of Kernel Weights Improves Protein Complex Comembership Prediction," *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 6, no. 3, pp. 427–437, 2009.

[15] A. Ibba, "A dissimilarity based rescaling invariant technique to solve histogram classification problems," *MSc. Thesis, Universitá degli studi di Cagliari*, 2007.

[16] P. M. Murphy and D. W. Aha, "UCI repository of machine learning databases," 1992, department of Information and Computer Science. University of California at Irvine.

[17] E. Pekalska and R. Duin, "On combining dissimilarity representations," in *Proc. Multiple classifier systems workshop*, 2001, pp. 359–368.

[18] E. Pekalska, M. Skurichina, and R. Duin, "Combining dissimilarity-based one-class classifiers," in *Proc. 5th Multiple classifier systems workshop*, 2004, p. 122.

[19] G. Lanckriet, N. Cristianini, P. Bartlett, L. Ghaoui, and M. Jordan, "Learning the kernel matrix with semidefinite programming," *JMLR*, vol. 5, pp. 27–72, 2004.