On the Informativeness of Asymmetric Dissimilarities

Yenisel Plasencia-Calaña[1 2], Veronika Cheplygina[2], Robert P. W. Duin[2], Edel García-Reyes[1], Mauricio Orozco-Alzate[3] David M. J. Tax[2], and Marco Loog[2]

[1] Advanced Technologies Application Center. 7ma A ♯ 21406, Playa, Havana - 12200, Cuba.
{yplasencia,egarcia}@cenatav.co.cu
[2] Pattern Recognition Laboratory, Delft University of Technology, The Netherlands
{v.cheplygina,D.M.J.Tax,m.loog}@tudelft.nl, r.duin@ieee.org
[3] Departamento de Informática y Computación, Universidad Nacional de Colombia - Sede Manizales. Kilómetro 7 vía al Aeropuerto, Campus La Nubia – Bloque Q, Piso 2, Manizales, Colombia
morozcoa@unal.edu.co

Abstract. A widely used approach to cope with asymmetry in dissimilarities is by symmetrizing them. Usually, asymmetry is corrected by applying combiners such as average, minimum or maximum of the two directed dissimilarities. Whether or not these are the best approaches for combining the asymmetry remains an open issue. In this paper we study the performance of the extended asymmetric dissimilarity space (EADS) as an alternative to represent asymmetric dissimilarities for classification purposes. We show that EADS outperforms the representations found from the two directed dissimilarities as well as those created by the combiners under consideration in several cases. This holds specially for small numbers of prototypes; however, for large number of prototypes the EADS may suffer more from overfitting than the other approaches. Prototype selection is recommended to overcome overfitting in these cases.

1 Introduction

Statistical and structural representations of patterns are two complementary approaches in pattern recognition. Recently, dissimilarity representations [14, 10] arose as a bridge between these representations. Dissimilarities can be computed from the original objects, but also on top of features or structures such as graphs or strings. This provides a way for bridging the gap between structural and statistical approaches. Dissimilarities are also a good alternative when the definition and selection of good features can be difficult or intractable (e.g. the search for the optimal subset of features has a computational complexity of $O(2^n)$, where n is the number of features) while a robust dissimilarity measure can be defined more easily for the problem at hand.

The classification of objects represented in a dissimilarity space (DS) has been an active research topic [16, 15, 17, 20, 4], but not much attention has been paid to the treatment of the asymmetry that can be present in the dissimilarities. Most traditional classification and clustering methods are devised for symmetric dissimilarity matrices, and therefore cannot deal with asymmetric input. In order to be suitable for these methods, asymmetric dissimilarities need to be symmetrized, for instance by averaging the matrix with its transpose. However, in the dissimilarity space, symmetry is not a required property and therefore a wider range of procedures for classification can be applied.

Asymmetric dissimilarity or similarity measures can arise in several situations; see [9] for a general analysis of the causes of non-Euclidean data. Asymmetry is common in human judgments. Including expert knowledge in defining a (dis)similarity measure, such as for fingerprint matching [4], may lead to asymmetry. In general, matching processes may often lead to asymmetric dissimilarities. Exact matches are often impossible and suboptimal procedures may lead to different matches from A to B than from B to A.

Symmetrization by averaging is widely used before embedding asymmetric dissimilarity data into (pseudo-)Euclidean spaces [14]. The use of a positive semi-definite matrix $K^T K$, where K denotes a nonsymmetric kernel [21] is also proposed in the context of kernel-based classification to make the kernel symmetric. A comparative study of methods for symmetrizing the kernel matrix for the application of the support vector machine (SVM) classifier can be found in [13]. While such methods that require symmetrized matrices show good results, it remains an open question whether asymmetry is an undesirable property, or that it, perhaps, contains useful information that is disregarded during symmetrization.

In this paper we explore using the information from asymmetric dissimilarities by concatenating them into an extended asymmetric dissimilarity space (EADS). Following up on [18], we investigate a broader range of circumstances where EADS may be a good choice for representation, and compare EADS to the directed dissimilarities, as well as to several symmetrization methods. The representation is studied for two shape matching and two multiple instance learning (MIL) problems. We show that EADS is able to outperform the directed and symmetrized dissimilarities, especially in cases where both directed dissimilarities are informative. It must be noted that EADS doubles the dimensionality of the problem, which may not be desirable. Therefore, we also include results using prototype selection in order to compare dissimilarity spaces with the same dimensionality, and show that EADS also leads to competitive results in the examples considered.

We begin with a number of examples that lead to asymmetric dissimilarities in Section 2. The dissimilarity space is explained in Section 3. Ways of dealing with asymmetry are then described: symmetrization (Section 4) and the proposed EADS (Section 5). Experimental results and discussion are provided in Section 6, followed by the conclusions in Section 7.

2 Asymmetric Dissimilarities

Although our notions of geometry may indicate otherwise, asymmetry is a natural characteristic when the concept of similarity or proximity is involved. Just think of a network of roads, where the roads can be one-way streets and one street is longer than the other. It is then clear that traveling from A to B may take longer than returning from B to A. Asymmetric dissimilarities also appear in human judgments [1]: it may be more natural to say that "Dutch is similar to German" than "German is similar to Dutch" because more people might be familiar with the German language and it is therefore a better point of reference for the comparison. Interestingly, this is also evidenced by the number of hits in Google: about ten times as many for the "Dutch is similar to German" sentence. When searching for these sentences in Dutch, the reverse is true.

Here we provide two examples of pattern recognition domains which may also naturally lead to asymmetric dissimilarities.

2.1 Shapes and Images

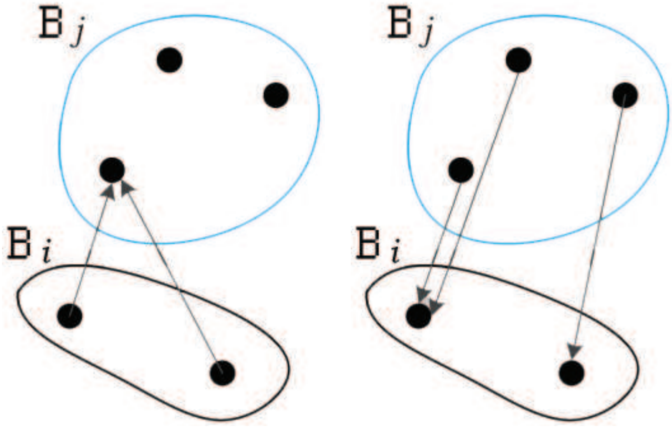
One possible cause of asymmetry is that the distances used directly on raw data such as images may be expensive to compute accurately. For example in [3], the edit distance used between shapes is originally symmetric. The distance has the problem that the returned values are different if the starting and ending points of the string representation of the shape is changed. In order to overcome this drawback, an improved rotation invariant distance was proposed. The computation of the new distance suffers from a higher computational complexity. Therefore, suboptimal procedures are applied in practice and, as a consequence, the distances returned are asymmetric.

In template matching, the dissimilarity measure may be designed to compute the amount of deformation needed to transform one image into the other as in [12]. The amount of deformation required to transform image I_j into image I_k is generally different from the amount of deformation needed to transform image I_k into I_j. This makes the resulting dissimilarity matrix asymmetric.

2.2 Multiple Instance Learning

Multiple instance learning (MIL) [6] extends traditional supervised learning methods in order to learn from objects that are described by a set (*bag*) of feature vectors (*instances*), rather than a single feature vector only. The bag labels are available, but the labels of the individual instances are not. A bag with n_i instances is therefore represented as (B_i, y_i) where $B_i = \{x_{ik}; k = 1...n_i\}$. In this setting, traditional supervised learning techniques cannot be applied directly.

It is often assumed that the instances have (hidden) labels which influence the bag label. For instance, one assumption is that a bag is positive if and only if at least one of its instances is positive. Such positive instances are also called concept instances. One application for MIL is image classification. An image with several regions or segments can be represented by a bag of instances, where each instance corresponds to a segment in the image. For images that are positive for the "Tiger" class, concept instances are probably segments containing (parts of) a tiger, rather than segments containing plants, trees and other surroundings. Patch-based Probabilistic Image Quality Assessment for Face Selection and Improved Video-based Face Recognition One of the approaches to MIL is to learn on bag level, by defining kernels [11] or (dis)similarities [22, 5] between bags. Such dissimilarities are often defined by matching the instances of one bag to instances of another bag, and defining a statistic (such as average or maximum) over these matches. This creates asymmetric dissimilarities, as illustrated in Fig.1.



(a) Dissimilarity from B_i to B_j (b) Dissimilarity from B_j to B_i

Fig. 1. Asymmetry in bag dissimilarities. The minimum distances of one bag's instances are shown. In this paper, the bag dissimilarity is defined as the average of these minimum distances.

The direction in which the dissimilarity is measured defines which instances influence the dissimilarity. When using a positive prototype, it is important that the concept instances are involved, as these instances are responsible for the differences between the classes. Therefore, for positive prototypes it is expected that the dissimilarity from the prototype to the bag is more informative than the dissimilarity from the bag to the prototype. A more detailed explanation of this intuition is given in [5].

3 Dissimilarity Space

The DS was proposed in the context of dissimilarity-based classification [14]. It was postulated as a Euclidean vector space, implying that classifiers proposed for feature spaces can be used there as well. The motivation for this proposal is that the proximity information is more important for class membership than features [14]. Let $R = \{r_1, ..., r_k\}$ be the representation set, where k is its cardinality. This set is usually a subset of the training set T, though a semi-supervised approach with more prototypes than training objects may be preferable [7]. In order to create the DS, using a proper dissimilarity measure d, the dissimilarities of training objects to the prototypes in R are computed. The object representation is a vector of the object's dissimilarities to all the prototypes. Therefore, each dimension of the DS corresponds to the dissimilarities to some prototype. The representation $\mathbf{d}_x$ of an object x is:

$$\mathbf{d}_x = [d(x, r_1) \; ... \; d(x, r_k)] \tag{1}$$

3.1 Prototype Selection

Prototype selection has been proposed for the dimension reduction of DS [16]. Supervised (wrapper) and unsupervised (filter) methods can be considered for this purpose as well as different optimization strategies to guide the search. They select the 'best' prototypes according to their criterion. The selected prototypes are used for the generation of the DS. Prototype selection allows one to obtain low-dimensional spaces avoiding as much as possible a decrease in performance (e.g. classification accuracy). Therefore, they are very useful to achieve a trade-off between the desirable properties of compact representation and reasonable classification accuracy. The approach considered in this study for selecting prototypes is the forward selection optimizing the leave-one-out (LOO) nearest neighbour (1-NN) error (so supervised) in the dissimilarity space for the training set. It starts from the empty set, and sequentially adds the prototype that together with the selected ones ensures the best 1-NN classification accuracy.

4 Combining the Asymmetry Information

For two point sets, there are different ways to combine the two directed asymmetric dissimilarities. The maximum, minimum and average are used extensively and are very intuitive. Let $A = \{a_1, ..., a_k\}$ and $B = \{b_1, ..., b_l\}$ be two sets of points, and $D_1 = d(A, B)$ and $D_2 = d(B, A)$ the two directed dissimilarities. The maximum, minimum and average combiners are defined in (2) to (4) respectively:

$$max(A, B) = \max(D_1, D_2) \tag{2}$$

$$min(A, B) = \min(D_1, D_2) \tag{3}$$

$$avg(A, B) = \frac{1}{2}(D_1 + D_2) \tag{4}$$

All these rules for combining asymmetry information ensure a symmetric measure.

5 Extended Asymmetric Dissimilarity Space

For the purpose of combining the asymmetry information in both directions, we study the EADS. From the two directed dissimilarities D_1, D_2, we have that $D_i \rightarrow X_i \in \mathbb{R}^k, i = 1,2$ represents the mapping of the dissimilarities to the dissimilarity space. The EADS is constructed by: $[D_1 \; D_2] \rightarrow X_1 \times X_2 \in \mathbb{R}^{k \times 2}$, which means that the extended space is the Cartesian product of the two directed spaces. Given the prototypes $R = \{r_1, ..., r_k\}$, the representation of an object in the EADS is defined by:

$$\mathbf{d}_x = [d(x, r_1) \; ... \; d(x, r_k) \; d(r_1, x) \; ... \; d(r_k, x)] \tag{5}$$

In the case that we have the full dissimilarity matrix using all training objects as prototypes, the EADS is constructed from the concatenation of the original matrix and its transpose. Rows of this new matrix correspond to the representation of objects in the EADS. As a result, the dimension of the EADS space is twice the dimension of the DS. Classifiers can be trained in the EADS in the same way they are trained in the DS. By doubling the dimension, the expressiveness of the representation is increased. This may be particularly useful when the number of prototypes is not very large. When the number of prototypes is large compared to the number of training objects, the EADS is expected to be more prone to overfitting than any of the symmetrized approaches.

Despite the fact that in the EADS symmetric distances or similarity measures can be used on top of the asymmetric representation this does not mean that we are not exploiting the asymmetry information present in the original dissimilarities. The original asymmetric dissimilarities in the two directions are used in the object representation that is the input for classifiers in the EADS. These classifiers can use any symmetric distance or kernel computed on top of the representation.

Note that if the asymmetry does not exist in the measure, the representation of objects in the EADS contains the same information replicated. These redundancies in the best case lead to the same classification results as in the standard DS using only one direction [18]. However, it may even be counterproductive since it may lead to overfitting and small sample size problems for some classifiers. Therefore, doubling the dimension is not the cause for possible classification improvements when using the EADS. The fact that the two asymmetric dissimilarities are taken into account in the representation is what may help the classifiers to improve their outcomes.

6 Experiments

In this section we first describe the datasets and how the corresponding dissimilarity matrices are obtained. This is followed by the experimental setup and a discussion of the results.

6.1 Datasets

The dissimilarity dataset Chickenpieces-35-45 is computed from the Chickenpieces image dataset [3]. The images are in binary format representing silhouettes from five different parts of the chicken: wing (117 samples), back (76), drumstick (96), thigh and back (61), and breast (96). From these images the edges are extracted and approximated by segments of length 35 pixels, and a string representation of the angles between the segments is derived. The dissimilarity matrix is composed by edit distances between these strings. The cost function between the angles is defined as the difference in case of substitution, and as 45 in case of insertion or deletion.

The Zongker digit dissimilarity data is based on deformable template matching. The dissimilarity measure was computed between 2000 handwritten NIST digits in 10 classes. The measure is the result of an iterative optimization of the non-linear deformation of the grid [12].

AjaxOrange is a dataset from the SIVAL multiple instance datasets [19]. The original dataset has 25 distinct objects (such as bottle of dish soap called AjaxOrange) portrayed against 10 different backgrounds, and from 6 different orientations, resulting in 60 images for each object. This dataset has been converted into 25 binary MIL datasets by taking one class (AjaxOrange) in this case as the positive class (with 60 bags), and all others (with 1440 bags) as the negative one. Each image is represented by a bag of segments, and each segment is described by a feature vector with color and texture features.

The dissimilarity of two images is computed by what we call the "meanmin" dissimilarity, which is similar to modified versions of the Hausdorff distance:

$$d_{meanmin}(B_i, B_j) = \frac{1}{|B_i|} \sum_{x_{ik} \in B_i} \min_{x_{jl} \in B_j} d(x_{ik}, x_{jl}) \tag{6}$$

where $d(x_{ik}, x_{jl})$ is the squared Euclidean distance between two feature vectors.

Winter Wren is one of the binary MIL bird song datasets [2], created in a similar one-against-all way as SIVAL. Here, a bag is a spectrogram of an audio fragment with different birds singing. A bag is positive for a particular bird species (e.g. Winter Wren) if its song is present in the fragment. There are 109 fragments where the Winter Wren song is heard, and 439 fragments without it. Also here we use (6) to compute the dissimilarities.

The datasets and their properties are shown in Table 1. For each dissimilarity matrix we computed its asymmetry coefficient as follows:

$$AC = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \frac{|d_{ij} - d_{ji}|}{\min(d_{ij}, d_{ji}) + \epsilon} \tag{7}$$

where n is the number of objects in the dataset. This coefficient measures the average normalized difference of the directed dissimilarities and is 0 for symmetric data.

The formulation in (7) assumes that $d_{ij} \neq 0$ for $i \neq j$, which may not necessarily be true for dissimilarity data. In the case that $d_{ij} = d_{ji}$, a term ϵ with a very small value such as 0.0001 must be added in the denominator to avoid divisions by zero.

Table 1. Properties of the datasets used in this study, AC refers to the asymmetry coefficient from (7); the larger the AC, the larger the asymmetry

Dataset	# Classes	# Obj. per class	AC
ChickenPieces-35-45	5	117, 76, 96, 61, 96	0.08
Zongker	10	10×200	0.18
AjaxOrange	2	60, 1440	0.31
Winter Wren	2	109, 439	0.23

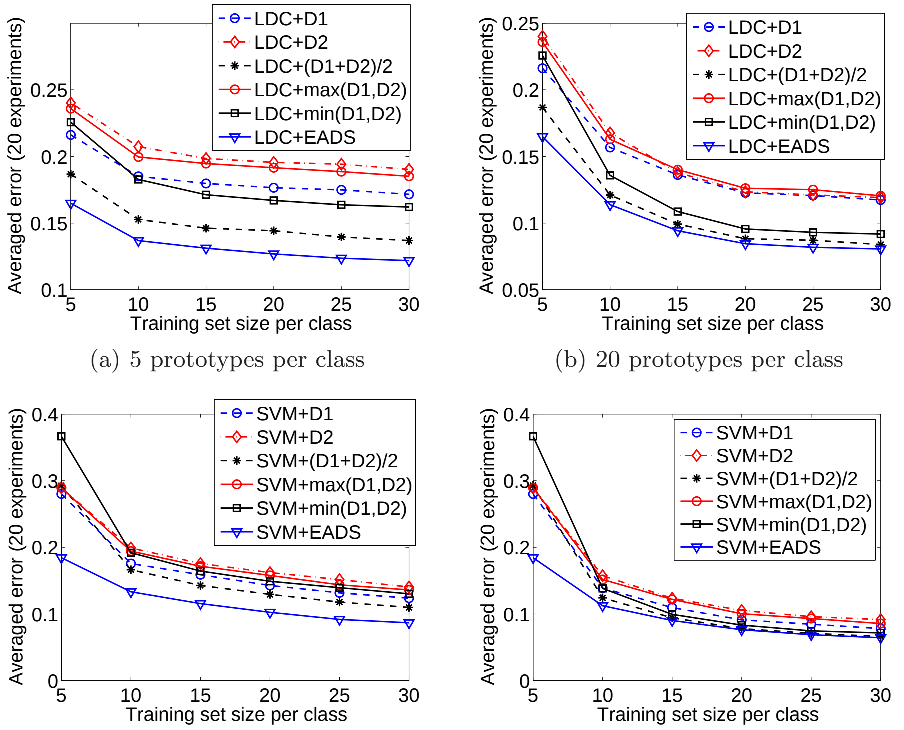
6.2 Experimental Setup

For each of the dissimilarity datasets, we evaluate the performances using asymmetric dissimilarity measures D_1 and D_2, the symmetrized measures (using minimum, average and maximum) and the EADS.

The classifiers compared are the linear discriminant classifier (LDA, but denoted LDC in our experiments) and the SVM, both in the dissimilarity space and implemented in PRTools [8]. For LDC we use regularization parameters $R = 0.01$ and $S = 0.9$, for SVM we use a linear kernel and a regularization parameter $C = 100$. These parameters show reasonable performances on all the datasets under investigation, and are, therefore, constant across all experiments and not optimized to fit a particular dataset.

We provide learning curves over 20 runs for each dissimilarity / classifier combination, for increasing training sizes from 5 to 30 objects per class. In each of the learning curves, the number of prototypes is fixed to either 5 or 20 per class in order to explore the behavior with a small and a large representation set size. This means that the dimensionality of the dissimilarity space is the same for D_1, D_2 and the symmetrized versions, but twice as much for the EADS. The approaches compared are:

- DS resulting from the computation of dissimilarities in the direction from the objects to the prototypes (D_1).
- DS resulting from the computation of dissimilarities from the prototypes to the objects (D_2).
- DS resulting from averaging the dissimilarities in the two directions ($(D_1 + D_2)/2$).
- DS resulting from the maximum of the two dissimilarities ($\max(D_1, D_2)$).
- DS resulting from the minimum of the two dissimilarities ($\min(D_1, D_2)$).
- The extended asymmetric dissimilarity space (EADS).



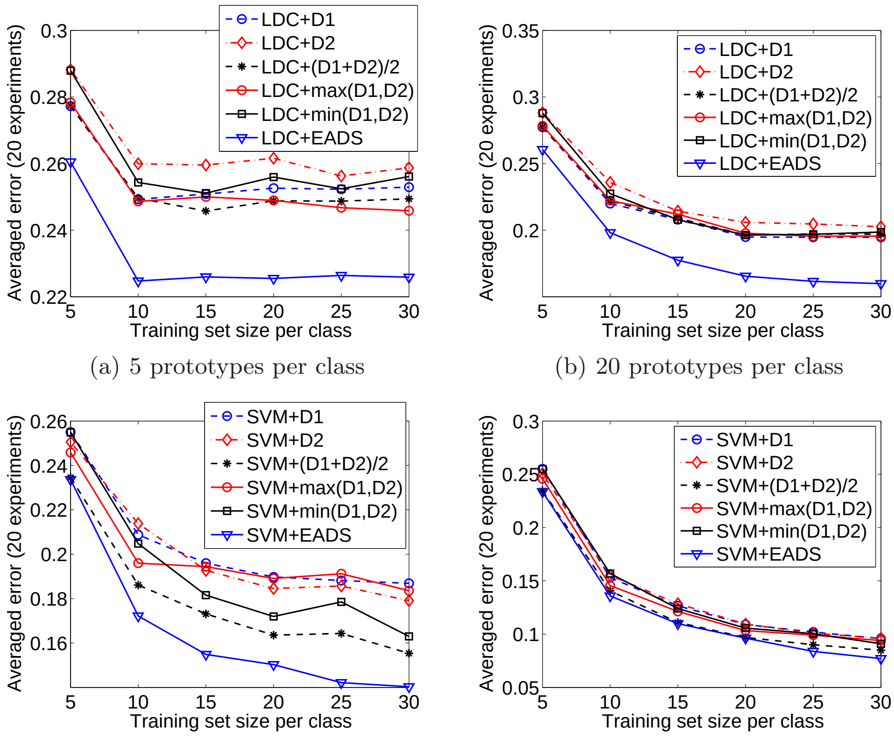
(a) 5 prototypes per class (b) 20 prototypes per class

(c) 5 prototypes per class (d) 20 prototypes per class

Fig. 2. LDC and SVM classification results in dissimilarity spaces for Zongker dataset

6.3 Results and Discussion

In Figs. 2 and 3 it can be seen from the results on the Zongker and Chicken Pieces datasets that the EADS outperforms the other approaches. This is especially true for a small number of prototypes (see Figs. 2 and 3 (a) and (c)). The results of the different approaches become more similar for the representation set of 20 prototypes per class, especially when SVM is used (see Figs. 2 and 3 (d)). The EADS is better than the individual spaces created from the directed dissimilarities, one explanation for this is that the directed dissimilarities provide complementary information so together they are more useful than individually. The EADS contains more information of the relations between the objects than an individual directed DS. The maximum operation is usually very sensitive to noise and outliers what explains its bad performance. The maximum dissimilarity makes objects belonging to the same class more different. These higher differences inside the class are likely to contain noise since objects of the same class should potentially be more similar. The average is more robust than maximum since it combines the information from both directed dissimilarities avoiding in some degree the influence of noise and outliers. Still, by averaging



(a) 5 prototypes per class

(b) 20 prototypes per class

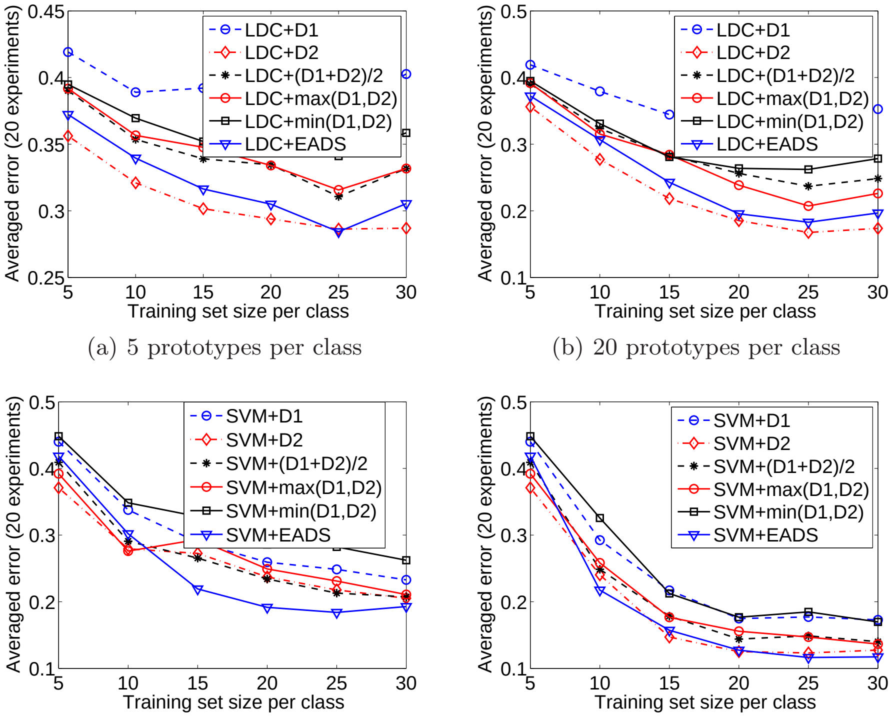
(c) 5 prototypes per class

(d) 20 prototypes per class

Fig. 3. LDC and SVM classification results in dissimilarity spaces for Chicken Pieces dataset

we may hamper the contribution of a very good directed dissimilarity if there is a noisy counterpart. The EADS may improve upon the average because the EADS does not obstructs the contribution of a good directed dissimilarity. The minimum operator is usually worse than EADS and averaging. One possible cause is that by using the minimum, the representation of all the objects is homogenized to some extent because for objects belonging to different classes the separability is decreased by selecting the minimum dissimilarity. Therefore, some discriminatory power is lost.

In AjaxOrange, it is an important observation that D_2 is more informative than D_1, especially for the LDC classifier (see Fig. 4 (a) and (b)). D_2 means that the dissimilarities are measured from the prototypes to the bags. The *meanmin* dissimilarity in (6) therefore insures that, for a positive prototype, the positive instances (the AjaxOrange bottle) influence the dissimilarity value by definition, as all instances of the prototype have to be matched to instances in the training bag. Measuring the dissimilarity to positive prototypes, on the other hand, may result in very similar values for positive and negative bags because of identical backgrounds, therefore creating class overlap.



(a) 5 prototypes per class (b) 20 prototypes per class

(c) 5 prototypes per class (d) 20 prototypes per class

Fig. 4. LDC and SVM classification results in dissimilarity spaces for AjaxOrange dataset

Because D_1 contains potentially harmful information, the combining methods do not succeed in combining this information from D_1 and D_2 in a way that is beneficial to the classifier. This is particularly evident for the LDC classifier (see Fig. 4 (a) and (b)), where only EADS has similar (but still worse) performance than D_2. For the SVM classifier, EADS performs well only when a few prototypes are used, but as more prototypes (and more harmful information from D_1) are involved, there is almost no advantage over D_2 alone.

From the results reported in Fig. 5 for Winter Wren, we again see that D_2 is more informative than D_1. However, what is different in this situation is that both directions contain useful information for classification, this is evident due to the success of the average, maximum and EADS combiners. The difference lies in the negative instances (fragments of other birds species, or background objects in the images) of positive bags. While in AjaxOrange, background objects are non-informative, the background in the audio fragments may be informative for the class of the sound. In particular, it is possible that some bird species are heard together more often: e.g. there is a correlation of 0.63 between the labels of Winter Wren and Pacific-slope Flycatcher. Therefore, also measuring

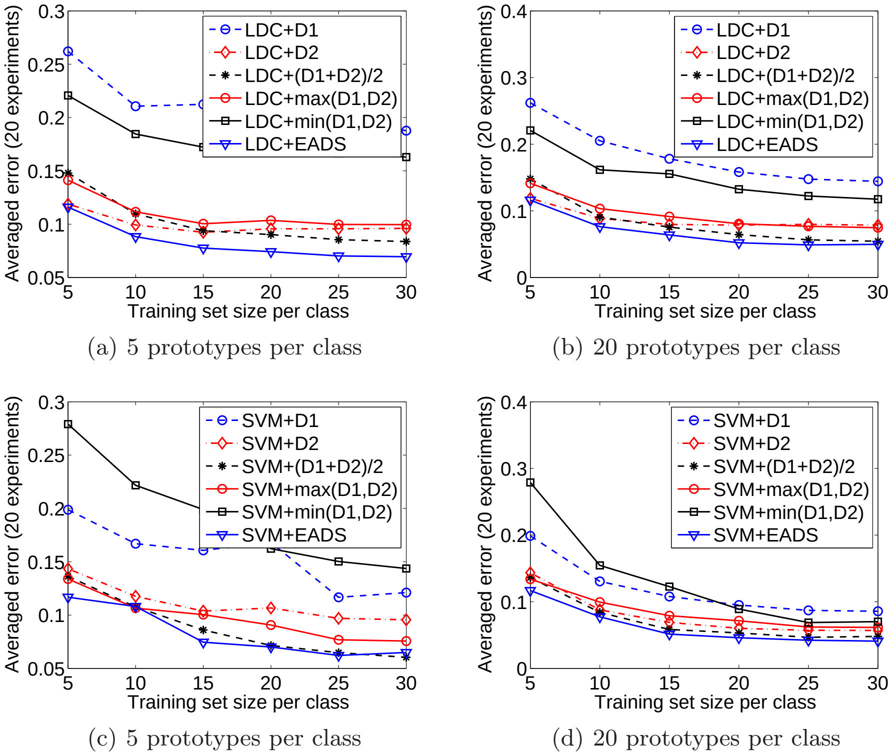
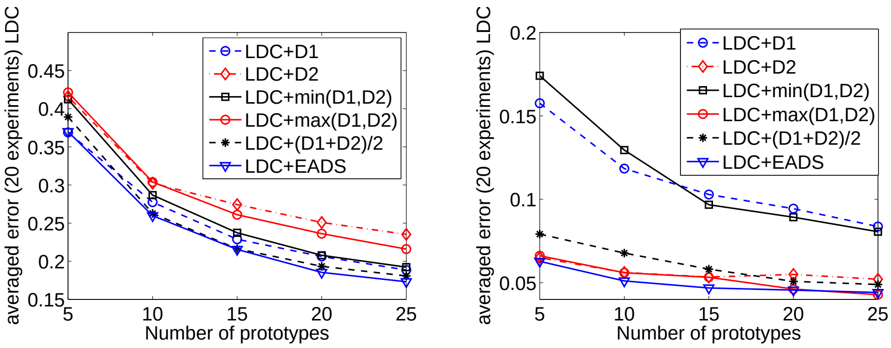


Fig. 5. LDC and SVM classification results in dissimilarity spaces for the Winter Wren dataset

dissimilarities to the prototypes produces dissimilarity values that are different for positive and negative bags.

The increased dimensionality of the EADS is one of the main problems of this approach, as in low sample size cases the increased dimensionality may lead to overfitting. In order to overcome this, prototype selection can be considered. We developed other experiments using prototype selection for all the spaces compared. A fixed training set size of 200 objects was used, leading to spaces of dimensionality 5, 10, 15, 20 and 25. The choice to perform the selection of the prototypes was the forward selection optimizing the LOO 1-NN classification error in the training set. One example of standard and MIL dissimilarity datasets were considered: the Zongker and Winter Wren. Prototypes are selected for EADS as it is usually done for a standard DS. Prototypes using the two directed dissimilarities are available as candidates but the prototype selection method may discard one of the two or maybe both if they are not discriminative according to the selection criterion. The EADS is compared now with the other spaces on the basis of the same dimensionality.



(a) Zongker (b) Winter Wren

Fig. 6. Classification results after prototype selection for the Zongker and Winter Wren datasets

From the results in Fig. 6 (a) it can be seen that, for the Zongker dataset, the best approaches are the EADS and the average. An interesting observation is that this dataset is intrinsically high-dimensional because the number of principal components (PCs) that retain 95% of the data variance is equal to 529. The average approach adds more information in each dimension since every dissimilarity encodes a combination of two. This implies that, for the dimensions considered that are small compared to 529, it performs as good as the EADS. On the contrary, the Winter Wren dataset is intrinsically low-dimensional, with the number of PCs retaining 95% of the data variance equal to 3. This is a possible explanation of why the EADS is the best in this case (see Fig. 6 (b)), because the average approach is likely to introduce noise.

One interesting issue of using prototype selection in EADS is that not only the dimensions are decreased, but also the accuracy of the EADS itself may be improved especially in the datasets where one of the directed dissimilarities is the best and the other is very bad (e.g. MIL datasets). The EADS without prototype selection in these cases may be worse than the best directed dissimilarity (see Fig. 4 (a) and (b)). However, by using a suitable prototype selection method in the EADS, only the prototypes from the best directed dissimilarity should be kept, and noisy prototypes from the bad directed dissimilarity should be discarded. This should make the results of the EADS similar to those of the best directed dissimilarity. This can be achieved if a proper prototype selection method is used. For example, in the prototype selection executed for the Winter Wren, where the first directed dissimilarity is remarkably better than the second, this can partially be seen. The method selected in one run 18 prototypes from the best directed dissimilarity in the set of 25 prototypes selected. Future work will include the study of suitable prototype selectors for EADS.

7 Conclusions

In this paper we study the EADS as an alternative to different approaches for dealing with asymmetric dissimilarities. The EADS outperforms the other approaches for a small number of prototypes in standard dissimilarity datasets, when both dissimilarities are about equally informative.

In MIL datasets, conclusions are slightly different because of the way the dissimilarities are created. It may be the case that the best option is one of the directed dissimilarities. However, if there is no knowledge on which directed dissimilarity is the best, the EADS may be the best choice. This especially holds when only a low number of prototypes is available.

It should be noted that the EADS increases the dimensionality as opposed to other combining approaches, therefore increasing the risk of overfitting. Prototype selection should be considered to keep the dimensionality low. After prototype selection, the EADS also shows good results in examples of intrinsically low and high-dimensional datasets. However, for intrinsically high-dimensional datasets, averaging is also worth considering as combining rule.

Our main conclusion is that asymmetry is not an artefact that has to be removed in order to apply embedding or kernel methods to the classification problem. On the contrary, asymmetric dissimilarities may contain very useful information, and it is advisable to consider the dissimilarity representation as a means to fully use this information.

References

1. Bowdle, B., Gentner, D.: Informativity and asymmetry in comparisons. Cogn. Psychol. 34(3), 244–286 (1997)
2. Briggs, F., Lakshminarayanan, B., Neal, L., Fern, X., Raich, R., Hadley, S., Hadley, A., Betts, M.: Acoustic classification of multiple simultaneous bird species: A multi-instance multi-label approach. J. Acoust. Soc. Am. 131, 4640 (2012)
3. Bunke, H., Bühler, U.: Applications of approximate string matching to 2D shape recognition. Pattern Recogn. 26(12), 1797–1812 (1993)
4. Bunke, H., Riesen, K.: Graph classification based on dissimilarity space embedding. In: Structural, Syntactic, and Statistical Pattern Recognition. pp. 996–1007. SSPR & SPR '08, Springer-Verlag, Berlin, Heidelberg (2008)
5. Cheplygina, V., Tax, D.M.J., Loog, M.: Class-dependent dissimilarity measures for multiple instance learning. LNCS, vol. 7626, pp. 602–610. Springer (2012)
6. Dietterich, T., Lathrop, R., Lozano-Pérez, T.: Solving the multiple instance problem with axis-parallel rectangles. Artif. Intell. 89(1-2), 31–71 (1997)
7. Dinh, C., Duin, R.P.W., Loog, M.: A study on semi-supervised dissimilarity representation. In: International Conference on Pattern Recognition (2012)
8. Duin, R.P.W., Juszczak, P., Paclik, P., Pekalska, E., De Ridder, D., Tax, D.M.J., Verzakov, S.: A Matlab toolbox for pattern recognition. PRTools version 3 (2000)
9. Duin, R.P.W., Pekalska, E.: Non-Euclidean dissimilarities: causes and informativeness. In: Proceedings of the 2010 joint IAPR international conference on Structural, Syntactic, and Statistical Pattern Recognition. pp. 324–333. SSPR & SPR'10, Springer-Verlag, Berlin, Heidelberg (2010)

10. Duin, R.P.W., Pekalska, E.: The dissimilarity space: bridging structural and statistical pattern recognition. Pattern Recogn. Lett. 33(7), 826–832 (2012)
11. Gärtner, T., Flach, P., Kowalczyk, A., Smola, A.: Multi-instance kernels. In: Proc. of the 19th Int. Conf. on Machine Learning. pp. 179–186 (2002)
12. Jain, A.K., Zongker, D.: Representation and recognition of handwritten digits using deformable templates. IEEE Trans. Pattern Anal. Mach. Intell. 19, 1386–1391 (December 1997)
13. Muñoz, A., de Diego, I.M., Moguerza, J.M.: Support vector machine classifiers for asymmetric proximities. In: Kaynak, O., Alpaydin, E., Oja, E., Xu, L. (eds.) Artificial Neural Networks and Neural Information Processing ICANN/ICONIP 2003, LNCS, vol. 2714, pp. 217–224. Springer Berlin Heidelberg (2003)
14. Pekalska, E., Duin, R.P.W.: The Dissimilarity Representation for Pattern Recognition: Foundations and Applications (Machine Perception and Artificial Intelligence). World Scientific Publishing Co., Inc., River Edge, NJ, USA (2005)
15. Pekalska, E., Duin, R.P.W.: Beyond traditional kernels: Classification in two dissimilarity-based representation spaces. IEEE Trans. Syst. Man Cybern. C, Appl. Rev. 38(6), 729–744 (2008)
16. Pekalska, E., Duin, R.P.W., Paclík, P.: Prototype selection for dissimilarity-based classifiers. Pattern Recogn. 39(2), 189–208 (2006)
17. Pekalska, E., Paclik, P., Duin, R.P.W.: A generalized kernel approach to dissimilarity-based classification. J. Mach. Learn. Res. 2, 175–211 (March 2002)
18. Plasencia-Calaña, Y., García-Reyes, E.B., Duin, R.P.W., Orozco-Alzate, M.: On using asymmetry information for classification in extended dissimilarity spaces. In: Álvarez, L., Mejail, M., Gómez, L., Jacobo, J.C. (eds.) CIARP. LNCS, vol. 7441, pp. 503–510. Springer (2012)
19. Rahmani, R., Goldman, S., Zhang, H., Krettek, J., Fritts, J.: Localized content based image retrieval. In: Proc. of the 7th ACM SIGMM International Workshop on Multimedia Information Retrieval. pp. 227–236. ACM (2005)
20. Riesen, K., Neuhaus, M., Bunke, H.: Graph embedding in vector spaces by means of prototype selection. In: Escolano, F., Vento, M. (eds.) GbRPR. pp. 383–393. LNCS, vol. 4538, Springer (2007)
21. Schölkopf, B., Mika, S., Burges, C.J.C., Knirsch, P., Müller, K.R., Rätsch, G., Smola, A.J.: Input space versus feature space in kernel-based methods. IEEE Trans. Neural Netw. 10(5), 1000–1017 (1999)
22. Tax, D.M.J., Loog, M., Duin, R.P.W., Cheplygina, V., Lee, W.J.: Bag dissimilarities for multiple instance learning. In: Similarity-Based Pattern Recognition. LNCS, vol. 7005, pp. 222–234. Springer (2011)