Towards cluster-based prototype sets for classification in the dissimilarity space

Yenisel Plasencia-Calaña[1,2], Mauricio Orozco-Alzate[3], Edel García-Reyes[1], and Robert P. W. Duin[2]

[1] Advanced Technologies Application Center. 7ma A ♯ 21406, Playa, Havana - 12200, Cuba.
{yplasencia,egarcia}@cenatav.co.cu

[2] Faculty of Electrical Engineering, Mathematics and Computer Sciences, Delft University of Technology, The Netherlands
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. The selection of prototypes for the dissimilarity space is a key aspect to overcome problems related to the curse of dimensionality and computational burden. How to properly define and select the prototypes is still an open issue. In this paper, we propose the selection of clusters as prototypes to create low-dimensional spaces. Experimental results show that the proposed approach is useful in the problems presented. Especially, the use of the minimum distances to clusters for representation provides good results.

Keywords: dissimilarity space, prototype selection, cluster-based prototypes

1 Introduction

The representation of objects is crucial for the success of a pattern recognition system. The feature space representation is the most common approach since a large number of techniques can be used. Dissimilarity representations [1] arose as an alternative and have been showing a good performance in several problems, where the dissimilarities may be computed by directly matching original objects [1] or on top of feature representations [2]. Three main approaches are presented in [1], the most promising being the dissimilarity space (DS).

In the DS, an object is represented by a vector of dissimilarities with other objects called prototypes. If a large set of prototypes is used, it leads to a high-dimensionality of the DS implying that computational costs of classification are increased as well as storage costs. In addition, a high-dimensionality leads to problems related to the “curse of dimensionality” and small sample sizes. Furthermore, high-dimensional representations are likely to contain noise since the intrinsic dimensionality of the data is usually small, leading to overfitting.

Prototype selection is a way to overcome these drawbacks. It has been studied [3] for reducing dimensions of DS with encouraging results. Several methods have been proposed such as Kcentres, Forward Selection (FS), Editing and Condensing, among others [3]. In these studies, the selected prototypes are objects. However, some efforts are also put in a different direction and, instead of objects, linear models are built, selecting out some of them for representation [4]. These studies showed that it is a feasible alternative to use a small number of carefully selected feature lines as prototypes instead of the original objects.

In this paper we study the selection of clusters for the generation of a low-dimensional generalized dissimilarity space (GDS). Our hypothesis is that clusters may be useful to obtain low-dimensional GDSs in case datasets are structured in clusters. A similar approach was presented in [5], however it was specifically developed for graph distances while our research is not restricted to graphs. Besides, they do not take into account the selection of the best clusters, while our goal is to find the clusters which allow a good classification with a decreased dimension of the space. We also included the subspace distance to clusters. Different approaches to compute the distances of the training and test objects to the clusters are presented. The paper is divided as follows. Section 2 introduces the DS and prototype selection. Section 3 describes the construction of the datasets based on cluster distances. Experimental results and discussions are provided in Sec. 4 followed by concluding remarks in Sec. 5.

2 Dissimilarity space

The DS was conceived with the purpose to address classification of data represented by dissimilarities that may be non-Euclidean or even non-metric. The dissimilarities of a training set X with a set of prototypes $R = \{r_1, ..., r_k\}$ are interpreted as coordinates in the DS. Thereby, the number of prototypes selected determines the dimension of the space. The DS was postulated as a Euclidean vector space, making suitable the use of statistical classifiers. The set of prototypes may satisfy $R \subseteq X$ or $R \cap X = \emptyset$. Once R is selected by any prototype selector, the dissimilarities of both training and test objects with R are computed. Let x be any training or test object and d a suitable dissimilarity measure for the problem at hand, the representation d_x of the object in the dissimilarity space is:

$$d_x = [d(x, r_1) \;\; d(x, r_2) \;\; ... \;\; d(x, r_k)]. \tag{1}$$

2.1 Prototype selection

Many approaches have been considered [2,3] for the selection of prototypes in the DS. Variants of wrapper or supervised methods [3] have been proposed. Other approaches are considered that use the distances or distribution of the prototypes over the dataset [2]; note that in these cases the class labels of the prototypes may not be needed. An interesting option is the genetic algorithm (GA) presented in [6]. The GA is an evolutionary method which uses heuristics in order to evolve an initial set of solutions (sets of prototypes) to better ones by using operations such as mutation and reproduction. Moreover, it is adequate to handle non-metric dissimilarities and it can find complicated relationships between the prototypes. For these reasons we propose to use a GA to select the clusters together with the leave-one-out nearest neighbor (LOO 1-NN) error in the DS as selection criterion. We adopt the same parameters for the GA as in [6]. Clusters present nice properties that good prototypes must have. For example, they do not provide redundant information since redundant or close objects must lie together in the same cluster and they cover the representation space better than a small set of objects.

3 Construction of models based on clusters

In this section we describe our methodology to construct the new dissimilarity datasets based on cluster distances computed from the originally given dissimilarities. In this study, the clusters are created per class by the Affinity Propagation algorithm [7]. In the clustering process representatives and their corresponding clusters emerge from a message-passing procedure between pairs of samples until stopping criteria are met. This method is reported to provide good clustering results. Furthermore, it is also of our convenience that it semi-automatically selects the proper number of clusters, emerging from the message-passing procedure but also from a user preference of the cluster representatives. The original dissimilarities must be transformed into similarities in order to apply the clustering procedure. We set the preferences for each object (i.e. the potential to be selected as cluster center) equal to the median similarity.

Different types of distances are used to measure the resemblance of objects with clusters such as: the minimum, maximum, average and subspace distances. The minimum distance is computed as the distance between the object and its nearest object in the cluster. The maximum distance is defined as the distance between the object and its farthest object in the cluster. The average distance is defined as the average of the distances between the object and all the cluster objects. The subspace distance is explained more carefully. Theory about it is sparse in the literature [8,9], especially for the case of data given in terms of non-metric dissimilarities. Therefore, one contribution of this paper is to describe the methodology to compute the (speeded-up) distance of objects to subspaces when data is provided in terms of non-metric dissimilarities.

The methodology to compute the subspace distance to clusters is as follows. First, a subspace is created for every cluster in order to compute the subspace distances. To achieve this, the set of dissimilarities is transformed into equivalent dot products (which can be interpreted as similarities) and centered according to the "double-centering" formula for each cluster:

$$S_{ij} = -\frac{1}{2}\left(D_{ij}^2 - \frac{1}{n}C_i - \frac{1}{n}C_j + \frac{1}{n^2}C_iC_j\right), \quad (2)$$

where D_{ij} is the dissimilarity between the cluster objects x_i and x_j, $C_i = \sum_j D_{ij}^2$, which is the i-th row sum of the dissimilarity matrix for the cluster objects, n is the number of objects in the cluster, and S_{ij} are the centered dot products. The eigendecomposition of S is performed and eigenvectors are sorted in descendent manner according to their eigenvalues. Only the eigenvectors associated with eigenvalues $\lambda > 0$ are used to compute the projections of new points to the subspace via the *Nyström* formula [10].

Each embedding coordinate of a cluster object x_i used to compute the kernel is given by $e_{ik} = \sqrt{\lambda_k}v_{ik}$ as for multidimensional scaling (MDS) [8], where λ_k is the k-th eigenvalue and v_{ik} is the i-th element of the k-th eigenvector of S, but the embedding for a new point is obtained via the *Nyström* approximation which is interpreted as the Kernel PCA projection [9] using S as the kernel matrix. The *Nyström* formula was generalized for extending MDS as suggested in [9], therefore, each embedding coordinate e_{ik} is computed by:

$$e_{ik}(x) = \frac{\sqrt{\lambda_k}}{\lambda_k}\sum_{i=1}^{n} v_{ik}S(x, x_i), \quad (3)$$

where x_i are the cluster objects and $S(x, x_i)$ is computed from a continuous version of the "double-centering" formula:

$$S(x, x_i) = -\frac{1}{2}\left(d(x, x_i)^2 - \frac{1}{n}\sum_j d(x, x_j)^2 - \frac{1}{n}\sum_j D_{ij}^2 + \frac{1}{n^2}\sum_{ij} D_{ij}^2\right). \quad (4)$$

$S(x, x_i)$ is a data-dependent kernel where $d(\cdot,\cdot)$ is the dissimilarity function. This *Nyström* embedding is applied to speed-up the embedding computation instead of recomputing the eigendecomposition including x in the whole process. However, in our case, the embedding is not directly used, instead, the embedding coordinates are used to compute the distance to the subspace. The squared subspace distance $d_L(x, L)^2$ is formulated as the difference between the squared length of the vector (its squared norm) given by $S(x, x)$ and the length of its projection on the space via *Nyström*:

$$d_L(x, L)^2 = S(x, x) - \sum_{k=1}^{m}\left(\frac{\sqrt{\lambda_k}}{\lambda_k}\sum_{i=1}^{n} v_{ik}S(x, x_i)\right)^2. \quad (5)$$

4 Experimental results

4.1 Datasets and experimental setup

The dissimilarity datasets were selected for the experiments based on the existence of clusters in the data. The Ionosphere dataset consists in radar data [11] where the $L1$ distance is used. The Kimia dataset is based on the shape contexts descriptor [12] computed for the Kimia shapes data [13]. The dissimilarity is based on sums of matching costs for the best matching points defining two shapes, plus the amount of transformation needed to align the shapes. The dissimilarity data set Chickenpieces-20-60 [14] is composed by edit distances from string representations of the angles between segments defining the contours of chicken pieces images. The Ringnorm dataset is the one presented in [15]; it is originally a 20-dimensional, 2-class data, where the first class is normally distributed with zero mean and covariance matrix 4 times the identity. The second class has unit covariance matrix and mean close to zero. We use only the first 2 features and the $L2$ distance. The characteristics of the datasets as well as the cardinality of the training sets used are presented in Table 1.

As classifier we used the support vector machine (SVM) classifier. For the SVM we used a linear kernel and a fixed appropriately selected cost parameter $C = 1$. Note that despite the fact that the curse of dimensionality was mentioned as a limitation of high-dimensional spaces, the SVM classifier is able to handle high dimensions well. This makes our comparisons more fair for the high-dimensional representations. However, the limitation was mentioned since in many applications people may want to use classifiers that suffer from the curse of dimensionality and resorting to low-dimensional representations by prototype selection is one option to overcome the problem. Our proposals are the following cluster-based methods: selection by GA of clusters created using minimum, maximum, average and subspace distances of training objects to the clusters. The cluster-based methods are compared with some of the best prototype selectors presented in the literature (which select objects as prototypes), with representatives of unsupervised and supervised methods: Forward selection [3] optimizing the LOO 1-NN error in the DS, Kcentres prototype selector [3], random selection, selection by GA of the best clusters centers, and selection by GA of the best prototypes from the whole candidate set. In addition, we compared the approach using all candidate objects as prototypes.

Table 1. Properties of the datasets used in this study, Symm. and Metric refers to whether the data is symmetric or metric, the $|T|$ column refers to the training set cardinality used for the experiments

Datasets	# Classes	# Obj. per class	Symm.	Metric	$\|T\|$
Ionosphere	2	225,126	yes	yes	140
Kimia	18	18 × 12	no	no	90
Rings	2	440,449	yes	yes	222
ChickenPieces-20-60	5	117,76,96,61,96	no	no	158

A set of 5 to 20 prototype clusters/objects are selected. However, the total number returned by the affinity propagation is about 25 clusters. Averaged errors and standard deviations over 30 experiments are reported in Table 2 for the dimension where the best result was obtained. Objects in each dataset are randomly split 30 times into training, representation, and test sets. Clusters are computed on the representation set which also contains the candidate objects for prototypes, the best clusters and objects are selected optimizing the criteria for the training set by which the classifiers are trained, and the final classification errors are computed for the test sets. We performed a t-test to find if the differences between the mean errors of the best overall result and the mean errors achieved by the other approaches was statistically significant, the level of significance used is 0.05. In the case that a cluster-based method was the best, the statistical significance is computed with respect to the non cluster-based approaches.

Table 2. Mean and standard deviation of errors over 30 experiments. The best overall result is reported for each dataset with the corresponding results of the other methods for the same dimension of the space (in parenthesis). When the difference of the best result with the other standard approaches is statistically significant, it is reported in bold.

Datasets Selectors	Ionosph(15)	Kimia(20)	Rings(20)	Chicken Pieces(20)
Clusters minimum	**0.063 ± 0.028**	**0.047 ± 0.032**	0.265 ± 0.0205	0.11 ± 0.025
Clusters maximum	0.09 ± 0.029	0.11 ± 0.054	**0.263 ± 0.0236**	0.15 ± 0.028
Clusters average	0.072 ± 0.023	0.06 ± 0.045	0.274 ± 0.0181	0.09 ± 0.024
Clusters subspace	0.073 ± 0.022	0.07 ± 0.048	0.276 ± 0.0193	0.088 ± 0.023
Random	0.086 ± 0.026	0.12 ± 0.057	0.274 ± 0.0181	0.17 ± 0.039
GA (whole set)	0.082 ± 0.028	0.1 ± 0.043	0.274 ± 0.0181	0.16 ± 0.028
GA (cluster centres)	0.085 ± 0.032	0.094 ± 0.05	0.275 ± 0.0177	0.15 ± 0.029
Forward selection	0.09 ± 0.027	0.12 ± 0.054	0.274 ± 0.0184	0.16 ± 0.036
Kcentres	0.082 ± 0.029	0.13 ± 0.061	0.274 ± 0.0181	0.15 ± 0.036
All	0.083 ± 0.033	0.068 ± 0.042	0.274 ± 0.0181	**0.077 ± 0.017**

4.2 Results and discussion

In Table 2 it can be seen that classification results in the GDS generated by selected clusters outperform the classification results in DS with selected objects as prototypes for the same dimensions of the spaces. For the Ionosphere and Kimia datasets the best method uses clusters with minimum distance, this is in agreement with previous findings for graph dissimilarities in [5]. In the Ionosphere and Kimia datasets, the selection of clusters using maximum distance is usually among the worse alternatives. This may be expected since it may be very sensitive to outliers. However, in the Rings dataset the clusters based on maximum distances provide the best overall result. In the case of Chicken Pieces, the best results are obtained using all objects as prototypes, perhaps because this dataset has a high intrinsic dimension (176) according to the number of significant eigenvalues of the covariance matrix in the DS. Therefore, in order to obtain good results, high-dimensional spaces are needed. However, the average and subspace distance to clusters outperformed the other approaches that create low-dimensional spaces.

Cluster-based approaches create irregular kernels which nonlinearly map the data to the GDS in a better way than the object-based approaches for the same dimensions. We computed the nonlinear mapping for the Rings data from the underlying feature space to a Hilbert space using a second degree polynomial kernel and applied SVM classification with this kernel and regularization parameter optimized. We corroborate that the results were very similar to the ones obtained using clusters in the dissimilarity space. Cluster-based prototypes allow one to apply linear classifiers with good results to originally nonlinear data. The same can be achieved by kernels and SVM if the dissimilarities are Euclidean (they are transformed to the equivalent kernel). However, the original SVM will not work anymore for a non-Euclidean dissimilarity matrix but a nonlinear mapping to the DS or GDS can still be achieved for non-Euclidean data (e.g. the Kimia dataset).

The main disadvantage of using cluster-based prototypes compared to object-based prototypes for spaces of the same dimension is the computational cost, since, when using clusters, more dissimilarities must be measured. In this case, for training and test objects, the dissimilarities with all the objects in the clusters must be computed in order to find the minimum, maximum and average dissimilarity. However, when compared to the approach using all objects as prototypes, the computational cost of the cluster-based approach is smaller because some clusters are discarded in the selection process and, thereby, less dissimilarity computations are made for training and test objects. Since the dissimilarity matrix is computed in advance before prototype selection is executed, the proposed approach as well as the standard prototype selection methods have limitations in case of very large datasets. This remains open for further research.

5 Conclusions

For the selection of prototypes not only the optimization method and criterion used are important, but also how the prototypes are devised is vital. We found that clusters may be useful to obtain low-dimensional GDSs in the case of datasets that present clusters. Our approach is useful for problems where the use of cluster-based prototypes make sense according to the data distribution. Note that our results hold for small and moderate training set sizes. When large training sets are available, they may compensate for bad mappings using objects as prototypes.

In general, we found that the minimum, average and subspace distances to clusters perform well in real-world datasets. However, there is no "best" approach among the cluster-based methods, it seems that the best option depends on specific data characteristics. Our intuition is that the minimum distance seems to be more meaningful for measuring distances with sets of objects with a shape such as the clusters. The cluster-based approaches improve the results of using DS of the same dimension but created by selected objects as well as DS using all the objects as prototypes (high-dimensional). Future works will be devoted to study the sensitivity to the choice of different clustering methods as well as the influence of numbers and sizes of the clusters.

References

1. 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)
2. Lozano, M., Sotoca, J.M., Sánchez, J.S., Pla, F., Pekalska, E., Duin, R.P.W.: Experimental study on prototype optimisation algorithms for prototype-based classification in vector spaces. Pattern Recogn. **39**(10) (2006) 1827–1838
3. Pekalska, E., Duin, R.P.W., Paclík, P.: Prototype selection for dissimilarity-based classifiers. Pattern Recogn. **39**(2) (2006) 189–208
4. Plasencia-Calaña, Y., Orozco-Alzate, M., García-Reyes, E., Duin, R.P.W.: Selecting feature lines in generalized dissimilarity representations for pattern recognition. Digit. Signal Process. **23**(3) (May 2013) 902–911
5. Riesen, K., Bunke, H.: Graph classification by means of Lipschitz embedding. Trans. Sys. Man Cyber. Part B **39**(6) (December 2009) 1472–1483
6. Plasencia-Calaña, Y., García-Reyes, E., Orozco-Alzate, M., Duin, R.P.W.: Prototype selection for dissimilarity representation by a genetic algorithm. In: Proceedings of the 20th International Conference on Pattern Recognition. ICPR '10, Washington, DC, USA, IEEE Computer Society (2010) 177–180
7. Frey, B.J.J., Dueck, D.: Clustering by passing messages between data points. Science **315** (January 2007) 972–976
8. Cox, T.F., Cox, M.: Multidimensional Scaling. 2 edn. Chapman and Hall/CRC (2000)
9. Bengio, Y., Paiement, J.F., Vincent, P., Delalleau, O., Roux, N.L., Ouimet, M.: Out-of-sample extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering. In: Advances in Neural Information Processing Systems, MIT Press (2003) 177–184
10. Baker, C.T.H.: The numerical treatment of integral equations. Clarendon Press, Oxford, New York. (1977)
11. Sigillito, V.G., Wing, S.P., Hutton, L.V., Baker, K.B.: Classification of radar returns from the ionosphere using neural networks. Johns Hopkins APL Technical Digest (1989) 262–266
12. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE Trans. Pattern Anal. Mach. Intell. **24**(4) (April 2002) 509–522
13. Sebastian, T.B., Klein, P.N., Kimia, B.B.: Recognition of shapes by editing their shock graphs. IEEE Trans. Pattern Anal. Mach. Intell. **26**(5) (May 2004) 550–571
14. Bunke, H., Buhler, U.: Applications of approximate string matching to 2D shape recognition. Pattern Recogn. **26**(12) (December 1993) 1797–1812
15. Breiman, L.: Bias, Variance, and Arcing Classifiers. Technical report, University of California, Berkeley (1996)