Mixed Data Balancing through Compact Sets Based Instance Selection

Yenny Villuendas-Rey[1] and María Matilde García-Lorenzo[2]

[1] Department of Computer Science, University of Ciego de Ávila, Carr. A Morón km 9 ½, Cuba
[2] Department of Computer Science, Universidad Central Marta Abreu of Las Villas,
Carr. A Camajuaní, km 5 ½, Cuba
yenny@informatica.unica.cu, mmgarcia@uclv.edu.cu

Abstract. Learning in datasets that suffer from imbalanced class distribution is an important problem in Pattern Recognition. This paper introduces a novel algorithm for data balancing, based on compact set clustering of the majority class. The proposed algorithm is able to deal with mixed, as well as incomplete data, and with arbitrarily dissimilarity functions. Numerical experiments over repository databases show the high quality performance of the method proposed in this paper according to area under the ROC curve and imbalance ratio.

Keywords: imbalanced data, mixed data, supervised classification.

1 Introduction

The training dataset plays a key role for supervised classification. Training data allows building classifiers able to estimate the label or class of a new unseeing instance. Several researchers have pointed out that if the dataset has an approximately equal amount of instances for every class, the classifier can produce predictions that are more accurate [1]. However, in several real-world applications, it is not possible to obtain a training set with classes equally distributed. The class imbalance problem occurs when one or several classes (the majority classes) vastly outnumber the other classes (the minority classes), which are usually the most important classes and often with the highest misclassification costs.This problem is known as the problem of learning in imbalanced scenarios.

Learning in imbalanced scenarios poses challenges for supervised classifiers, such as Nearest Neighbor (NN). Several researchers have addressed the impact of data imbalance in NN performance [2, 3]. The problem of class imbalance has been addressed by numerous approaches at both algorithmic and data levels. At algorithmic level, the methods usually modify the learning algorithm to favor the detection of the minority class, while the solutions at data level obtain an approximately equally distributed data set, by means of re-sampling, either by oversampling the minority class [4] or undersampling the majority class [5-7]. Oversampling techniques create artificial objects of the minority class, and increase the computational cost of the

J. Ruiz-Shulcloper and G. Sanniti di Baja (Eds.): CIARP 2013, Part I, LNCS 8258, pp. 254–261, 2013.

learning algorithms, and the storage cost of the dataset, while undersampling techniques preserve minority class and obtains a small representation of majority class.

This paper proposes a novel algorithm for undersampling. The algorithm is based on Compact Sets (CS) structuralizations, and is able to deal with mixed and incomplete data. The use of CS based clustering allows selecting a highly representative set of the majority class, preserving the objects of minority class. The thorough experimental study carried out shows the significant performance gains of the proposed approach when compared to other state-of-the-art algorithms.

2 Compact Sets Based Data Balancing by Under-Sampling

One of the greatest challenges in undersampling techniques is to obtain a good representation of the majority class. Instead of using a classical prototype selection strategy, this paper introduces the idea of structuralize the majority class by means of compact sets, and then obtain the desired number of prototypes. Compact sets have been used successfully for prototype selection in mixed and incomplete data, and also for clustering [8, 9]. A compact set is a connected component of a Maximum Similarity Graph. A Maximum Similarity Graph is a directed graph, such as it connects each object with its most similar neighbors [10]. Formally, let be $G = (X, \theta)$ a MSG for a set of objects X, with arcs θ. Two objects $x_i, x_j \in X$ form an arc $(x_i, x_j) \in \theta$ if $\max_{x \in X}\{sim(x_i, x)\} = sim(x_i, x_j)$, where $sim(x_i, x_j)$ is a similarity function, usually $sim(x_i, x_j) = 1 - \Delta(x_i, x_j)$ and $\Delta(x_i, x_j)$ is a dissimilarity function. Each connected component of such graph is called a compact set. Compact sets are formed by highly similar instances, and allow structuralizing datasets. Formally, a subset $N \neq \emptyset$ of X is a β_0 compact set if and only if [10]:

$$a)\forall x_j \in X \left[x_i \in N \wedge \left(\begin{array}{c} \max\limits_{\substack{x_k \in X \\ x_k \neq x_i}} \{\mathrm{sim}(\mathrm{x_i}, \mathrm{x_k})\} = \mathrm{sim}(\mathrm{x_i}, \mathrm{x_j}) \geq \beta_0 \\ \vee \max\limits_{\substack{x_k \in X \\ x_k \neq x_i}} \{\mathrm{sim}(\mathrm{x_k}, \mathrm{x_i})\} = \mathrm{sim}(\mathrm{x_j}, \mathrm{x_i}) \geq \beta_0 \end{array} \right) \right] \Rightarrow x_j \in N$$

$$b)\forall x_i, x_j \in N, \exists x_{i_1}, \cdots, x_{i_q} \in N \left[\begin{array}{c} x_i = x_{i_1} \wedge x_j = x_{i_q} \wedge \forall \mathrm{p}\,\{1, \cdots, \mathrm{q}-1\} \\ \left[\begin{array}{c} \max\limits_{\substack{x_t \in X \\ x_t \neq x_{i_p}}} \left\{\mathrm{sim}\left(x_{i_p}, \mathrm{x_t}\right)\right\} = \mathrm{sim}\left(x_{i_p}, x_{i_{p+1}}\right) \geq \beta_0 \\ \vee \max\limits_{\substack{x_t \in X \\ x_t \neq x_{i_p}}} \left\{\mathrm{sim}\left(x_{i_{p+1}}, \mathrm{x_t}\right)\right\} = \mathrm{sim}\left(x_{i_{p+1}}, x_{i_p}\right) \geq \beta_0 \end{array} \right] \end{array} \right]$$

$c)$ Every isolated object is a compact set, degenerated.

The proposed algorithm, called CDB (Compact set based Data Balancing) starts by dividing the dataset into majority and minority classes (Figure 1 and 2). Then, it computes the compact sets of the majority class, and each is considered as a group. Next, for each group, it finds the more similar objects with respect to every other object in the group (holotype) to represent the group.

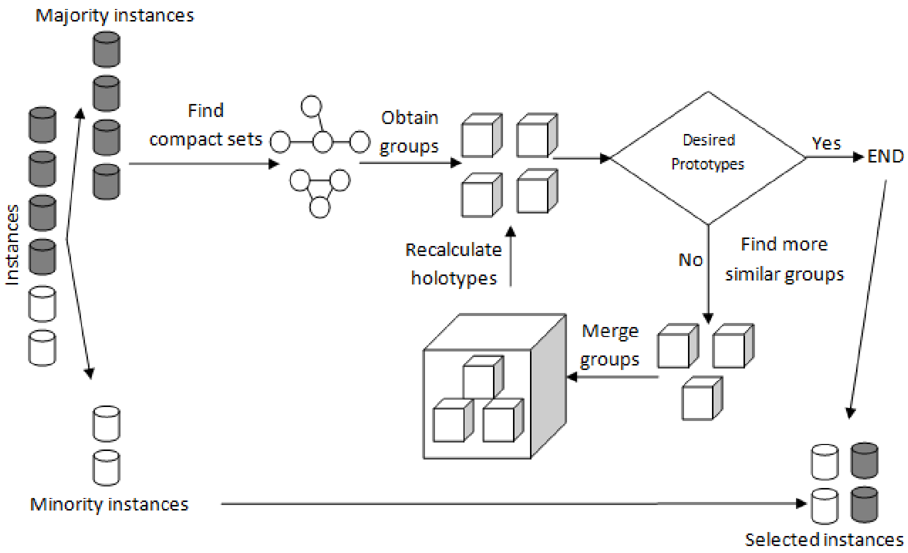

Fig. 1. Compact Set based Data Balancing algorithm (CDB).

The algorithm finds the most similar groups, and merges them, until the desired number of prototypes (groups) is reached. This proposal directly merges all possible groups which have less dissimilarity in a single step. This makes faster the merging process, and avoids order dependence.

```
Compact set based Data Balancing (CDB) algorithm
Inputs: I: set of instances, Δ: similarity function
Output: P: prototype set
1.  C = ϕ, P = ϕ
2.  Move the objects of I belonging to majority class to a set M, and add to P the remaining (minority) objects of I.
3.  Create a maximum similarity graph of the objects in the set M using $\beta_0=0$ and $\Delta$
4.  Add to C each connected component of the graph created at step 3.
    4.1. Select as cluster centre (holotype) the object that maximizes the overall similarity with respect to every object in the cluster
5.  Merge all more similar groups, using as cluster similarity the similarity between cluster centers.
    5.1. Recalculate cluster holotypes.
6.  Repeat step 5, until $|C| \leq |P|$, that is, until the amount of clusters is less or equal to the amount of objects of minority class.
7.  Add to P the holotypes of C
8.  Return P
```

Fig. 2. Pseudocode of Compact set based Data Balancing (CBD) algorithm

The new compact set based algorithm differs from previously reported algorithms in the following: It clearly defines the amount of prototypes to select from majority class. It also deals with mixed and incomplete data, by using compact sets and a hierarchical approach that selects representative instances instead of constructing artificial ones. In addition, it uses the similarity between holotypes as intergroup similarity, avoiding additional instance similarity computation and it merges at each stage all groups selected as more similar, avoiding order-dependence.

3 Experimental Results

To compare the performance of the proposed algorithms, there were used 44 databases from the KEEL dataset repository [11].

Table 1. Databases used in the experiments

Databases	Att.	Instances	IR	Databases	Att.	Instances	IR
abalone9-18	8	2934	17	glass04-5	9	368	9
abalone19	8	16706	130	glass06-5	9	432	11
cleveland0-4	13	708	13	glass2-5	9	856	12
ecoli01-235	7	976	9	glass4-5	9	856	15
ecoli01-55	6	960	11	glass5-5	9	856	23
ecoli0137-26	7	1124	39	led7digit1	7	1772	11
ecoli0146-5	6	1120	13	page-blocks13-4	10	1888	16
ecoli0147-2356	7	1344	11	shuttlec0-4	9	7316	14
ecoli0147-56	6	1328	12	shuttlec2-4	8	516	21
ecoli0234-5	7	808	9	vowel0	13	3952	10
ecoli0267-35	7	896	9	yeast0256-3789	8	4016	9
ecoli034-5	7	800	9	yeast02579-368	8	4016	9
ecoli0346-5	7	820	9	yeast0359-78	8	2024	9
ecoli0347-56	7	1028	9	yeast05679-4	8	2112	9
ecoli046-5	6	812	9	yeast1-7	3	1836	14
ecoli067-35	7	888	9	yeast1289-7	8	3788	31
ecoli067-5	6	880	10	yeast1458-7	8	2772	22
ecoli4-5	7	1344	16	yeast2-4	8	3056	14
glass0146-2	9	820	11	yeast2-8	8	1928	23
glass015-2	9	688	9	yeast4	8	5936	28
glass016-2	9	768	10	yeast5	8	5936	33
glass016-5	9	736	19	yeast6	8	5936	41

These databases were modified from its original version, to obtain highly imbalanced data sets, having only one minority and one majority class [11]. The name of the datasets represents the index of minority and majority classes. Table 1 shows the characteristics of the selected databases. The second and third columns show the amount of attributes (Att.) and instances of the dataset, and the fourth, the Imbalance Ratio of each database. Imbalance Ratio (IR) is defined as the ratio between the instances count of majority class, with respect to the count of instances of minority class.

For numerical comparison, there were selected the HEOM (equation 1) dissimilarity function, proposed by Wilson and Martínez [12], which is able to deal with mixed and incomplete data.

$$HEOM(x,y) = \sqrt{\sum_{a=1}^{m} d_a(x_a,y_a)}\,, d_a = \begin{cases} 1 \\ overlap(x_a,y_a), \\ diff(x_a,y_a) \end{cases} \qquad (1)$$

$$overlap(p,q) = \begin{cases} 0 & if\ p = q \\ 1\ in\ other\ case \end{cases}, diff(p,q) = |p-q|/max_a - min_a$$

In addition, the SEC [5], NCL [6], and GGE [7] algorithms were selected for comparison purposes, because they are among best undersampling algorithms for mixed data balancing. All algorithms were implemented in C# language, and the experiments were carried out in a laptop with 3.0GB of RAM and Intel Core i5 processor with 2.67HZ.

To compare the performance of the algorithms, it was used the area under the ROC curve (AUC).The Area under the ROC curve is another quality measure widely used to evaluate classifiers in problems with unequal costs, such as imbalanced problems. In [13] are shown some of the advantages of using the AUC measure over other quality measures, such as classifier error. To compute the AUC (equation 2) for a discrete classifier, a simple method is proposed in [14], based on a confusion matrix (table 2). It was also computed the Imbalance Ratio (IR) for every algorithm, in order to determine their effectiveness in balancing the datasets.

$$AUC = \frac{TPR+TNR}{2} \text{ where } TPR = (tp)/(tp+fp) \text{ and } TNR = (tn)/(tn+fn) \qquad (2)$$

Table 2. Confusion matrix for two class problems

	Positive Prediction	Negative Prediction
Positive Class	True Positive (tp)	False Negative (fn)
Negative Class	False Positive (fp)	True Negative (tn)

Table 3 and 5 show the results according to AUC and Imbalance Ratio, respectively, in the testing phase. As shown in table 3, the proposed CDB algorithm obtains the highest area under the ROC curve in 28 databases, 16 of them above 0.9. These results show the high performance of the proposed method.

Table 3. AUC of the algorithms. In bold best results

Databases	**NCL**	**SEC**	**GGE**	**CDB**	**Databases**	**NCL**	**SEC**	**GGE**	**CDB**
abalone9-18	0.58	0.60	0.61	**0.83**	glass04-5	0.73	**0.81**	0.78	0.67
abalone19	0.50	0.50	0.50	**0.78**	glass06-5	0.67	0.70	0.65	**0.88**
cleveland0-4	0.74	0.83	0.73	**0.93**	glass2-5	0.51	0.50	0.46	**0.90**
ecoli01-235	0.76	0.73	0.77	**0.90**	glass4-5	0.48	0.53	0.51	**0.68**
ecoli01-55	0.87	0.86	**0.90**	**0.90**	glass5-5	0.69	0.71	0.67	**0.96**
ecoli0137-26	0.50	0.50	0.50	**0.88**	led7digit1	0.64	0.85	0.66	**1.00**
ecoli0146-5	0.87	0.89	0.90	**0.91**	page-blocks13-4	0.92	**0.96**	**0.96**	0.95
ecoli0147-2356	0.79	0.78	0.81	**0.90**	shuttlec0-4	**1.00**	**1.00**	**1.00**	0.99
ecoli0147-56	0.87	0.88	**0.91**	0.86	shuttlec2-4	0.95	**1.00**	0.95	0.75
ecoli0234-5	0.82	0.84	0.50	**0.93**	vowel0	0.94	**0.95**	0.94	0.86
ecoli0267-35	0.78	0.83	0.79	**0.86**	yeast0256-3789	0.70	**0.73**	0.72	0.72
ecoli034-5	0.82	0.82	0.57	**0.88**	yeast02579-368	0.81	0.82	**0.85**	0.79
ecoli0346-5	0.80	0.82	0.81	**0.91**	yeast0359-78	**0.67**	**0.67**	**0.67**	0.62
ecoli0347-56	0.50	0.50	0.50	**0.86**	yeast05679-4	0.67	0.69	**0.73**	0.64
ecoli046-5	0.87	**0.89**	0.88	0.86	yeast1-7	0.48	0.48	**0.63**	0.58
ecoli067-35	0.83	0.83	0.85	**0.93**	yeast1289-7	0.57	0.59	0.57	**0.90**
ecoli067-5	0.82	0.84	**0.85**	0.72	yeast1458-7	0.60	0.63	0.62	**0.77**
ecoli4-5	0.70	**0.74**	0.69	0.56	yeast2-4	0.50	0.73	0.73	**0.78**
glass0146-2	0.55	0.54	0.56	**0.66**	yeast2-8	0.72	0.71	0.70	**0.96**
glass015-2	0.53	0.52	0.50	**0.92**	yeast4	0.59	0.64	0.68	**0.80**
glass016-2	0.54	0.51	0.52	**0.94**	yeast5	0.73	0.79	**0.80**	0.66
glass016-5	0.64	0.66	0.67	**0.98**	yeast6	0.72	0.72	**0.75**	0.69

However, to determine the existence or not of significant differences in algorithm´s performance it was used the Wilcoxon test [15]. It was set as null hypothesis no difference in performance between the proposed method and every other algorithm, and as alternative hypothesis that CDB had better performance. It was set a significant value of 0.05, for a 95% of confidence. Table 4 summarizes the results of the Wilcoxon test, according to area under the ROC curve. The Wilcoxon test concludes the proposed method has significantly better performance than the other compared methods, according to the area under the ROC curve.

Table 4. Wilcoxon test comparing area under the ROC curve

CDB vs.	**NCL**	**SEC**	**GGE**
wins – looses – ties	31-13-0	29-15-0	26-17-1
probability	**0.000**	**0.001**	**0.001**

Table 5. Imbalance Ratio of the algorithms. In bold best results (near to 1).

Databases	NCL	SEC	GGE	CDB	Databases	NCL	SEC	GGE	CDB
abalone9-18	16.36	7.24	13.27	**1.00**	glass04-5	8.64	4.17	7.42	**1.00**
abalone19	129.3	31.7	126.1	**1.00**	glass06-5	10.50	5.69	8.44	**1.00**
cleveland0-4	12.29	3.17	11.69	**1.00**	glass2-5	11.35	7.26	7.93	**1.00**
ecoli01-235	8.95	2.72	8.50	**1.00**	glass4-5	15.06	5.85	12.85	**1.14**
ecoli01-55	10.76	2.81	10.66	**1.00**	glass5-5	22.22	9.17	21.08	**1.00**
ecoli0137-26	38.61	11.0	37.14	**1.00**	led7digit1	10.87	1.73	10.61	**1.00**
ecoli0146-5	12.76	3.26	12.69	**1.00**	page-blocks13-4	15.73	3.51	15.63	**1.00**
ecoli0147-2356	10.35	3.53	9.72	**1.00**	shuttlec0-4	13.85	0.04	13.87	**1.00**
ecoli0147-56	12.03	3.39	11.60	**1.00**	shuttlec2-4	20.00	**1.00**	20.50	**1.00**
ecoli0234-5	8.90	2.36	8.63	**1.00**	vowel0	9.92	1.81	9.68	**1.00**
ecoli0267-35	8.98	3.00	8.40	**1.00**	yeast0256-3789	9.04	4.76	7.35	**1.00**
ecoli034-5	8.78	2.29	8.53	**1.00**	yeast02579-368	9.05	4.11	8.14	**1.00**
ecoli0346-5	9.04	2.49	8.79	**1.00**	yeast0359-78	8.93	4.86	6.94	**1.00**
ecoli0347-56	9.03	2.97	8.53	**1.00**	yeast05679-4	9.16	4.51	7.64	**1.00**
ecoli046-5	8.94	2.21	8.76	**1.00**	yeast1-7	14.13	5.48	12.32	**1.00**
ecoli067-35	8.89	2.91	8.35	**1.00**	yeast1289-7	30.38	15.00	27.73	**1.00**
ecoli067-5	9.79	3.16	9.18	**1.00**	yeast1458-7	21.90	11.42	18.88	**1.00**
ecoli4-5	15.56	4.60	14.88	**1.00**	yeast2-4	8.93	3.99	8.26	**1.00**
glass0146-2	10.79	6.69	7.94	**1.00**	yeast2-8	22.91	9.70	21.35	**1.00**
glass015-2	8.88	5.57	5.74	**1.00**	yeast4	27.90	11.27	25.98	**1.00**
glass016-2	10.01	6.28	7.19	**1.00**	yeast5	32.57	7.80	31.37	**1.00**
glass016-5	18.86	8.64	17.75	**1.00**	yeast6	41.09	14.93	38.79	**1.00**

As shown, CDB obtains a perfectly balanced dataset, in 43 of 44 databases, with only 1.14 of Imbalance Ratio in the glass4-5 database. These results confirm the proposed approach is able to obtain an adequate balance of data, without losing representative objects of majority class.

4 Conclusions

Prototype selection for data balancing is an important preprocessing step for learning in imbalance scenarios. In this paper, a novel method is introduced, using Compact Sets for hierarchical clustering of majority class. The method keeps minority objects, and selects representative objects of majority class, from compact sets structuralizations. The method is also able to deal with databases containing objects described by features no exclusively numeric or categorical. Experimental results carried out over several repository data show the high performance of the proposed method, which obtains a perfectly balanced datasets with high area under the ROC curve.

References

1. Weiss, G.M.: Learning with rare cases and small disjuncts. In: Proceedings of the International Conference on Machine Learning, ICML 2003, pp. 558–565 (2003)
2. Hand, D.J., Vinciotti, V.: Choosing k for two-class nearest neighbor classifiers with imbalanced classes. Pattern Recognition Letters 24, 1555–1562 (2003)
3. Zhang, J., Mani, I.: kNN approach to unbalanced data distribution: a case study involving information extraction. In: Proceedings of Workshop on Learning from Imbalanced Datasets (2003)
4. Moreno, J., Rodriguez, D., Sicilia, M.A., Riquelme, J.C., Ruiz, R.: SMOTE-I: improvement of SMOTE algorithm for minority classes balancing. In: Proceedings of Workshops of Software Engineering and Databases 3 (2009) (in Spanish)
5. García, V.: Distributions of non-balanced classes: metrics, complexity analysis and learning algorithms. PhD Dissertation Thesis, Department of Languages and Computer Systems, University Jaume I, Spain (2010)
6. Laurikkala, J.: Instance-based data reduction for improved identification of difficult small classes. Intelligent Data Analysis 6, 311–322 (2002)
7. Alejo, R., Valdovinos, R.M., García, V., Pacheco-Sanchez, J.H.: A hybrid method to face class overlap and class imbalance on neural networks and multi-class scenarios. Pattern Recognition Letters 34, 380–388 (2013)
8. García-Borroto, M., Ruiz-Shulcloper, J.: Selecting prototypes in Mixed and Incomplete data. In: Sanfeliu, A., Cortés, M.L. (eds.) CIARP 2005. LNCS, vol. 3773, pp. 450–459. Springer, Heidelberg (2005)
9. Villuendas-Rey, Y., Rey-Benguría, C., Caballero-Mota, Y., García-Lorenzo, M.M.: Nearest prototype classification of special school families based on hierarchical compact sets clustering. In: Pavón, J., Duque-Méndez, N.D., Fuentes-Fernández, R. (eds.) IBERAMIA 2012. LNCS, vol. 7637, pp. 662–671. Springer, Heidelberg (2012)
10. Ruiz-Shulcloper, J., Abidi, M.A.: Logical combinatorial Pattern Recognition: A review. In: Pandalai, S.G. (ed.) Recent Research Developments in Pattern Recognition. Transword Research Networks, pp. 133–176 (2002)
11. Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F.: KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. Journal of Multiple-Valued Logic and Soft Computing 17, 255–287 (2011)
12. Wilson, R.D., Martinez, T.R.: Improved heterogeneous distance functions. Journal of Artificial Intelligence Research 6, 1–34 (1997)
13. Bradley, A.: The use of Area under the ROC curve in the evaluation of Machine Learning Algorithms. Pattern Recognition 30, 1145–1159 (1997)
14. Sokolova, M., Japkowicz, N., Szpakowicz, S.: Beyond Accuracy, F-Score and ROC: a family of Discriminant measures for Performance evaluations. In: Proceedings of the Australian Conference on Artificial Intelligence, pp. 1015–1021 (2006)
15. Demsar, J.: Statistical comparison of classifiers over multiple datasets. Journal of Machine Learning Research 7, 1–30 (2006)