One-Sided Prototype Selection on Class Imbalanced Dissimilarity Matrices

M. Millán-Giraldo[1,2], V. García[2], and J.S. Sánchez[2]

[1] Intelligent Data Analysis Laboratory, University of Valencia
Av. Universitat s/n, 46100 Burjassot, Valencia (Spain)
[2] Institute of New Imaging Technologies
Department of Computer Languages and Systems, University Jaume I
Av. Sos Baynat s/n, 12071 Castelló de la Plana (Spain)

Abstract. In the dissimilarity representation paradigm, several prototype selection methods have been used to cope with the topic of how to select a small representation set for generating a low-dimensional dissimilarity space. In addition, these methods have also been used to reduce the size of the dissimilarity matrix. However, these approaches assume a relatively balanced class distribution, which is grossly violated in many real-life problems. Often, the ratios of prior probabilities between classes are extremely skewed. In this paper, we study the use of renowned prototype selection methods adapted to the case of learning from an imbalanced dissimilarity matrix. More specifically, we propose the use of these methods to under-sample the majority class in the dissimilarity space. The experimental results demonstrate that the one-sided selection strategy performs better than the classical prototype selection methods applied over all classes.

1 Introduction

In the traditional approach to Statistical Pattern Recognition, each object is represented in terms of n observable features or attributes, which can be regarded as a vector in an n-dimensional feature space. An alternative is the *dissimilarity space* proposed by Duin and Pekalska [1, 2]. To build the dissimilarity space, a representation set of r objects (or prototypes), $R = \{p_1, \ldots, p_r\}$, is needed. The dissimilarity representation allows to symbolize individual feature-patterns by pairwise dissimilarities computed between examples from the training set T and objects from the representation set R. Thus the dissimilarity vectors can be interpreted as numerical features and describe the relation between each object with the rest of objects [3].

Given a training set of m objects in the feature space, $T = \{x_1, \ldots, x_m\}$, the classifier is built using a dissimilarity matrix $D(T, R)$ that describes the proximities between the m training set objects and the r prototypes. The representation set can be chosen as the complete training set T, a set of constructed prototypes, a subset of T that covers all classes, or even an arbitrary set of labeled or unlabeled objects.

The dimensionality in the dissimilarity space is determined by the amount of prototypes in the set R. When $R = T$, the dissimilarity matrix $D(T, T)$ might impose high computational requirements on the classifier [4] and adversely affect the performance [5]. To face this drawback, several works have proposed to reduce the dimensionality of the dissimilarity space by selecting a small representation set from the training data [6]. Obviously, a pruned representation set will lead to reduce the distance matrix $D(T,T)$ to $D(T,R)$. In this context, prototype selection constitutes one of the most active research lines, which has primarily been addressed in two ways: (i) finding a small representation set capable of generating a low-dimensional dissimilarity space [4, 6, 7], and (ii) reducing the original dissimilarity matrix [8, 9].

Prototype selection methods have demonstrated to perform well in dissimilarity space classification when the classes are balanced. However, in many real-life problems the ratios of prior probabilities between classes can be extremely skewed. This situation is known as the class imbalance problem [10, 11]. A data set is said to be imbalanced when the examples from one class (the majority class) heavily outnumber the examples from the other (minority) class. This topic is particularly important in practical applications where it is costly to misclassify examples from the minority (or positive) class, such as medical diagnosis and monitoring, fraud/intrusion detection, credit risk and bankruptcy prediction, information retrieval and filtering tasks.

In this work, we explore the use of well-known prototype selection procedures (originally designed to be applied in the feature space) on the dissimilarity matrix $D(T,T)$ when this is imbalanced. Here, we propose to exploit these methods in a biased fashion, where only the majority class is pruned. In fact, this can be viewed as an under-sampling strategy, which is one of the common solutions to the class imbalance problem in feature spaces [12]. The experimental results show that this one-sided strategy performs significantly better than the standard application of prototype selection on both classes.

2 Prototype Selection Methods

Several prototype selection algorithms have been adapted and/or developed in order to select a small representation set R or to reduce the dissimilarity matrix $D(T,R)$. For example, Lozano et al. [13] employed prototype optimization methods often applied in vector spaces, such as editing and condensing, for constructing more general dissimilarity-based classifiers. Kim and Oommen [8] used the well-known condensed nearest neighbor rule [14] to reduce the original training set before computing the dissimilarity-based classifiers on the entire data. Other new methods have been evolved to be applied in the dissimilarity space, such as Kcentres, Edicon, ModeSeek, Featsel and a genetic algorithm [6, 9].

However, all these proposals do not consider the skewness in the class distribution. In this work, we concentrate on using four prototype selection methods, commonly applied to feature-based classification models, for the reduction of the Euclidean distance representation $D(T,T)$ (here called the original dissimilarity matrix) in domains with class imbalance. Two different families of prototype selection methods exist in the literature: editing and condensing. Editing removes erroneously labeled and atypical examples from the original set and "cleans" possible overlapping between classes, which usually leads to significant improvements in performance. Condensing, on the other hand, aims at selecting a sufficiently small subset of examples that yields approximately the same performance as using the whole training set.

The simplest procedure to pick up a small subset corresponds to random selection (RS). However, this may throw out potentially useful data. Paradoxically, it has empirically been shown to be an effective prototype selection method. Unlike the random approach, many other proposals are based upon a more intelligent selection strategy. For example, Wilson [15] introduced a popular editing algorithm (WE) that tries to remove noisy instances and/or border points. This algorithm discards training examples whose label does not agree with that of their majority k neighbors. Another early prototype selection method is the condensed nearest neighbor (CNN) proposed by Hart [14], which is focused on selecting a consistent subset from the training set but keeping or even improving the classification accuracy. Nevertheless, as this approach could retain noisy objects, the joint use of editing and condensing algorithms (e.g., WE+CNN) is commonly employed to select an appropriate reduced subset.

3 Performance Evaluation in Imbalanced Domains

Traditionally, standard performance metrics have been classification accuracy and/or error rates. For a two-class problem, these can be easily derived from a 2 $\times$ 2 confusion matrix as that given in Table 1.

Table 1. Confusion matrix for a two-class problem

	Predicted as positive	*Predicted as negative*
Positive class	True Positive (TP)	False Negative (FN)
Negative class	False Positive (FP)	True Negative (TN)

However, as pointed out by many authors [16, 17], the performance of a classification process over imbalanced data sets should not be expressed in terms of the plain accuracy and/or error rates because these measures are strongly biased towards the majority class. This has motivated to search for new performance evaluation metrics based upon simple indices, such as the true positive rate (TPr) and the true negative rate (TNr). The TPr (or TNr) is the percentage of positive (or negative) examples correctly classified.

One of the most widely-used evaluation methods in the context of imbalanced class distributions is the ROC curve. Here, we will utilize the area under the ROC curve (AUC), which is a quantitative representation of a ROC curve. For a binary problem, the AUC criterion defined by a single point on the ROC curve is also referred to as balanced accuracy [18]:

$$AUC_b = \frac{TPr + TNr}{2} \tag{1}$$

where $TPr = \frac{TP}{TP+FN}$ measures the percentage of positive examples that have been classified correctly, whereas $TNr = \frac{TN}{TN+FP}$ corresponds to the percentage of negative cases predicted as negative.

4 Experimental Setup

Eight real data sets were employed in the experiments. In order to force the class imbalance, all data sets were transformed into two-class problems by keeping one original class (the minority one) and joining the objects of the remaining classes. The fifth column in Table 2 indicates the original classes that were joined to shape the majority class.

Table 2. Data sets used in the experiments

Data Set	#Positive	#Negative	#Classes	Majority Class	Source
Breast	81	196	2	1	UCI[1]
Ecoli	35	301	8	1,2,3,5,6,7,8	UCI
German	300	700	2	1	UCI
Haberman	81	225	2	1	UCI
Laryngeal$_2$	53	639	2	1	Library[2]
Pima	268	500	2	1	UCI
Vehicle	212	634	4	2,3,4	UCI
Yeast	429	1055	10	1,3,4,5,6,7,8,9,10	UCI

[1] UCI Machine Learning Database Repository `http://archive.ics.uci.edu/ml/`

[2] Library `http://www.vision.uji.es/~sanchez/Databases/`

A stratified five-fold cross-validation method was adopted for the present experiments. For each fold, four parts were pooled as the training data T, and the remaining block was employed as an independent test set S. Ten repetitions were run for each trail. The results from classifying the test samples were averaged across the 50 runs. For each database, the whole training set ($R = T$) was used to compute the original dissimilarity matrix $D(T, T)$, with the Euclidean distance as a dissimilarity measure. This procedure was also applied to the test set, $D(S, T)$, to be represented in the dissimilarity space.

The four prototype selection methods described in Sect. 2 were utilized for the experiments: random selection (RS), condensed nearest neighbor (CNN), Wilson's editing (WE), and the combination of this with Hart's condensing (WE+CNN). All these methods were implemented following two different strategies: (i) *hard selection* over both existing classes, and (ii) *one-sided selection* only over the majority (negative) class. In this latter case, like occurs in typical under-sampling processes, we did not remove minority (positive) examples because they are too limited and important to be discarded. The Fisher and the nearest neighbor (1-NN) learning algorithms were used to each original dissimilarity matrix and also to matrices that were previously pruned by the different prototype selection methods.

Note that the RS procedure allows to control the number of prototypes to be chosen. Here, we extracted 50% out of each class for the hard selection strategy, and a number of negative examples equal to the size of the positive class $|P|$ for the one-sided selection strategy.

5 Results

In order to analyze the effect of the class imbalance on the performance of the prediction models, we generated different dissimilarity matrices, each one with an amount of positive examples, by randomly increasing the minority class size until reaching its original size. The number of objects in the majority class keeps constant for all dissimilarity matrices. Figure 1 shows the TPr and TNr for two illustrative examples of these data sets when using the Fisher classifier, where the x-axis represents the number of positive samples in the dissimilarity matrix. Note that both TPr and TNr have been plotted in a different scale in order to make these graphics clearer.

As expected, when the dissimilarity matrices are strongly imbalanced, the Fisher performance on the minority class is significantly worse than that on the majority class: the TNr is close to 0.90, but the TPr is below 0.40. As the size of the minority class increases, the TPr improves and the TNr lessens. It is worth noting, however, that the poor results of TPr remain even when all the positive examples are put into the dissimilarity matrix. In such an imbalance scenario, this effect demonstrates the need of using some strategy to generate more appropriate (balanced) dissimilarity matrices.

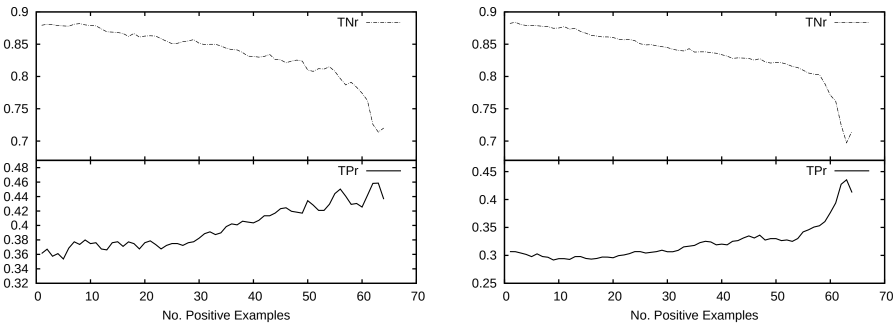

Fig. 1. Effect of the class imbalance on Fisher classifier performance for the Breast (left) and Haberman (right) databases

Tables 3 and 4 report the average AUC_b with the 1-NN and the Fisher classifiers respectively, when using the original dissimilarity matrix $D(T,T)$ and after pruning this by means of the prototype selection methods. The column "*One-S*" contains the results from applying the prototype selection procedures only over the majority class, whereas the column "*Hard*" refers to the results obtained when pruning both classes. For each data set, the best case has been highlighted in bold type. Average rankings of the Friedman statistic (distributed according to chi-square with 8 degrees of freedom) have also been included.

From the results in Tables 3 and 4, one can observe that both classifiers are affected by the class imbalance problem when they are trained with the original dissimilarity matrix, yielding relatively low AUC_b values. On the other hand, when employing the

Table 3. Average AUC_b results obtained with the 1-NN classifier

	Original	RS		CNN		WE		WE+CNN	
	matrix	One-S	Hard	One-S	Hard	One-S	Hard	One-S	Hard
Breast	0.575	0.574	0.562	0.620	0.598	**0.646**	0.587	0.626	0.610
Ecoli	0.791	**0.866**	0.774	0.706	0.668	0.838	0.776	0.815	0.676
German	0.535	0.551	0.539	**0.699**	0.693	0.681	0.587	0.692	0.623
Haberman	0.575	0.575	0.578	0.575	0.565	0.600	0.585	0.602	0.589
Laryngeal$_2$	0.775	0.846	0.746	0.830	0.793	**0.887**	0.741	0.849	0.738
Pima	0.624	0.632	0.625	0.687	0.674	**0.707**	0.686	0.694	0.690
Vehicle	0.579	0.606	0.580	0.699	0.673	0.679	0.572	**0.728**	0.588
Yeast	0.660	0.668	0.641	0.692	0.676	**0.719**	0.662	0.710	0.653
Average rankings	7.125	5.375	7.375	3.875	5.500	1.875	6.125	2.000	5.750

prototype selection methods over both classes (hard selection), the behavior varies from one data set to another: the AUC_b values are even worse than those achieved with the original dissimilarity matrix for some databases and better for others.

Table 4. Average AUC_b results obtained with the Fisher classifier

	Original	RS		CNN		WE		WE+CNN	
	matrix	One-S	Hard	One-S	Hard	One-S	Hard	One-S	Hard
Breast	**0.629**	0.625	0.609	0.567	0.560	0.596	0.530	0.583	0.552
Ecoli	0.736	0.857	0.738	0.794	0.773	0.860	0.760	**0.861**	0.737
German	0.678	**0.693**	0.658	0.535	0.530	0.570	0.553	0.566	0.552
Haberman	0.575	**0.604**	0.580	0.586	0.575	0.586	0.601	0.592	0.600
Laryngeal$_2$	0.872	**0.883**	0.833	0.792	0.766	0.815	0.694	0.834	0.704
Pima	0.**693**	0.687	0.676	0.612	0.604	0.657	0.673	0.649	0.671
Vehicle	0.660	**0.742**	0.647	0.584	0.580	0.606	0.573	0.611	0.574
Yeast	0.690	**0.710**	0.672	0.659	0.643	0.688	0.663	0.680	0.654
Average rankings	3.437	1.500	4.375	6.312	7.687	4.312	6.375	4.125	6.875

The results obtained with the application of prototype selection over the majority class (one-sided selection) show that all these techniques perform better, in terms of AUC_b, than the original dissimilarity matrix. It is also interesting to remark that this biased selection is significantly better than the classical approaches to prototype selection over both classes.

As a further confirmation of the findings with the AUC_b values, we have run a Wilcoxon signed-ranks test [19] between each pair of techniques. The upper diagonal half of Tables 5 and 6 summarizes this statistic for a significance level of 0.10 (10% or less chance), whereas the lower diagonal half corresponds to a significance level of 0.05. The symbol "•" indicates that the method in the row significantly improves the method of the column, and the symbol “○” means that the method in the column performs significantly better than the method of the row. The two bottom rows show how many times the algorithm of the column has been significantly better than the rest of procedures for $\alpha = 0.10$ and $\alpha = 0.05$.

Table 5. Summary of the Wilcoxon statistic for the prototype selection methods with the 1-NN classifier

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
(1) One-sided RS	-	●			○		○	
(2) Hard RS	○	-			○		○	
(3) One-sided CNN			-	●	○		○	●
(4) Hard CNN			○	-	○		○	
(5) One-sided WE	●	●	●	●	-	●		●
(6) Hard WE					○	-	○	
(7) One-sided WE+CNN		●	●	●		●	-	●
(8) Hard WE+CNN					○		○	-
$\alpha = 0.10$	1	0	2	0	6	0	6	0
$\alpha = 0.05$	1	0	1	0	6	0	5	0

It is worth pointing out that, as can be observed in Tables 5 and 6, the one-sided selection has been significantly better than the hard selection strategy for all the prototype selection algorithms (for $\alpha = 0.10$ and $\alpha = 0.05$), both with the 1-NN classifier and the Fisher classifier. This allows to assert that such a biased selection of prototypes for the construction of a more balanced dissimilarity matrix (with all the positive examples and only a subset of negative examples) can be deemed as an appropriate solution to the class imbalance problem in dissimilarity spaces.

In the case of the 1-NN classifier, it seems that the best prototype selection method corresponds to Wilson’s editing, whose one-sided version has performed significantly better than other six algorithms at both significance levels. The WE+CNN procedure presents a very similar behavior, being significantly better than other five algorithms at a significance level of 0.05. Clearly, the random selection and Hart’s condensing methods have achieved the worst results when statistically compared in terms of AUC_b.

Paradoxically, for the Fisher classifier, Table 6 shows that the one-sided random selection constitutes the best procedure, with a performance significantly better than any other algorithm. Not too far from the best alternative, one can see that the Wilson’s editing with one-sided selection has been significantly better than other four strategies.

6 Conclusions

Prototype selection methods have been widely used in the dissimilarity-based approach for the selection of a small representation set (from the whole training set) and/or the reduction of the original dissimilarity matrix. When the data set and/or the dissimilarity

Table 6. Summary of the Wilcoxon statistic for the prototype selection methods with the Fisher classifier

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
(1) One-sided RS	-	●	●	●	●	●	●	●
(2) Hard RS	○	-		●				●
(3) One-sided CNN	○		-	●	○		○	
(4) Hard CNN	○	○	○	-	○		○	
(5) One-sided WE	○		●	●	-	●		●
(6) Hard WE	○				○	-	○	
(7) One-sided WE+CNN	○		●	●			-	●
(8) Hard WE+CNN	○				○			-
$\alpha = 0.10$	7	2	1	0	4	0	4	0
$\alpha = 0.05$	7	1	1	0	4	0	2	0

matrix are imbalanced, however, the selection process could produce reduced data sets and/or dissimilarity matrics that do not accurately represent the true class distribution, what may lead to an increase in the class skewness.

In this paper, we have carried out some experiments using four renowned prototype selection algorithms for under-sampling the original dissimilarity matrix in domains with class imbalance. The empirical results suggest that the application of these techniques to both classes produces poor performance on the minority class. On the contrary, the strategy based upon the biased selection on the majority class significantly increases the prediction rate on the positive class and the value of average AUC_b, being statistically demonstrated by means of a Wilcoxon signed-ranks test.

Acknowledgment

This work has partially been supported by the Spanish Ministry of Education and Science under grants CSD2007–00018, AYA2008–05965–0596 and TIN2009–14205, the Fundació Caixa Castelló-Bancaixa under grant P1–1B2009–04, and the Generalitat Valenciana under grant PROMETEO/2010/028.

References

1. Duin, R.P.W., Pękalska, E.: The dissimilarity space: Bridging structural and statistical pattern recognition. Pattern Recognition Letters **33**(7) (2012) 826–832
2. Pekalska, E., Duin, R.P.W.: The Dissimilarity Representation for Pattern Recognition: Foundations and Applications. World Scientific (2005)
3. Pekalska, E., Duin, R.P.W.: Dissimilarity representations allow for building good classifiers. Pattern Recognition Letters **23**(8) (2002) 943–956
4. Kim, S.W.: An empirical evaluation on dimensionality reduction schemes for dissimilarity-based classifications. Pattern Recognition Letters **32**(6) (2011) 816–823

5. Duin, R.P.W., Pękalska, E.: The dissimilarity representation for structural pattern recognition. In: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. Springer-Verlag (2011) 1–24
6. Pekalska, E., Duin, R.P.W., Paclík, P.: Prototype selection for dissimilarity-based classifiers. Pattern Recognition **39**(2) (2006) 189–208
7. Plasencia-Calaña, Y., García-Reyes, E., Duin, R.P.W.: Prototype selection methods for dissimilarity space classification. Technical report, Advanced Technologies Application Center CENATAV
8. Kim, S.W., Oommen, B.J.: On using prototype reduction schemes to optimize dissimilarity-based classification. Pattern Recognition **40**(11) (2007) 2946–2957
9. Plasencia-Calaña, Y., García-Reyes, E., Orozco-Alzate, M., Duin, R.P.W.: Prototype selection for dissimilarity representation by a genetic algorithm. In: Proc. 20th International Conference on Pattern Recognition. (2010) 177–180
10. Chawla, N., Japkowicz, N., Kotcz, A.: Editorial: Special issue on learning from imbalanced data sets. SIGKDD Explorations **6**(1) (2004) 1–6
11. Sun, Y., Wong, A., Kamel, M.S.: Classification of imbalanced data: A review. International Journal of Pattern Recognition and Artificial Intelligence **23**(4) (2009) 687–719
12. Batista, G.E.A.P.A., Prati, R.C., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data. SIGKDD Explorations **6**(1) (2004) 20–29
13. 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 Recognition **39** (2006) 1827–1838
14. Hart, P.E.: The condensed nearest neighbor rule. IEEE Trans. on Information Theory **14** (1968) 515–516
15. Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data. IEEE Trans. on Systems, Man and Cybernetics **2**(3) (1972) 408–421
16. Daskalaki, S., Kopanas, I., Avouris, N.: Evaluation of classifiers for an uneven class distribution problem. Applied Artificial Intelligence **20**(5) (2006) 381–417
17. Provost, F., Fawcett, T.: Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. In: Proc. 3rd International Conference on Knowledge Discovery and Data Mining. (1997) 43–48
18. Sokolova, M., Japkowicz, N., Szpakowicz, S.: Beyond accuracy, F-score and ROC: A family of discriminant measures for performance evaluation. In: Proc. 19th ACS Australian Joint Conference on Artificial Intelligence. (2006) 1015–1021
19. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research **7**(1) (2006) 1–30