The class imbalance problem in pattern classification and learning

V. García J.S. Sánchez R.A. Mollineda R. Alejo J.M. Sotoca

Pattern Analysis and Learning Group

Dept.de Llenguatjes i Sistemes Informàtics

Universitat Jaume I

12071 Castelló de la Plana (SPAIN)

{sanchez,mollined,sotoca}@uji.es

Abstract

It has been observed that class imbalance (that is, significant differences in class prior probabilities) may produce an important deterioration of the performance achieved by existing learning and classification systems. This situation is often found in real-world data describing an infrequent but important case. In the present work, we perform a review of the most important research lines on this topic and point out several directions for further investigation.

1 Introduction

In recent years, the class imbalance problem has received considerable attention in areas such as Machine Learning and Pattern Recognition. A two-class data set is said to be imbalanced (or skewed) when one of the classes (the minority one) is heavily under-represented in comparison to the other class (the majority one).

This issue is particularly important in real-world applications where it is costly to misclassify examples from the minority class, such as diagnosis of rare diseases, detection of fraudulent telephone calls, text categorization, information retrieval and filtering tasks. Because of examples of the minority and majority classes usually represent the presence and absence of rare cases, respectively, they are also known as positive and negative examples.

Traditionally, research on this topic has mainly focused on a number of solutions both at the data and algorithmic levels. However, there have recently appeared other research lines within the general framework of class imbalance. These can be categorized into four groups:

- Resampling methods for balancing the data set.
- Modification of existing learning algorithms.
- Measuring the classifier performance in imbalanced domains.
- Relationship between class imbalance and other data complexity characteristics.

In the following sections, we explore each of these major issues, and comment several features to be considered in the general case of imbalanced multi-class problems. This paper does not intend to be a complete review of all the methods and algorithms covering the class imbalance problem, but to remark the main research developments carried out in this area.

2 Resampling techniques

Data level methods for balancing the classes consists of resampling the original data set, either by over-sampling the minority class or by under-sampling and/or under-sampling the majority class, until the classes are approximately equally represented. However, both

strategies have shown important drawbacks. Under-sampling may throw out potentially useful data, while over-sampling artificially increases the size of the data set and consequently, worsens the computational burden of the learning algorithm. Nevertheless, both methods have mainly been criticized due to the fact of altering the original class distribution.

2.1 Over-sampling

The simplest method to increase the size of the minority class corresponds to random over-sampling, that is, a non-heuristic method that balances the class distribution through the random replication of positive examples [5, 34]. Nevertheless, this method can increase the likelihood of overfitting, since it makes exact copies of the minority class instances [9].

Chawla et al. [9] proposed a technique for over-sampling the minority class and, instead of merely replicating cases belonging to the minority class, this generates new synthetic minority instances by interpolating between several positive examples that lie close together. This method, called SMOTE, allows the classifier to build larger decision regions that contain nearby instances from the minority class.

From the original SMOTE algorithm, several modifications have been proposed in the literature, most of them pursuing to determine the region in which the positive examples should be generated. For instance, Borderline-SMOTE [21] consists of using only positive examples close to the decision boundary, since these are more likely to be misclassified.

Cohen et al. [11] define new artificial positive examples by using the centroids obtained from preprocessing the data set with a clustering algorithm.

Although over-sampling increase the computational cost of the learning algorithm, experiments carried out by Batista et al. [5] show the convenience of applying this technique when the data set has very few positive (minority) examples. Similar conclusions were also obtained by Barandela et al. [3] in data sets with a very high majoritiy/minority ratio.

2.2 Under-sampling

Random under-sampling [25, 52] aims at balancing the data set through the random removal of negative examples. Despite its simplicity, it has empirically been shown to be one of the most effective resampling method. The major problem of this technique is that it can discard data potentially important for the classification process.

Unlike the random method, many proposals are based on a more intelligent selection of the majority class examples to eliminate. For example, Kubat and Matwin [30] proposed an under-sampling technique that selectively removes only those negative instances that are "redundant" or that "border" the minority class examples (they assume that these bordering cases are noise). The border examples were detected using the Tomek links concept [45]. On the other hand, Barandela et al. [1] introduced a method that eliminates noisy instances of the majority class by means of the Wilson's algorithm [50] for cleaning the data. Similarly, Batista et al. [5] remove negative examples whose labels do not match with the label of their 3-nearest neighbors, along with those samples of the majority class that misclassified positive examples. One of the main drawbacks of these methods is that there does not exist a control to remove patterns of the majority class, thus the resulting class distribution might not be balanced.

A quite different alternative corresponds to the restricted decontamination technique [2], which consists of discarding some negative instances while relabelling some others. This obtains a decrease in the amount of examples of the majority class. At the same time, some instances, originally in the majority class, are incorporated (by changing their labels) to the minority class, thus increasing its size.

Other works explore the utility of genetic algorithms to reduce the data set until obtaining a balance between the classes [18]. Similar experiments were carried out by Barandela et al. [4] to simultaneously select attributes and remove majority class instances in imbalance domains.

Finally, it is worth mentioning that several

investigations state the convenience of applying the under-sampling strategies when the level of imbalance is very low.

3 Solutions at the algorithmic level

As a way of facing the drawbacks of the resampling techniques, different proposals addresses the imbalance problem from an algorithmic point of view, that is, by adapting existing algorithms and techniques to the especial characteristics of the imbalanced data sets. Within this group, it is worth mentioning cost-sensitive learning, one-class classifiers, and classifier ensembles (or multiple classifier systems).

Also, belonging to this category, we can mention a set of proposals based on internally biasing the discrimination based process so as to compensate for the class imbalance. For example, Pazzani et al. [40] assign different weights to the instances of the different classes. Ezawa et al. [14] bias the classifier in favor of certain attribute relationships. Kubat et al. [32] use some counter-examples to bias the recognition process.

Barandela et al. [1] propose a weighted distance function to be used in the k-nearest neighbors classification. The basic idea behind this weighted distance is to compensate for the imbalance in the training sample without actually altering the class distribution. Thus, weights are assigned to the respective classes and not to the individual instances. In such a way, since the weighting factor is greater for the majority class than for the minority one, the distance to positive minority class examples becomes much lower than the distance to samples of the majority class. This produces a tendency for the new objects to find their nearest neighbor among the instances of the minority class.

Similarly, in the framework of support vector machines, some approaches bias the algorithm so that the learned hyperplane is further away from the positive class [46]. This is done in order to compensate for the skew associated with imbalanced data sets which pushes the hyperplane closer to the positive class.

3.1 Cost-sensitive learning

Traditional learning models implicitly assume the same misclassification costs for all classes. Nevertheless, in some domains the cost of a particular kind of error can be different from others. In general, misclassification costs may be described by an arbitrary cost matrix C, with $C(i, j)$ being the cost of predicting that an example belongs to class i when in fact it belongs to class j. The diagonal elements are usually set to zero,meaning that correct classification has no cost.

The realization that non-uniform costs are very usual in many real-world applications has led to an increased interest in algorithms for cost-sensitive learning. The goal in cost-sensitive learning is to minimize the cost of misclassification. Maloof [36] establishes that the problem of class imbalance, although not the same as learning when misclassification costs are unequal, can be handled in a similar manner.

Within this line, some works assign distinct costs to the classification errors for positive and negative examples [13, 20]. Japkowicz and Sthephen [25] propose the use of non-uniform error costs defined by means of the class imbalance ratio present in the data set. Liu and Zhou [37] employ a method to normalize the error costs in terms of the number of examples in each class.

3.2 One-class classifiers

In contrast with normal classification problems where there exist two (or more) classes of objects, one-class classification tries to describe one class of objects (the target class), and distinguish it from all other possible objects (the outlier class). The target class is assumed to be sampled well, in the sense that the training data reflect the area that the target class covers in the feature space. On the other hand, the outlier class can be sampled very sparsely, or can be totally absent. It might be that this class is very hard to measure, or it might be very expensive to do the measurements on these types of objects. In principle, a one-class classifier should be able to work,

solely on the basis of target examples.

The general ideas of one-class classification can be utilized in the class imbalance problem. In this case, the minority class can be viewed as the target class, whereas the majority class will be the outlier class. In particular, Raskutti and Kowalczyk [43] show that one-class learning is particularly useful on extremely unbalanced data sets with a high dimensional noisy feature space. Juszczak and Duin [27] combine one-class classifiers with a resampling method with the aim of adding information into the training set, both from the target (minority) class and the outlier (majority) class.

3.3 Classifier ensembles

Ensembles have been defined as consisting of a set of individually trained classifiers whose decisions are combined when classifying new objects. Combination (ensembles) of classifiers is now a well-established research line, mainly because it has been observed that the predictive accuracy of a combination of independent classifiers excels that of the single best classifier.

In the field of class imbalance, ensembles have mainly been used to combine the results of several classifiers, each induced after over-sampling or under-sampling the data with different over/under-sampling rates [29].

Chan and Stolfo [8] first run preliminary experiments to determine the best class distribution for learning and then generate multiple training samples with such a distribution. This is accomplished by including all the positive examples and some of the negative instances in each sample. Afterwards, they run a learning algorithm on each of the data sets and combine the induced classifiers to form a composite learner. This method ensures that all of the available training instances are used, since each negative example will be found in at least one of the training sets.

A similar approach for partitioning the data and learning multiple classifiers has been used with support vector machines. The resulting ensemble [51] was shown to outperform both under-sampling and over-sampling. While these ensemble approaches are effective for dealing with rare classes, they assume that a good class distribution is known. This can be estimated using some preliminary runs, but this increases the time required to learn.

Barandela et al. [1] proposed to replace an individual classification model with an imbalanced training set, by a combination of several classifiers, each using a balanced set for its learning process. To achieve this, as many training subsamples as required to get balanced subsets are generated. The number of subsamples is determined by the difference between the number of negative instances and that of the minority class. Then, each of the individual classifiers is trained with a learning set consisting of all the positive examples and the same number of training instances selected from among those belonging to the majority class.

SMOTEBoost [10] is based upon the well-known boosting iterative algorithms. Unlike the general boosting strategy, that is, changing the distribution of training data by updating the weights associated with each example, SMOTEBoost alters the distribution by adding new minority-class examples using the SMOTE algorithm.

Another method that follows the general idea of classifier ensembles employs a progressive-sampling algorithm to build larger and larger training sets, where the ratio of positive to negative examples added in each iteration is chosen based on the performance of the various class distributions evaluated in the previous iteration [49].

4 Performance measures

Most of performance measures for two-class problems are built over a 2×2 confusion matrix as illustrated in Table 1. From this, four simple measures can be directly obtained: TP and TN denote the number of positive and negative cases correctly classified, while FP and FN refer to the number of misclassified positive and negative examples, respectively.

The most widely used metrics for measuring the performance of learning systems are

Table 1: Confusion matrix for a two-class problem

	Positive prediction	*Negative prediction*
Positive class	True Positive (TP)	False Negative (FN)
Negative class	False Positive (FP)	True Negative (TN)

the *error rate* and the *accuracy*, defined as $err = (FP+FN)/(TP+FN+TN+FP)$ and $acc = (TP + TN)/(TP + FN + TN + FP)$, respectively. Nevertheless, it has widely been demonstrated that, when the prior class probabilities are very different, these measures are not appropriate because they do not consider misclassification costs, are strongly biased to favor the majority class, and are sensitive to class skews [42, 23, 12, 33]. For example, consider a problem where only 1% of the instances are positive. In such a situation, a simple strategy of labelling all new objects as negative would give a predictive accuracy of 99%, but failing on all positive cases.

Thus, in environments with imbalanced data, alternative metrics that measure the classification performance on positive and negative classes independently are needed.

The *true positive rate*, also referred to as *recall* or *sensitivity*, $TPrate = TP/(TP + FN)$, is the percentage of correctly classified positive instances. Analogously, the *true negative rate* (or *specificity*), $TNrate = TN/(TN + FP)$, is the percentage of correctly classified negative examples. The *false positive rate*, $FPrate = FP/(FP + TN)$, refers to the percentage of misclassified positive examples. The *false negative rate*, $FNrate = FN/(TP + FN)$ is the percentage of misclassified negative examples.

A way to combine the TP and FP rates consists of using the ROC curve. The ROC curve is a two-dimensional graph to visualize, organize and select classifiers based on their performance. It also depicts trade-offs between benefits (true positives) and costs (false positives) [42, 15]. In the ROC curve, the TP rate is represented on the Y-axis and the FP rate on the X-axis. Several points on a ROC graph should be noted. The lower left point (0,0) represents that the classifier labeled all examples as negative the upper right point (1,1) is the case where all examples are classified as positive, the point (0,1) represents perfect classification, and the line $y = x$ defines the strategy of randomly guessing the class.

In order for assessing the overall performance of a classifier, one can measure the fraction of the total area that falls under the ROC curve (AUC) [23]. AUC varies between 0 and +1. Larger AUC values indicate generally better classifier performance.

Kubat et al. [30] use *the geometric mean* (g-mean) of accuracies measured separately on each class, $g - mean = \sqrt{recall \times specificity}$. This measure relates to a point on the ROC curve and the idea is to maximize the accuracy on each of the two classes while keeping these accuracies balanced. An important property of the g-mean is that it is independent of the distribution of examples between classes. Another property is that it is nonlinear, that is, a change in recall (or specificity) has a different effect on this measure depending on the magnitude of recall (or specificity).

Several investigations establish that those measures being independent of class priors present a disadvantage in imbalanced environments. Consequently, it is used the *precision* (or *purity*), $Precision = TP/(TP+FP)$, which is defined as the proportion of positive instances that are actually correct. This measure, in combination with recall, employs a ROC analysis methodology [7, 33]. Another measure from the information retrieval community that is used in imbalanced domains corresponds to the $F - measure$, which combines precision and recall; it allows to control the influence of recall and precision separately. Investigations carried out by Daskalaki et al. [12] show that the use a geometric mean of precision and recall, which can be defined

as $gpr = \sqrt{precision \times recall}$, has a behavior similar to the $F - measure$.

5 Other data complexity characteristics

Many studies on the behavior of several standard classification systems in imbalance domains have shown that significant loss of performance is mainly due to skew of class distributions. However, recent investigations have pointed out that there does not exist a direct correlation between class imbalance and the loss of performance [41]. These works suggest that the class imbalance is not a problem by itself, but the degradation of performance is also related to other factors.

Size of the data set [39], distribution of the data within each class [24], small disjuncts [26, 41, 48], data duplication [28], density and overlap complexity [19, 47] have been reported among the most relevant situations in which classifier accuracy results negatively affected. These studies focus on distinct learning algorithms, from decision trees to neural networks and support vector machines, leading to different conclusions depending on the model employed, that is, similar situations may produce different results. Besides, it has also been shown that some classifiers are less affected by overlap, noise, small disjunct and imbalance, depending on their local or global nature.

6 Some comments on multi-class data sets

Most research efforts on imbalanced data sets has traditionally concentrated on two-class problems. However, this is not the only scenario where the class imbalance problem prevails. In the case of multi-class data sets, it is much more difficult to define the majority and minority classes. For example, one class A can be majority with respect to class B, but minority with respect to another class C.

There are not many works addressing the imbalance multi-class problem. Although multi-class problems can be converted into a series of binary classification problems and then methods effective in two-class learning can be used, recent studies [53] have shown that some two-class techniques are often not so useful when being applied to multi-class problems directly.

One of the first methods applicable to multi-class problems is MetaCost [13], which consists of a procedure for making a classifier cost-sensitive. Recently, Sun et al. [44] have developed a cost-sensitive boosting algorithm to improve the classification performance of imbalanced data involving multiple classes.

7 Conclusions

The purpose of this paper has been to review some of the most important investigations carried out in the field of class imbalance. First, we have categorized the main research lines into four groups: resampling techniques, adaptation of existing algorithms, performance measures, and the connection between class imbalance and other data complexity characteristics. For each of these groups, we have discussed the main developments, giving a number of references that can be useful for further research.

On the other hand, it has to be remarked that class imbalance is not only a problem of different class priors, but it is also necessary to consider the nature of the learning algorithm, since both issues are strongly related.

Acknowledgment

This work has been partially supported by grants DPI2006–15542 from the Spanish CICYT and GV/2007/105 form Generalitat Valenciana.

References

[1] Barandela R, Sánchez JS, García V, Rangel E (2003) Strategies for learning in class imbalance problems. Pattern Recognition 36:849–851.

[2] Barandela R, Rangel E, Sánchez JS, Ferri FJ (2003) Restricted decontamination for the imbalanced training sample problem. In: 8th Ibero-american Congress on Pattern Recognition, pp. 424–431.

[3] Barandela R, Valdovinos RM, Sánchez JS, Ferri FJ (2004) The Imbalanced Training Sample Problem: Under or Over Sampling?. In: Structural, Syntactic, and Statistical Pattern Recognition, Joint IAPR International Workshops, pp. 806–814.

[4] Barandela R, Hernández JK, Sánchez JS, Ferri FJ (2005) Imbalanced training set reduction and feature selection through genetic optimization. In: Artificial Intelligence Research and Developments, pp. 215–222.

[5] Batista GE, Pratti RC, Monard MC (2004) A study of the behavior of several methods for balancing machine learning training data. SIGKDD Explorations 6:20–29.

[6] Beyer KS, Goldstein J, Ramakrishnan R, Shaft U (1999) When is "nearest neighbor" meaningful?. Proceedings of 7th. Intl. Conf. on Database Theory 217–235.

[7] Cárdenas AA, Baras JS (2006) B-ROC curves for the assesment of classifiers over Imbalanced Data Sets. In: Proc. 21th National Conference on Artificial Intelligence (AAAI 06).

[8] Chan P, Stolfo S (1998) Toward scalable learning with non-uniform class and cost distributions: a case study in credit card fraud detection. In: Proc. 4th Int. Conf. on Knowledge Discovery and Data Mining, pp.164–168.

[9] Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research 16:321–357.

[10] Chawla NV, Lazarevic A, Hall LO, Bowyer KW (2003) Smoteboost: improving prediction of the minority class in boosting. In: Proc. 7th European Conf. on Principles and Practice of Knowledge Discovery in Databases, pp. 107–119.

[11] Cohen G, Hilario M, Sax H, Hugonnet S, Geissbuhler A (2005) Learning from imbalanced data in surveillance of nosocomial infection. Artificial Intelligence in Medicine 37:7–18.

[12] Daskalaki S, Kopanas I, Avouris N (2006) Evaluation of classifiers for an uneven class distribution problem. Applied Artificial Intelligence 20:381–417.

[13] Domingos P (1999) Metacost: a general method for making classifiers cost-sensitive. In: Proc. 5th Intl. Conf. on Knowledge Discovery and Data Mining, pp. 155–164.

[14] Ezawa KJ, Singh M, Norton SW (1996) Learning goal oriented Bayesian networks for telecommunications management. In: Proc. 13th Intl. Conf. on Machine Learning, pp. 139–147.

[15] Fawcett T (2006) ROC graphs with instance-varying costs. Pattern Recognition Letters 27:882–891.

[16] Fawcett T, Provost F (1996) Adaptive fraud detection. Data Mining and Knowledge Discovery 1:291–316.

[17] Friedman JH (1997) On bias, variance, 0/1–loss, and the curse of dimensionality. Data Mining and Knowledge Discovery 1:55–77.

[18] García S, Cano JR, Fernández A, Herrera F (2006) A proposal of evolutionary prototype selection for class imbalance problem. In: Proc. Intl. Conf. on Intelligent Data Engineering and Automated Learning, pp. 1415–1423.

[19] García V, Alejo R, Sánchez JS, Sotoca JM, Mollineda RA (2006) Combined effects of class imbalance and class overlap on instance-based classification. In: Proc. Intl. Conf. on Intelligent Data Engineering and Automated Learning, pp. 371-378.

[20] Gordon DF, Perlis D (1989) Explicitly biased generalization. Computational Intelligence 5:67–81.

[21] Han H, Wang WY, Mao BH (2005) Borderline-smote: a new oversampling method in imbalanced data sets learning. In: Proc. Intl. Conf. on Intelligence Computing, pp. 878–887.

[22] Hand DJ, Vinciotti V (2003) Choosing k for two-class nearest neighbour classifiers with unbalanced classes. Pattern Recognition Letters 24:1555–1562.

[23] Huang J, Ling CX (2005) Using AUC and accuracy in evaluating learning algorithms. IEEE Trans. on Knowledge and Data Engineering 17:299–310.

[24] Japkowicz N (2001) Concept-learning in the presence of between-class and within-class imbalances. In: Proc. 14th Conf. of the Canadian Society for Computational Studies of Intelligence, pp.67–77.

[25] Japkowicz N, Stephen S (2002) The class imbalance problem: a systematic study. Intelligent Data Analysis 6:429–450.

[26] Jo T, Japkowicz N (2004) Class imbalances versus small disjuncts. SIGKDD Explorations 6:40–49.

[27] Juszczak P, Duin RPW (2003) Uncertainty sampling methods for one-class classifiers. In: Proc. Intl. Conf. on Machine Learning, Workshop on Learning with Imbalanced Data Sets II, pp. 81–88.

[28] Kolez A, Chowdhury A, Alspector J (2003) Data duplication: an imbalance problem? In: Proc. Proc. Intl. Conf. on Machine Learning, Workshop on Learning with Imbalanced Data Sets II.

[29] Kotsiantis S, Pintelas P (2003) Mixture of expert agents for handling imbalanced data sets. Annals of Mathematics, Computing & TeleInformatics 1:46–55.

[30] Kubat M, Matwin S (1997) Adressing the curse of imbalanced training sets: one-sided selection. Proc. 14th Intl. Conf. on Machine Learning, pp. 179–186.

[31] Kubat M, Chen WK (1998) Weighted projection in nearest-neighbor classifiers. Proc. of 1st Southern Symposium on Computing, pp. 27–34.

[32] Kubat M, Holte R, Matwin S (1998) Detection of oil-spills in radar images of sea surface. Machine Learning 30:195–215.

[33] Landgrebe TCW, Paclick P, Duin RPW (2006) Precision-recall operating characteristic (P-ROC) curves in imprecise environments. In: Proc. 18th Intl. Conf. on Pattern Recognition, pp. 123–127.

[34] Ling CX, Li C (1998) Data mining for direct marketing: problems and solutions, In: Proc. 4th Intl. Conf. on Knowledge Discovery and Data Mining, pp. 73–79

[35] Little RJA, Rubin DB (2002) Statistical Analysis with Missing Data. John Wiley & Sons, New York.

[36] Maloof, MA (2003) Learning when data sets are imbalanced and when costs are unequal and unknow. In: Workshop on Learning from Imbalaned Data Sets (ICML03).

[37] Liu XY, Zhou ZH (2006) The influence of class imbalance on cost-sensitive learning: an empirical study. In: Proc. 6th Intl. Conf. on Data Mining, pp. 970–974.

[38] Okamoto S, Yugami N (2003) Effects of domain characteristics on instance-based learning algorithms. Theoretical Computer Science 298:207–233.

[39] Orriols A, Bernardó E (2005) The class imbalance problem in learning classifier systems: a preliminary study. In: Proc. Intl. Conf. on Genetic and Evolutionary Computation, pp. 74–78

[40] Pazzani M, Merz C, Murphy P, Ali K, Hume T, Brunk C (1994) Reducing misclassification costs. In: Proc. 11th Intl. Conf. on Machine Learning, pp. 217–225.

[41] Prati RC, Batista GE, Monard MC (2004) Class imbalance versus class overlapping: an analysis of a learning system behavior. In: Proc. 3rd Mexican Intl. Conf. on Artificial Intelligence, pp. 312–321.

[42] Provost F, Fawcett T (1997) Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. In: Proc. 3rd Intl. Conf. on Knowledge Discovery and Data Mining, pp. 43–48.

[43] Raskutti B, Kowalczyk A (2004) Extreme rebalancing for svms: a case study. SIGKDD Explorations 6:60–69.

[44] Sun Y, Kamel MS, Wang Y (2006) Boosting for learning multiple classes with imbalanced class distribution. In: Proc. 6th Int. Conf. on Data Mining, pp. 592–602.

[45] Tomek I (1976) Two modifications of CNN. IEEE Trans. on Systems, Man and Cybernetics 6:769–772.

[46] Veropoulos K, Campbell C, Cristianini N (1999) Controlling the sensitivity of support vector machines. In: Proc. Intl. Joint Conf. on Artificial Intelligence, pp. 55–60.

[47] Visa S, Ralescu A (2003) Learning from imbalanced and overlapped data using fuzzy sets. In: Proc. Intl. Conf. on Machine Learning, Workshop on Learning with Imbalanced Data Sets II, pp. 97–104.

[48] Weiss GM (2003) The Effect of small disjuncts and class distribution on decision tree learning. PhD thesis, Rutgers University.

[49] Weiss GM, Provost F (2003) Learning when training data are costly: the effect of class distribution on tree induction. Journal of Artificial Intelligence Research 19:315–354.

[50] Wilson DL (1972) Asymptotic properties of nearest neighbor rules using edited data sets. IEEE Trans. on Systems, Man and Cybernetics 2:408–421.

[51] Yan R, Liu Y, Jin R, Hauptmann A (2003) On predicting rare classes with SVM ensembles in scene classification. In: IEEE Intl. Conf. on Acoustics, Speech and Signal Processing, Vol. 3, pp. 21–24.

[52] Zhang J, Mani I (2003) kNN approach to unbalanced data distributions: a case study involving information extraction. In: Proc. Intl. Conf. on Machine Learning, Workshop on Learning with Imbalanced Data Sets II, pp. 42–48.

[53] Zhou ZH, Liu, XY (2006) Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Trans. on Knowledge and Data Engineering 18:63–77.