Merging multiple sources of information using dissimilarity based approaches

Alessandro Ibba*, Robert P.W. Duin, Wan-Jui Lee

Faculty of Electrical Engineering, Mathematics and Computer Science, Mekelweg 4, 2628 CD Delft, TU Delft, The Netherlands

Abstract. The ways distances are computed (the metric used) or measured (by mean of different sources) enable us to have different representations of the same objects. Therefore we need either to make a selection or to combine them. A combination of differently measured (or computed) dissimilarities can occur at different stages of a pattern recognition system, e.g. using the outputs of classifiers built on each of them separately but also by combining the various dissimilarity directly. The key point of classifier combination lies either in a proper averaging over different experts/sources or in an integration of different and hopefully complementary approaches. In this paper we want to focus on possible ways of merging different sources of informations given by different dissimilarity representations. The combination step is employed in a selection or weighted average of the data used for our classification task. A simple averaging of these matrices is shown ([1], [2], [3]) to allow for classification accuracies that outperform the single ones. We compare this approach here with a dissimilarity forward selection and other techniques based on the learning of weights of linear and quadratic forms. The more advanced forms of combination of such multiple representations can lead to a better representation of the underlying data and therefore better classification accuracies but this does not hold always. Our general conclusion is that combining given distance matrices before any training stage is apparently always worthwhile.

1 Introduction

One of possible representations of data that differ 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, or in case the experts are not able to define proper features, or 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

* Corresponding Author. Tel: +31 (0)15 27 88433 - Fax: +31 (0)15 27 81843 - e-mails: a.ibba@tudelft.nl, r.p.w.duin@tudelft.nl, w.j.lee@tudelft.nl

in which a dissimilarity representation might perform better in the recognition tasks.

Dissimilarity representations allow the use of huge data, after a proper dimensionality reduction obtained via prototype selection techniques. Another advantage is the possibility to use a larger variety of different classifiers rather than only SVM ones as the constraint of fulfilling the Mercer's theorem is not a demand in this approach, it thereby allows the use of non-euclidean and even non-metric distances.

Combining dissimilarity representations (and kernels) has already received some attention in the literature as researchers realized that different dissimilarity measure may emphasize different types of information of objects and classes to be distinguished. [4], [5] and [6] studied combination of kernels for use by support vector machines. [7] and [8] studied to optimization of distance measures in feature space. In this paper however we focus on given dissimilarity matrices, as they may arise in practical applications, and study just the (weighted) average of them judged by the performance of the linear SVM in dissimilarity space [9]

Although dissimilarity representations can already be seen as a form of classifier combination namely a combination of NN classifiers, in this paper we want to focus our study on possible (feasible) tecniques designed in order to gain from the combination of different dissimilarities. In a previous stuy on this topic [1] we have 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 used weights are the normalization factors. The experimental results of the mentioned papers triggered some questions that are listed below:

- 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.
  These issues will be fully discussed in the following sessions, but our main focus in this work has been on the last point which is related to solving convex and non-convex optimization problems with the aim of finding set of weights able to outperform our previously attained results.

2 Combining dissimilarities

A weighted sum of different dissimilarity representations.

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

The weights ω_i are computed in such a way that the maximum distances are equal (to 1) over all the matrices. 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 has been compared with other methods used to determine the weights of a linear combination of dissimilarity matrices as in the previous expression.

2.1 Forward selection

The dissimilarity forward selection approach gives binary weights as output. The first matrix is selected with respect to the leave-one-out nearest neighbour error computed on the training set (the entire square matrix), the following is the one that summed to the first minimize the criterion. The procedure goes on until the *nn* error computed on the obtained summed matrix start rising.

2.2 Optimization

In this section we will give a brief overview of the techniques used in order to find the weights of our linear combination of distance matrices. In order to solve the folowing optimization problems the conjugate gradient method [10] has been employed.

2.3 Fisher

This procedure makes use of a kind of Fisher criterion in the dissimilarity space that resemble a method that is often used in kernel combination: the kernel alignment [11]. The objective function to be minimized is given by the following expression:

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

where $\mathcal{S} = \{(x_i, x_j)|c_i = c_j\}$ and $\mathcal{D} = \{(x_i, x_j)|c_i \neq c_j\}$ 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.

2.4 MCML and NCA

In the optimization procedures MCML [12] and NCA [13] the elements of the matrix of a Mahalanobis distance between the orginal 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 the diagonal versions of the mentioned methods. This variation leads to a much lighter computational

load and it has also been proven to be provide sufficiently good results [14]. 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 is $p(j|i)$. This distribution $p(j|i)$ is computed as the following function of the weighted sum of squared distances.

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

Since a distribution $p_0(j|i) = 1$ 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 one, the MCML algorithm minimizes the KullbackLeibler divergence [15] 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).

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

The Nearest Component Analysis method is based on the maximization of the following function:

$$F(\omega_1, \omega_2, ..., \omega_K) = \Sigma_i log(p_i) \tag{5}$$

where $p_i = \Sigma_{j \in c_i} p(j|i)$. This method optimizes a continuous version of leave one out kNN algorithm, 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.

3 Data description

In this section we will give a brief description of the data used for our experiments:

- Chicken pieces silhouettes dataset [16]
- Biological data [17]
- Flowcytometry
- M-feat

3.1 Chicken pieces silhouettes dataset

subsectionChicken Pieces Silhouttes Database The chicken pieces dataset contains several silhouettes of chicken pieces. The contour line of each silhouette is extracted using an edge detector. Then the resulting contour line is approximated by a sequence of normalized vectors of constant length. A string consisting of the angles between consecutive vectors is constructed from this vector sequence , which leads to a rotation-invariant cyclic string of relative angles representing the original chicken piece silhouette. The distances between these strings have been computed using an efficient cyclic string edit distance algorithm to make them rotation-invariant edit distances.

This dataset consists of 446 images of chicken pieces. Each piece belongs to one of five categories, which represent specific parts of the chicken: wing (117 samples), back (76), drumstick (96), thigh and back (61), and breast (96). Each one of the given image is in binary format containing the silhouette of a particular piece. Pieces were placed in a natural way without considering orientation. The applied normalization value n indicates that the contour has been normalized to segments of n-pixels length. These segments could have been chosen as symbols for the strings. But due to the following two constraints:

- the images have to be rotation invariant,
- there should be a mirror symmetry

better string representation has been used.
Therefore, the sequence of angles between the segments were chosen as the string representation. Additionally, the approximate algorithm of Bunke and Buhler [16], which handles rotation invariance and axis symmetry, was applied. Cost Functions: The cost functions are defined as the angle difference in case of substitution and as a constant k in case of inserting or deleting a symbol. In the following equation, α and β are arbitrary angles, and ε stands for the empty symbol. $c_k(\alpha \rightarrow \beta) = |\alpha\text{-}\beta|$ (angle difference)
$c_k(\varepsilon \rightarrow \alpha) = k$
$c_k(\alpha \rightarrow \varepsilon) = k$

3.2 Data description of protein pairs kernels and dissimilarity matrices

The objects in this dataset are protein pairs, these are obtained converting features on individual proteins into features on pairs of proteins. Using Pearson correlation for the mRNA expression vectors. Protein sequence kernels are converted in two different ways: using a protein similarity kernel (a linear kernel on sequence kernels), and using a pairwise kernel on sequence kernels. In total, we have P = 49 kernels. The class labels needed to create a train and test-set were extracted from the MIPS yeast complex database. Category 550, which covers complexes determined by high-throughput experiments, was excluded because these high-througput experi- ments are used as features.

In this dataset there are two types of similarity measures:

- LKC (Linear Kernel Combination)

  $$k_{LKC}(x_i, x_j) = \Sigma_{p=1}^{P} \omega_p k_p(x_i, x_j)$$

  where P is the number of kernels combined. As adding kernel functions k_p amounts to concatenating their individual kernel spaces, the inuence of each individual kernel space can be changed (scaled) by its corresponding kernel weight ω_p . For this combination method, the RBF kernel has been used to represent the feature vectors as well as the pairwise kernels.
  Here, feature vectors are represented by linear kernel functions.

Kernel weights and hyperparameters were optimized using the CMA- ES software. Each kernel weight optimization is stopped using CMA-ES after 2500 function evaluations (i.e. cross-validations). Optimizations of hyperparameters are stopped after 250 function evaluations. The authors have provided 5 different sets of weights namely belonging to local optima obtained through the optimaization procedure.

Since these kernel are **positive semi-definite** we derived the related euclidean distance defined as:

$$d(X,Y) = \sqrt{K(X,X) + K(Y,Y) - 2K(X,Y)}$$

and we used it to compute a set of 5 dissimilarity matrices. For computational reason we decided to compute 1200 order matrices with a half of the objects belonging to the positive class and the others to the negative taking into account the original priors of the given dataset.

3.3 Flowcytometry data

The data used for the research were contained in four datasets provided by Dr.Marius Nap and Nick van Rodijnen from Pathology Department of "De Wever Hospital", Heerlen, The Netherlands. Each dataset is made by 833 patterns (samples of tissue) described by 256 features: the wavelength channels of flowcytometer, divided in three classes (as described above) labelled: aneuploid, diploid and tetraploid. The first and last two histogram bins which contain only noise have not been taken into account, this leads to have a reduced number of feature (252 channels). Each value of the datasets represents the number of cells of the tissue sample that have been recognized through the particular wavelength which is the corresponding feature.

Making the histograms' information rescaling invariant In this work a simple procedure has been used to compute rescaling invariant distance matrices. These matrices are actually built in such a way that each distances between two objects is computed as the minimum of squared distances between a histogram H_k and all the possible rescaling (with a correction factor α_i) of all the remaining $n-1$ histograms.

$$\alpha_i = \alpha_{min} + \frac{i}{h}(\alpha_{max} - \alpha_{min})$$

$$\text{with } i \in [1...i_{max}] \text{ and } h = cost$$

If we consider $X(k)$ a continuous function as a polynomial interpolant starting from $\{k, X_k\}$ data points, then the function $\tilde{X}(k) = \gamma X(\alpha_i k)$ is the outcome of a rescaled $X(k)$ with the parameters α_i and γ that lead to a horizontal and vertical rescaling. Then we can consider $\tilde{X}_k$ as the discrete version of the rescaled function $\tilde{X}(k)$ and then consider it as an histogram. The correction factor α_i

varies accordingly as described above, and induces an horizontal stretching to the histogram's shape, the γ parameter is used to fullfill the unit area constraint then it is equal to the inverse of the rescaled histogram's area.

$$\gamma = \frac{1}{\Sigma_{k=1}^{n} |X_k|}$$

Given X_k and Y_k two histograms the squared euclidean distance is defined as follows:

$$d_2{}^2 = \|X_k - Y_k\|_2^2 = \Sigma_{k=1}^{n} [X_k - Y_k]^2$$

The rescaling invariant distance is defined as:

$$d_{inv}(X_k, Y_k) = \min[d^*(X_k, Y_k), d^*(Y_k, X_k)]$$

$$\text{where } d^*(X_k, Y_k) = \min_{\alpha_i}[d_2^2(\tilde{X}_k, Y_k)]$$

This nonnegative function $d_{inv}(X_k, Y_k)$ describing the "rescaling invariant distance" between pairs of histograms taken from a given set satisfies the conditions to be a metric for the histograms' space.

3.4 M-feat

This dataset consists of features of handwritten numerals ('0'–'9') extracted from a collection of Dutch utility maps. Six different feature sets are extracted and stored separately.

1. 76 Fourier coefficients of the character shapes.
2. 216 profile correlations.
3. 64 Karhunen-Loeve coefficients.
4. 240 pixel averages in 2 x 3 windows.
5. 47 Zernike moments.
6. 6 morphological features.

This public domain dataset can be obtained from the UCI repository [18]. We used Euyclidean distances in each of the 6 feature spaces.

4 Experimental setup

We have conducted our experiments using the four datasets mentioned above and applying five different combination techniques (the four mentioned before: Forward selection, NCA, MCML, Fisher and the simple normalized sum) compared with the best performing individual ones. The performance measure used in our experiments has been the classification error of a linear support vector machine ([19]; [20]) in the 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, 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 optmization procedures (NCA, MCML and Fisher) the weights have been internally (in the routines) normalized with the Froebenious norm. The linear support vector classifier has been trained on the determined train sets and the phase has been employed making of the test sets computed with the weights previously computed on the train sets. 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 portion of 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 heve 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 indivual matrices.

5 Results

Our experimental results are provided in the form of the following tables (1, 2, 3). They show the classification errors using a linear support vector machine (libsvm with default parameters), the given values are the means (and standard deviations) of the classification errors computed as described previously by two-fold crossvalidation repeated 40 times for the four given datasets making use of six different approaches. These results show that for all the studied datasets the five methods that make use of combinations of the given matrices outperform the BIO (Best Individual Ones) that shows the error for the best dissimilarity matrix according 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 1) and in particular for the full collection of 44 chicken pieces matrices (table 2) this procedure scores even not significantly 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 (Normalized Sum) 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 Fisher technique is much worse than the other methods. These results might therefore suggest that for multimodal data distributions a normalized sum 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. From table 3 we can see that obviously the BIO results are still in line with the previous ones, and the NS performances are in this case heavily deteriorated. It is also clear that the NCA and MCML techniques are al-

ways better than the other approaches (with the 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.

In figure 1 we show the classification errors (top) for the Bio data using NCA (left) and MCML (right) compared with the normalized objective function value (bottom) as a function of the number of iterations using a vector of ones (in this case 5), corresponding to equal weights, for initialization. This example shows that for these two methods the classification error does not decrease as the objective function values. This further explains the results previously described (worse than NS as from 1).

Table 1. 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)
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)

Table 2. Classification errors (standard deviations) for the full chicken pieces dataset (44 matrices)

	Combining methods					
Datasets	BIO	FS	NS	NCA	MCML	Fisher
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)

Table 3. Classification errors (standard deviations) for the given datasets with additional noise

	Combining methods					
Noisy data	BIO	FS	NS	NCA	MCML	Fisher
m_feat	4.2 (0.8)	12.7 (15.7)	10.2 (1.5)	**3.0 (0.5)**	**3.0 (0.5)**	12.8 (9.1)
flow_cyto	30.9 (1.8)	28.4 (15.9)	44.3 (0.9)	**20.5 (2.0)**	22.3 (2.3)	39.7 (7.3)
chicken_pieces	**8.3 (2.0)**	**13.8 (20.7)**	39.6 (4.3)	11.7 (2.5)	12.8 (2.6)	37.2 (14.7)
Bio_data_lkc	**8.9 (1.6)**	12.1 (7.0)	18.6 (1.9)	10.8 (1.4)	10.7 (1.4)	20.0 (6.9)

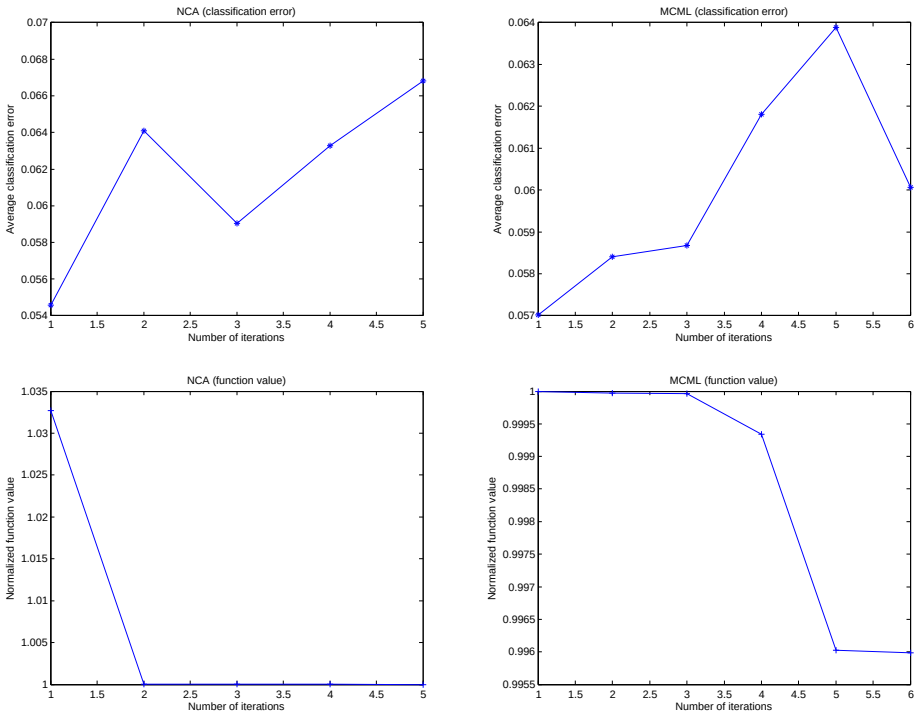

Fig. 1. Classification error and value of objective functions (NCA left and MCML right) as a function of the number of iterations starting from weights equal to one (NS) computed on the Bio dataset

6 Discussion and Conclusion

Previous works in the field of combining dissimilarity representations ([1], [2], [3]) 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. Now 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.

A significant aspect of the presented procedure is the normalization of the dissimilarities. In this study this is done by the largest distance. In other experiments we normalized by the average dissimilarity, which is less outlier sensitive and showed similar results.

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 does not hold for multi-modal data distributions. In case of particularly noisy data NCA and MCML are showing the best performances for all the studied datasets.

This work a study of combining different sources of information namely dissimilarity matrices originating from different measurements or form different computations of them. We have shown that combining before the training stage generally helps and that using techniques previously used for metric learning [14] these linear combinations can lead to even better results. In the future we will investigate further the influence of the multi-modality of the data distributions.

Acknowledgements

The authors acknowledge financial support from the FET programme within the EU FP7, under the SIMBAD project (contract 213250). They also like to thank Dr.Marius Nap and Nick van Rodijnen from the Pathology Department of the Atrium Medical Center in Heerlen, The Netherlands for providing the flowcytometry datasets.

References

1. Ibba, A., Duin, R.: A Multiscale Approach in Combining Classifiers in Dissimilarity Representations
2. Pekalska, E., Duin, R.: On combining dissimilarity representations. Lecture notes in computer science (2001) 359–368
3. Pekalska, E., Skurichina, M., Duin, R.: Combining dissimilarity-based one-class classifiers. In: Multiple classifier systems: 5th international workshop, MCS 2004, Cagliari, Italy, June 2004: proceedings, Springer-Verlag New York Inc (2004) 122
4. de Diego, I.M., Moguerza, J.M., Muñoz, A.: Combining kernel information for support vector classification. In Roli, F., Kittler, J., Windeatt, T., eds.: Multiple Classifier Systems. Volume 3077 of Lecture Notes in Computer Science., Springer (2004) 102–111
5. Joachims, T., Cristianini, N., Shawe-Taylor, J.: Composite kernels for hypertext categorisation. In Brodley, C.E., Danyluk, A.P., eds.: ICML, Morgan Kaufmann (2001) 250–257
6. Lee, W.J., Verzakov, S., Duin, R.P.W.: Kernel combination versus classifier combination. In Haindl, M., Kittler, J., Roli, F., eds.: MCS. Volume 4472 of Lecture Notes in Computer Science., Springer (2007) 22–31

7. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification (February 2009)
8. Yang, L., Jin, R., Sukthankar, R., Liu, Y.: An efficient algorithm for local distance metric learning. In: AAAI, AAAI Press (2006)
9. Pękalska, E., Duin, R.: The Dissimilarity Representation for Pattern Recognition. Foundations and Applications. World Scientific, Singapore (2005)
10. Avriel, M.: Nonlinear programming: analysis and methods. Dover Pubns (2003)
11. Cristianini, N., Kandola, J., Elissee, A.: On kernel target alignment. Advances in Neural Information Processing Systems 14 (2001)
12. Globerson, A., Roweis, S.: Metric learning by collapsing classes. Advances in Neural Information Processing Systems **18** (2006) 451
13. Goldberger, J., Roweis, S., Hinton, G., Salakhutdinov, R.: Neighbourhood components analysis. Advances in Neural Information Processing Systems **17** (2005) 513–520
14. Woznica, A., Kalousis, A., Hilario, M.: Learning to combine distances for complex representations. In: Proceedings of the 24th international conference on Machine learning, ACM (2007) 1038
15. Kullback, S., Leibler, R.: On information and sufficiency. The Annals of Mathematical Statistics **22**(1) (1951) 79–86
16. Bunke, H., Buhler, U.: Applications of approximate string matching to 2D shape recognition. Pattern recognition **26**(12) (1993) 1797–1812
17. Hulsman, M., Reinders, M., de Ridder, D.: Evolutionary Optimization of Kernel Weights Improves Protein Complex Comembership Prediction. IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) **6**(3) (2009) 427–437
18. Murphy, P.M., Aha, D.W.: UCI repository of machine learning databases (1992) Department of Information and Computer Science. University of California at Irvine. `anonymous ftp from archive.ics.uci.edu/ml/datasets/`.
19. Scholkopf, B., Smola, A., Williamson, R., Bartlett, P.: New support vector algorithms. Neural Computation **12**(5) (2000) 1207–1245
20. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. (2001) Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.