Learning from Imbalanced Data in Presence of Noisy and Borderline Examples

Krystyna Napierała, Jerzy Stefanowski, and Szymon Wilk

Institute of Computing Science, Poznań University of Technology,
ul. Piotrowo 2, 60–965 Poznań, Poland
{krystyna.napierala,jerzy.stefanowski,szymon.wilk}@cs.put.poznan.pl

Abstract. In this paper we studied re-sampling methods for learning classifiers from imbalanced data. We carried out a series of experiments on artificial data sets to explore the impact of noisy and borderline examples from the minority class on the classifier performance. Results showed that if data was sufficiently disturbed by these factors, then the focused re-sampling methods – NCR and our SPIDER2 – strongly outperformed the oversampling methods. They were also better for real-life data, where PCA visualizations suggested possible existence of noisy examples and large overlapping ares between classes.

1 Introduction

In some real-life problems, the distribution of examples in classes is highly imbalanced, which means that one of the classes (further called a *minority class*) includes much smaller number of examples than the other majority classes [1,3]. Class imbalance constitutes a difficulty for most learning algorithms, which assume even class distribution and are biased toward learning and recognition of the majority classes. As a result, minority examples tend to be misclassified.

This problem has been intensively researched in the last decade and several methods have been proposed (see [1,3] for a review). They are usually divided into solutions on the data level and the algorithmic level. Solutions on the data level are classifier-independent and consist in transforming an original data distribution to change the balance between classes, e.g., by re-sampling techniques. Solutions on the algorithmic level involve modification of either learning or classification strategies. Some researchers also generalize ensembles or transform the imbalance problem to cost sensitive learning [3].

In this paper we are interested in focused re-sampling techniques, which modify the class distribution taking into account local characteristics of examples. Inspired by [6] we distinguish between safe, borderline and noisy examples. *Borderline examples* are located in the area surrounding class boundaries, where the minority and majority classes overlap. *Safe examples* are placed in relatively homogeneous areas with respect to the class label. Finally, by *noisy examples* we understand individuals from one class occurring in safe areas of the other class. We claim that the distribution of borderline and noisy examples causes difficulties for learning algorithms, thus we focus our interest on careful processing of these examples.

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 158–167, 2010.

Our study is related to earlier works of Stefanowski and Wilk on selective pre-processing with the SPIDER (Selective Preprocessing of Imbalanced Data) method [8,9]. This method employs the Edited Nearest Neighbor Rule (ENNR) to identify the local characteristic of examples, and then it combines removing the majority class objects that may result in misclassifying objects from the minority class with local over-sampling of these objects from the minority class that are "overwhelmed" by surrounding objects from the majority classes. Experiments showed that this method improved the recognition of the minority class and was competitive to the most related approaches SMOTE and NCR [9]. The observed improvements varied over different imbalanced data sets, therefore, in this study we have decided to explore conditions, where the SPIDER method could be more efficient than simpler re-sampling methods. To achieve this goal we have planned controlled experiments with special artificial data sets.

According to related works many experiments were conducted on real-life data sets (e.g., coming from UCI). The most well known studies with artificial data are the works of Japkowicz [4,5], who showed that simple class imbalance ratio was not the main difficulty. The degradation of performance was also related to other factors, mainly to small disjuncts, i.e., the minority class being decomposed into many sub-clusters with very few examples. Other researchers also explored the effect of overlapping between imbalanced classes – more recent experiments on artificial data with different degrees of overlapping also showed that overlapping was more important than the overall imbalance ratio [2].

Following these motivations we prepare our artificial data sets to analyze the influence of the presence and frequency of the noisy and borderline examples. We also plan to explore the effect of the decomposition of this class into smaller sub-clusters and the role of changing decision boundary between classes from linear to non-linear shapes. The main aim of our study is to examine which of these factors were critical for the performance of the methods dealing with imbalanced data. In the experiments we compare the performance of the SPIDER method and the most related focused re-sampling NCR method with the oversampling methods suitable to handle class decomposition [5] and the basic versions of tree- or rule-based classifiers.

2 Focused Re-sampling Methods

Here we discuss only these re-sampling methods that are used in our experiments. The simplest oversampling randomly replicates examples from the minority class until the balance with cardinality of the majority classes is obtained. Japkowicz proposed an advanced oversampling method (*cluster oversampling*) that takes into account not only *between-class imbalance* but also *within-class imbalance*, where classes are additionally decomposed into smaller sub-clusters [5]. First, random oversampling is applied to individual clusters of the majority classes so that all the sub-clusters are of the same size. Then, minority class clusters are processed in the same way until class distribution becomes balanced. This approach was successfully verified in experiments with decomposed classes [5]. In

[6] one side sampling was proposed, where noisy and borderline examples from the majority class are removed and the minority class remains unchanged. A similar idea was employed in the NCR (Neighborhood Cleaning Rule) method [7]. NCR applies ENNR to identify and remove noisy and borderline examples from the majority classes. NCR demonstrates a few undesirable properties (e.g., improvement of sensitivity at the cost of specificity) and their critical analysis has become a starting point for the family of the SPIDER methods. Following [6], they rely on the local characteristics of examples discovered by analyzing their k-nearest neighbors. SPIDER2 is presented in Alg. 1. To simplify the notation we do not distinguish between noisy and borderline examples and refer to them simply as not-safe.

Algorithm 1. SPIDER2

Input : DS – data set; c_{min} – the minority class; k – the number of nearest neighbors; *relabel* – relabeling option (yes, no); *ampl* – amplification option (no, weak, strong)
Output: preprocessed DS

```
1  c_maj := an artificial class combining all classes except c_min
2  foreach x ∈ class(DS, c_maj) do
3  |  if correct(DS, x, k) then flag x as safe
4  |  else flag x as not-safe
5  RS := flagged(DS, c_maj, not-safe)
6  if relabel then
7  |  foreach y ∈ RS do
8  |  |  change classification of y to c_min
9  |  |  SR := SR \ {y}
10 else DS := DS \ RS
11 foreach x ∈ class(DS, c_min) do
12 |  if correct(DS, x, k) then flag x as safe
13 |  else flag x as not-safe
14 if ampl = weak then
15 |  foreach x ∈ flagged(DS, c_min, not-safe) do amplify(DS, x, k)
16 else if ampl = strong then
17 |  foreach x ∈ flagged(DS, c_min, not-safe) do
18 |  |  if correct(DS, x, k + 2) then amplify(DS, x, k)
19 |  |  else amplify(DS, x, k + 2)
```

In the pseudo-code we use the following auxiliary functions: $\texttt{correct}(S, x, k)$ – classifies example x using its k-nearest neighbors in set S and returns true or false for correct and incorrect classification respectively; $\texttt{class}(S, c)$ – returns a subset of examples from S that belong to class c; $\texttt{flagged}(S, c, f)$ – returns a subset of examples from S that belong to class c and are flagged as f; $\texttt{knn}(S, x, k, c)$ – identifies and returns these examples among the k-nearest neighbors of x in S that belong to class c; $\texttt{amplify}(S, x, k)$ – amplifies example x by creating its n-copies and adding them to S. n is calculated as $|\texttt{knn}(DS, x, k, c_{maj})| - |\texttt{knn}(DS, x, k, c_{min})| + 1$. In these functions we employ the heterogeneous value distance metric (HVDM) [7] to identify the nearest neighbors of a given example.

SPIDER2 consists of two phases corresponding to pre-processing of c_{maj} and c_{min} respectively. In the first phase (lines 2–10) it identifies the characteristics of examples from c_{maj}, and depending on the *relabel* option it either removes or relabels noisy examples from c_{maj} (i.e., changes their classification to c_{min}). In the second phase (lines 11–19) it identifies the characteristic of examples from c_{min} considering changes introduced in the first phase. Then, noisy examples from c_{min} are amplified (by replicating them) according to the ampl option. This two-phase structure is a major difference from the first SPIDER version [8], which first identified the nature of examples and then simultaneously processed c_{maj} and c_{min}. As we noticed in [9] such processing could result in too extensive modifications in some regions of c_{maj} and deteriorated specifity – this drawback has been addressed in SPIDER2. Minor differences include the scope of relabeling noisy examples from c_{maj} and the degree of amplifying noisy examples from c_{min}.

3 Experiments with Artificial Data Sets

3.1 Preparation of Data Sets

Following the discussion in Section 1 on the factors influencing the performance of classifiers learned from imbalanced data, we decided to prepare artificial data sets in order to control these factors. We focused on binary classification problems (the minority vs. the majority class) with examples randomly and uniformly distributed in the two-dimensional space (both attributes were real-valued).

We considered three different shapes of the minority class: *subclus*, *clover* and *paw*, all surrounded uniformly by the majority class. In *subclus*, examples from the minority class are located inside rectangles following related works on small disjuncts [4]. *Clover* represents a more difficult, non-linear setting, where the minority class resembles a flower with elliptic petals (Fig. 1 shows *clover* with 5 petals). Finally, in *paw* the minority class is decomposed into 3 elliptic subregions of varying cardinalities, where two subregions are located close to each other, and the remaining smaller sub-region is separated (see Fig. 2). Such a shape should better represent real-life data than *clover*. Moreover, both *clover* and *paw* should be more difficult to learn than simple circles that were considered in some related works.

We generated multiple data sets with different numbers of examples (ranging from 200 to 1200) and imbalance ratios (from 1:3 to 1:9). Additionally, following Japkowicz's research on small disjuncts [4], we considered a series of the *subclus* and *clover* shapes with the number of sub-regions ranging from 1 to 5, and from 2 to 5 respectively. In a preliminary experiment we used tree- and rule-based classifiers on these data sets. Due to the space limit, we are not able to present the complete results. Let us only comment that they are consistent with the observations reported in [5] – increasing the number of sub-regions combined with decreasing the size of a data set degraded the performance of a classifier.

According to the results of the preliminary experiment we finally selected a group of data sets with 800 examples, the imbalance ratio of 1:7, and 5 subregions for the *subclus* and *clover* shapes. All these sets presented a significant

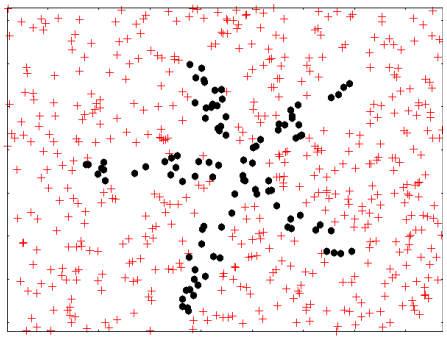

Fig. 1. Clover data set

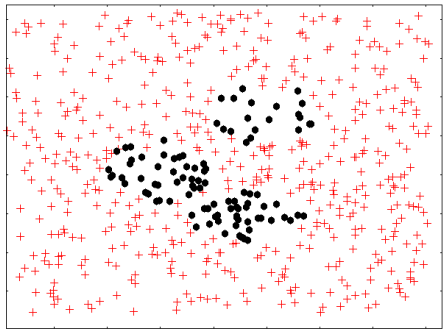

Fig. 2. Paw data set

challenge for a stand-alone classifier. We also observed similar behavior for data sets with 600 examples, but due to space limit we did not describe these data sets in the paper.

3.2 Disturbing Borders of the Minority-Class Subregions

In the first series of experiments we studied the impact of disturbing the borders of sub-regions in the minority class. We simulated it by increasing the ratio of borderline examples from the minority class subregions. We changed this ratio (further called the *disturbance ratio*) from 0 to 70%. The width of the borderline overlapping areas was comparable to the width of the sub-regions. We employed rule- and tree-based classifiers induced with the MODLEM and C4.5 algorithms, as they had been used in earlier studies [9] and had shown to be sensitive to the class imbalance. Both algorithms were run without prunning to get more precise description of the minority class.

The constructed classifiers were combined with the following pre-processing methods: random oversampling (RO), cluster oversampling (CO), NCR and SPIDER2 (SP2). Cluster oversampling was limited to the minority class, and our method was used with relabeling and strong amplification (*relabel* = yes, *ampl* = strong – see Section 2 for details) as such combination performed best in our earlier studies. For baseline results (Base), we ran both classifiers without any pre-processing. As evaluation measures we used sensitivity and specificity for the minority class, their geometric mean (G-mean), and the overall accuracy. We chose G-mean over AUC because it was more intuitive and suited to deterministic rule- and tree-based classifiers. All measures were estimated by 10-fold cross validation repeated 5 times.

Table 1 presents sensitivity recorded for data sets of different shapes (*subclus*, *clover* and *paw*) and different degrees of disturbance (0, 30, 50 and 70%). Increasing this degree strongly deteriorated the performance of both baseline classifiers. Pre-processing always improved performance in comparison to Base. RO and CO performed comparably on all data sets, and on non-disturbed data

Table 1. Sensitivity for artificial data sets with varying degree of the disturbance ratio

Data set	MODLEM					C4.5				
	Base	RO	CO	NCR	SP2	Base	RO	CO	NCR	SP2
subclus-0	0.8820	0.8820	0.9040	0.8640	0.8800	0.9540	0.9500	0.9500	0.9460	0.9640
subclus-30	0.5600	0.5520	0.5500	0.5540	0.5540	0.4500	0.6840	0.6720	0.7160	0.7720
subclus-50	0.3400	0.3580	0.3960	0.5300	0.4360	0.1740	0.6160	0.6000	0.7020	0.7700
subclus-70	0.1980	0.2380	0.2600	0.4300	0.3900	0.0000	0.6380	0.7000	0.5700	0.8300
clover-0	0.5720	0.5740	0.6060	0.6380	0.6560	0.4280	0.8340	0.8700	0.4300	0.4860
clover-30	0.4300	0.4300	0.4520	0.5700	0.5000	0.1260	0.7180	0.7060	0.5820	0.7260
clover-50	0.2860	0.3420	0.3380	0.5420	0.4040	0.0540	0.6560	0.6960	0.4460	0.7700
clover-70	0.2100	0.2520	0.2740	0.5100	0.3700	0.0080	0.6340	0.6320	0.5460	0.8140
paw-0	0.8320	0.8460	0.8560	0.8640	0.8180	0.5200	0.9140	0.9000	0.4900	0.5960
paw-30	0.6100	0.6260	0.6180	0.6660	0.6440	0.2640	0.7920	0.7960	0.8540	0.8680
paw-50	0.4560	0.5000	0.4980	0.6260	0.5500	0.1840	0.7480	0.7200	0.8040	0.8320
paw-70	0.2880	0.3700	0.3600	0.5900	0.4740	0.0060	0.7120	0.6800	0.7460	0.8780

sets they often over-performed NCR and SP2. On more difficult sets (disturbance = 50–70%) neighbor-based methods (NCR and SP2) were better than oversampling. Finally, MODLEM worked better with NCR, while C4.5 with SP2, especially on more difficult data sets.

In terms of specificity, Base performed best as expected. As previously, RO and CO were comparable and they were the second ones. Moreover, the relationship between NCR and SP2 was dependent on the induction algorithm – NCR performed better than SP2 when combined with C4.5, and for MODLEM SP2 won over NCR. However, considering G-mean (see Table 2), NCR and SP2 were better than oversampling methods. Finally, linear rectangle shapes (*subclus*) were easier to learn than non-linear ones (*clover* or *paw*). We are aware that tree- and rule-based classifiers are known to be sensitive to non-linear decision boundaries, and in future research we plan to study other classifiers (e.g., support vector machines) as well.

3.3 Impact of Different Types of Testing Examples

In the second series of experiments we concentrated on the impact of noisy examples from the minority class, located outside the borderline area, on the performance of a classifier. To achieve this, we introduced new noisy examples (single and pairs) and denoted them with C. Similarly to the first series of experiments we used data sets of three shapes (*subclus*, *clover* and *paw*), 800 examples and the imbalance ratio of 1:7. We also employed rule- and tree-based classifiers combined with the same pre-processing methods. However, we changed the 10-fold cross validation to the train-test verification in order to ensure that learning and testing sets had similar distributions of the C noise. In each training set 30% of the minority class examples were safe examples located inside subregions, 50% were located in the borderline area (we denote them with B), and the remaining 20% constituted the C noise.

Table 2. G-mean for artificial data sets with varying degree of the disturbance ratio

Data set	MODLEM					C4.5				
	Base	RO	CO	NCR	SP2	Base	RO	CO	NCR	SP2
subclus-0	0.9373	0.9376	0.9481	0.9252	0.9294	0.9738	0.9715	0.9715	0.9613	0.9716
subclus-30	0.7327	0.7241	0.7242	0.7016	0.7152	0.6524	0.7933	0.7847	0.7845	0.8144
subclus-50	0.5598	0.5648	0.6020	0.6664	0.6204	0.3518	0.7198	0.7113	0.7534	0.7747
subclus-70	0.4076	0.4424	0.4691	0.5957	0.5784	0.0000	0.7083	0.7374	0.6720	0.7838
clover-0	0.7392	0.7416	0.7607	0.7780	0.7908	0.6381	0.8697	0.8872	0.6367	0.6750
clover-30	0.6361	0.6366	0.6512	0.7221	0.6765	0.2566	0.7875	0.7652	0.6758	0.7686
clover-50	0.5066	0.5540	0.5491	0.6956	0.6013	0.1102	0.7453	0.7570	0.6184	0.7772
clover-70	0.4178	0.4658	0.4898	0.6583	0.5668	0.0211	0.7140	0.7027	0.6244	0.7665
paw-0	0.9041	0.9126	0.9182	0.9184	0.8918	0.6744	0.9318	0.9326	0.6599	0.7330
paw-30	0.7634	0.7762	0.7701	0.7852	0.7780	0.3286	0.8374	0.8334	0.8527	0.8337
paw-50	0.6587	0.6863	0.6865	0.7517	0.7120	0.3162	0.8013	0.7858	0.8200	0.8075
paw-70	0.5084	0.5818	0.5691	0.7182	0.6506	0.0152	0.7618	0.7472	0.7824	0.8204

For each training set we prepared 4 testing sets containing the following types of examples from the minority class: only safe examples, only B examples, only C examples, and B and C examples combined together (BC). Results are presented in Table 3. They clearly show that for the "difficult" noise (C or BC) SP2 and in most cases NCR were superior to RO, CO and Base. SP2 was also comparable to RO and CO in case of safe and (sometimes) B examples.

4 Experiments on Real-Life Data Sets

The goal of the third series of experiments was to discover the differences between those real-life data sets where NRC and SPIDER2 were superior to oversampling, and those, for which there was no such advantage. Moreover, we wanted to relate these differences to the factors explored in the previous experiments (see Section 3.1 and 3.2).

Experiments in this series were conducted on imbalanced data sets that we had used in our previous study [9]. They came either from the UCI repository or

Table 3. Sensitivity for artificial data sets with different types of testing examples

Data set	MODLEM					C4.5				
	Base	RO	CO	NCR	SP2	Base	RO	CO	NCR	SP2
subcl-safe	0.5800	0.5800	0.6200	0.7800	0.6400	0.3200	0.8400	0.8600	0.9800	1.0000
subcl-B	0.8400	0.8400	0.8400	0.8600	0.8400	0.0000	0.8200	0.8400	0.3600	0.9200
subcl-C	0.1200	0.1000	0.1600	0.2400	0.2600	0.0000	0.5400	0.0000	0.0000	0.5200
subcl-BC	0.4800	0.4700	0.5000	0.5500	0.5500	0.0000	0.6800	0.4200	0.1800	0.7200
clover-safe	0.3000	0.3800	0.4400	0.7000	0.6000	0.0200	0.9600	0.9200	0.0400	0.9800
clover-B	0.8400	0.8200	0.8200	0.8400	0.8600	0.0400	0.9400	0.9200	0.0400	0.9400
clover-C	0.1400	0.0800	0.1400	0.2400	0.3600	0.0000	0.3000	0.0200	0.0000	0.4000
clover-BC	0.4900	0.4500	0.4800	0.5400	0.6100	0.0200	0.6200	0.4700	0.0200	0.6700
paw-safe	0.8400	0.9200	0.8400	0.8400	0.8000	0.4200	0.9000	0.9600	0.7400	1.0000
paw-B	0.8800	0.8800	0.8600	0.8800	0.9000	0.1400	0.9000	0.9000	0.4000	0.9200
paw-C	0.1600	0.1400	0.1200	0.2600	0.1600	0.0400	0.2000	0.0000	0.0000	0.3400
paw-BC	0.5200	0.5100	0.4900	0.5700	0.5300	0.0900	0.5500	0.4500	0.2000	0.6300

Table 4. Sensitivity for real data sets

Data set	MODLEM					C4.5				
	Base	RO	NCR	SP1	SP2	Base	RO	NCR	SP1	SP2
Acl	0.8050	0.8050	0.9000	0.8250	0.8350	0.8550	0.8400	0.9200	0.8500	0.8450
Breast can.	0.3186	0.3430	0.6381	0.5386	0.5983	0.3867	0.4683	0.6478	0.5308	0.5611
Bupa	0.5199	0.5931	0.8734	0.8047	0.8580	0.4910	0.5720	0.7549	0.6995	0.7487
Cleveland	0.0850	0.1717	0.3433	0.2350	0.2300	0.2367	0.2383	0.3983	0.3017	0.3067
Ecoli	0.4000	0.5400	0.6833	0.6367	0.6217	0.5800	0.5567	0.7583	0.6900	0.7100
Haberman	0.2397	0.2961	0.6258	0.4828	0.5431	0.4103	0.6069	0.6081	0.6600	0.6775
Hepatitis	0.3833	0.4017	0.4550	0.4367	0.4867	0.4317	0.5583	0.6217	0.4750	0.5633
New thyr.	0.8117	0.8733	0.8417	0.8650	0.8867	0.9217	0.9217	0.8733	0.9133	0.8917
Pima	0.4853	0.5206	0.7933	0.7377	0.8188	0.6013	0.6512	0.7678	0.7146	0.7655

from our medical case studies (*acl*). As in the previous two series of experiments, we used C4.5 and MODLEM without prunning to induce classifiers and combined them we the pre-processing methods listed in Section 3.2. We only had to exclude cluster oversampling due to difficulties with defining the proper number of sub-clusters in the minority class. Moreover, for comprehensive comparison we included the first version of SPIDER with strong amplification (SP1). Evaluation measures were estimated in 10-fold cross validation repeated 5 times and the results for sensitivity are given in Table 4.

We used the Wilcoxon Signed Ranks Test (with confidence $\alpha = 0.05$) for pairwise comparison of pre-processing methods over all data sets. For MODLEM, all the pre-processing methods outperformed Base. The same conclusion applied to C4.5 with the exception of RO. Moreover, SP2 outperformed SP1, and differences between NCR and SP2 were not significant according to the test. Although NCR demonstrated slightly better sensitivity, its specificity (not reported here) was lower than for SP2.

When examining the performance of pre-processing methods on individual data sets we found some (e.g. *new thyroid*) for which all methods were comparable. Moreover, for data sets like *acl*, the advantage of SP2 or NCR over RO and CO was smaller than for the others, e.g., *breast cancer, bupa* or *pima*. We wanted to explore the characteristic of these sets by visualizing the distributions of the minority and majority classes in the two-dimensional space. Since all data sets included more than two attributes, we used the PCA method to identify two most important principal components for visualization. We are aware that such analysis may have yielded approximate results (for some data sets more than two components may have been important), nevertheless it led to interesting observations that are reported below.

On the one hand, the minority and majority classes in *acl* and *new thyroid* were easily separable (see Fig. 3 for *new thyroid*), thus even high imbalance ratio was not a serious problem and oversampling methods (RO, CO) were comparable to focused re-sampling (SP2, NCR). On the other hand, the distributions of classes in data sets where NCR or SP2 outperformed RO and CO, e.g., *haberman*, *bupa* or *pima*, were definitely more complicated (see Fig. 4 for *haberman*). Examples from the minority and majority classes were shuffled, there was no clear class boundary, the overlapping area was very large and there were many noisy

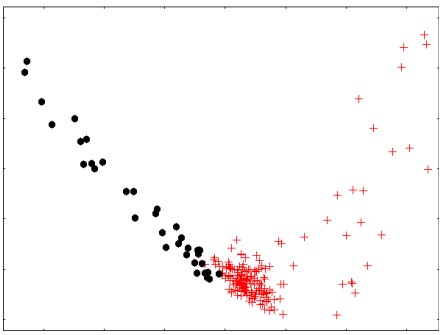

Fig. 3. New thyroid data set

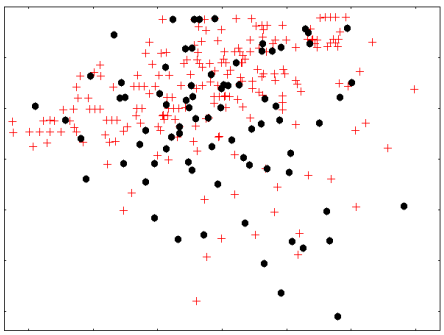

Fig. 4. Haberman data set

examples. This may explain the superior performance of focused re-sampling, as both employed methods (NCR and SPIDER2) were meant to deal with such conditions.

5 Conclusions and Final Remarks

We have presented an experimental study on the impact of critical factors on re-sampling methods dealing with imbalanced data. The first series of experiments show that the degradation in performance of a classifier is strongly affected by the number of borderline examples. If the overlapping area is large enough (in comparison to the area of the minority sub-clusters), and at least 30% of examples from the minority class are located in this area (i.e., are borderline examples), then focused re-sampling methods (NCR, SPIDER2) strongly outperform random and cluster oversampling with respect to sensitivity and G-mean. Moreover, the performance gain increases with the number of borderline examples. On the contrary, if the number of borderline examples is small, then oversampling methods sufficiently improve the recognition of the minority class.

The second series of experiments reveals the superiority of SPIDER2 and in most cases NCR in handling noisy examples located inside the majority class (also accompanied with borderline ones). Such result has been in a way expected, as both methods were introduced to handle such situations. The experiments also demonstrate that oversampling is comparable to SPIDER2 and better than NCR in classifying safe examples from the minority class.

The last series of experiments on real-life imbalanced data sets also provides interesting observations on their nature. We think that PCA-based visualizations of the data sets, on which NCR and both SPIDER methods performed best, are similar to visualizations of artificial data sets with multiple noisy examples and large overlapping ares. In the data sets, where all pre-processing methods worked comparatively, the minority and majority classes are easily separable and the number of "disturbances" is very limited. Thus, we can hypothesize

that difficulties with real-life data are associated with distributions and shapes of classes, their decomposition, overlapping and noise, however, this should be investigated closer in future research.

Although other authors [4,2] have already claimed that class imbalance is not a problem in itself, but the degradation of classification performance is related to other factors related to data distributions (e.g., small disjuncts), we hope that our experimental results expand the body of knowledge on the critical role of borderline and noisy examples.

References

1. Chawla, N.: Data mining for imbalanced datasets: An overview. In: Maimon, O., Rokach, L. (eds.) The Data Mining and Knowledge Discovery Handbook, pp. 853–867. Springer, Heidelberg (2005)
2. Garcia, V., Sanchez, J., Mollineda, R.: An empirical study of the behavior of classifiers on imbalanced and overlapped data sets. In: Rueda, L., Mery, D., Kittler, J. (eds.) CIARP 2007. LNCS, vol. 4756, pp. 397–406. Springer, Heidelberg (2007)
3. He, H., Garcia, E.: Learning from imbalanced data. IEEE Transactions on Data and Knowledge Engineering 21(9), 1263–1284 (2009)
4. Japkowicz, N.: Class imbalance: Are we focusing on the right issue? In: Proc. II Workshop on Learning from Imbalanced Data Sets, ICML, pp. 17–23 (2003)
5. Jo, T., Japkowicz, N.: Class Imbalances versus small disjuncts. SIGKDD Explorations 6(1), 40–49 (2004)
6. Kubat, M., Matwin, S.: Addresing the curse of imbalanced training sets: one-side selection. In: Proc. of the 14th Int. Conf. on Machine Learning, pp. 179–186 (1997)
7. Laurikkala, J.: Improving identification of difficult small classes by balancing class distribution. Tech. Report A-2001-2, University of Tampere (2001)
8. Stefanowski, J., Wilk, S.: Improving rule based classifiers induced by MODLEM by selective pre-processing of imbalanced data. In: Proc. of the RSKD Workshop at ECML/PKDD, pp. 54–65 (2007)
9. Stefanowski, J., Wilk, S.: Selective pre-processing of imbalanced data for improving classification performance. In: Song, I.-Y., Eder, J., Nguyen, T.M. (eds.) DaWaK 2008. LNCS, vol. 5182, pp. 283–292. Springer, Heidelberg (2008)