Handling imbalanced datasets: A review

Sotiris Kotsiantis, Dimitris Kanellopoulos, Panayiotis Pintelas

Educational Software Development Laboratory
Department of Mathematics, University of Patras, Greece
sotos@math.upatras.gr, dkanellop@teipat.gr, pintelas@math.upatras.gr

Abstract. Learning classifiers from imbalanced or skewed datasets is an important topic, arising very often in practice in classification problems. In such problems, almost all the instances are labelled as one class, while far fewer instances are labelled as the other class, usually the more important class. It is obvious that traditional classifiers seeking an accurate performance over a full range of instances are not suitable to deal with imbalanced learning tasks, since they tend to classify all the data into the majority class, which is usually the less important class. This paper describes various techniques for handling imbalance dataset problems. Of course, a single article cannot be a complete review of all the methods and algorithms, yet we hope that the references cited will cover the major theoretical issues, guiding the researcher in interesting research directions and suggesting possible bias combinations that have yet to be explored.

1 Introduction

High imbalance occurs in real-world domains where the decision system is aimed to detect a rare but important case. The problem of imbalance has got more and more emphasis in recent years. Imbalanced data sets exists in many real-world domains, such as spotting unreliable telecommunication customers, detection of oil spills in satellite radar images, learning word pronunciations, text classification, detection of fraudulent telephone calls, information retrieval and filtering tasks, and so on [27], [32], [33].

A number of solutions to the class-imbalance problem were previously proposed both at the data and algorithmic levels. At the data level [5], these solutions include many different forms of re-sampling such as random oversampling with replacement, random undersampling, directed oversampling (in which no new examples are created, but the choice of samples to replace is informed rather than random), directed undersampling (where, again, the choice of examples to eliminate is informed), oversampling with informed generation of new samples, and combinations of the above techniques. At the algorithmic level [26], solutions include adjusting the costs of the various classes so as to counter the class imbalance, adjusting the probabilistic estimate at the tree leaf (when working with decision trees), adjusting the decision threshold, and recognition-based (i.e., learning from one class) rather than discrimination-based (two class) learning. Mixture-of-experts approaches [20] (combining methods) have been

also used to handle class-imbalance problems. These methods combine the results of many classifiers; each usually induced after over-sampling or under-sampling the data with different over/under-sampling rates.
The error-prone nature of small disjuncts is a direct result of rarity. Therefore, an understanding of why small disjuncts are so error prone will help explain why rarity is a problem. One explanation is that some small disjuncts may not represent rare, or exceptional, cases, but rather something else—such as noisy data. Thus, only small disjuncts that are "meaningful" should be kept. Most classifier induction systems have some means of preventing overfitting, to remove subconcepts (i.e., disjuncts) that are not believed to be meaningful. Inductive bias also plays a role with respect to rare classes. Many induction systems will tend to prefer the more common classes in the presence of uncertainty (i.e., they will be biased in favor of the class priors).
Gary Weiss [30] presents an overview of the field of learning from imbalanced data. He pays particular attention to differences and similarities between the problems of rare classes and rare cases. He then discusses some of the common issues and their range of solutions in mining imbalanced datasets. This paper describes more techniques for handling imbalance dataset problems. The reader should be cautioned that a single article cannot be a comprehensive review of all these techniques. Instead, our goal has been to provide a representative sample of existing lines of research in each technique. In each of our listed areas, there are many other papers that more comprehensively detail relevant work.
Our next section covers the data level techniques for handling imbalance datasets, whereas algorithmic level techniques are described in section 3. Section 4 presents the combining methods for handling imbalance datasets. The metrics for evaluating of performance of classifiers in learning from imbalanced data are covered in section 5. Section 6 deals with other problems related with imbalance such as the small disjunct. Finally, the last section concludes this work.

2 Data level methods for handling imbalance

As we have already mentioned, data level solutions include many different forms of re-sampling such as random oversampling with replacement, random undersampling, directed oversampling, directed undersampling, oversampling with informed generation of new samples, and combinations of the above techniques.

2.1 Undersampling

Random under-sampling [23] is a non-heuristic method that aims to balance class distribution through the random elimination of majority class examples. The rationale behind it is to try to balance out the dataset in an attempt to overcome the idiosyncrasies of the machine learning algorithm. The major drawback of random undersampling is that this method can discard potentially useful data that could be important for the induction process. Another problem with this approach is that the purpose of machine learning is for the classifier to estimate the probability distribution of the

target population. Since that distribution is unknown we try to estimate the population distribution using a sample distribution. Statistics tells us that as long as the sample is drawn randomly, the sample distribution can be used to estimate the population distribution from where it was drawn. Hence, by learning the sample distribution we can learn to approximate the target distribution. Once we perform undersampling of the majority class, however, the sample can no longer be considered random.
Given two examples E_i and E_j belonging to different classes, and $d(E_i, E_j)$ is the distance between E_i and E_j; a (E_i, E_j) pair is called a Tomek link if there is not an example E_1, such that $d(E_i, E_1) < d(E_i, E_j)$ or $d(Ej , E_1) < d(E_i, E_j)$. If two examples form a Tomek link, then either one of these examples is noise or both examples are borderline. Tomek links can be used as an under-sampling method or as a data cleaning method. As an under-sampling method, only examples belonging to the majority class are eliminated, and as a data cleaning method, examples of both classes are removed.
Kubat and Matwin [24] randomly draw one majority class example and all examples from the minority class and put these examples in E´. Afterwards, use a 1-NN over the examples in E´ to classify the examples in E. Every misclassified example from E is moved to E´. The idea behind this implementation of a consistent subset is to eliminate the examples from the majority class that are distant from the decision border, since these sorts of examples might be considered less relevant for learning.

2.2 Oversampling

Random over-sampling is a non-heuristic method that aims to balance class distribution through the random replication of minority class examples. Several authors [5], [24] agree that random over-sampling can increase the likelihood of occurring overfitting, since it makes exact copies of the minority class examples. In this way, a symbolic classifier, for instance, might construct rules that are apparently accurate, but actually cover one replicated example. In addition, oversampling can introduce an additional computational task if the data set is already fairly large but imbalanced.
SMOTE generates synthetic minority examples to over-sample the minority class [5]. Its main idea is to form new minority class examples by interpolating between several minority class examples that lie together. For every minority example, its k (which is set to 5 in SMOTE) nearest neighbors of the same class are calculated, then some examples are randomly selected from them according to the over-sampling rate. After that, new synthetic examples are generated along the line between the minority example and its selected nearest neighbors. Thus, the overfitting problem is avoided and causes the decision boundaries for the minority class to spread further into the majority class space.

2.3 Feature Selection for imbalance datasets

Zheng et al [35] suggest that existing measures used for feature selection are not very appropriate for imbalanced data sets. They propose a feature selection framework, which selects features for positive and negative classes separately and then

explicitly combines them. The authors show simple ways of converting existing measures so that they separately consider features for negative and positive classes.

3 Algorithm level methods for handling imbalance

Drummond and Holte [9] report that when using C4.5's default settings, over-sampling is surprisingly ineffective, often producing little or no change in performance in response to modifications of misclassification costs and class distribution. Moreover, they note that over-sampling prunes less and therefore generalizes less than under-sampling, and that a modification of the C4.5's parameter settings to increase the influence of pruning and other overfitting avoidance factors can re-establish the performance of over-sampling.
For internally biasing the discrimination procedure, a weighted distance function is proposed in [1] to be used in the classification phase of kNN. 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, unlike in the usual weighted k-NN rule, to the respective classes and not to the individual prototypes. 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 prototypes becomes much lower than the distance to prototypes of the majority class. This produces a tendency for the new patterns to find their nearest neighbor among the prototypes of the minority class.
Another approach to dealing with imbalanced datasets using SVM biases the algorithm so that the learned hyperplane is further away from the positive class. This is done in order to compensate for the skew associated with imbalanced datasets which pushes the hyperplane closer to the positive class. This biasing can be accomplished in various ways. In [32] an algorithm is proposed that changes the kernel function to develop this bias. Veropoulos et al [28] suggested using different penalty constants for different classes of data, making errors on positive instances costlier than errors on negative instances.

3.1 Threshold method

Some classifiers, such as the Naıve Bayes classifier or some Neural Networks, yield a score that represents the degree to which an example is a member of a class. Such ranking can be used to produce several classifiers, by varying the threshold of an example pertaining to a class [30].

3.2 One-class learning

An interesting aspect of one-class (recognition-based) learning is that, under certain conditions such as multi-modality of the domain space, one class approaches to solv-

ing the classification problem may in fact be superior to discriminative (two-class) approaches (such as decision trees or Neural Networks) [17].
Ripper [7] is a rule induction system that utilizes a separate-and-conquer approach to iteratively build rules to cover previously uncovered training examples. Each rule is grown by adding conditions until no negative examples are covered. It normally generates rules for each class from the most rare class to the most common class. Given this architecture, it is quite straightforward to learn rules only for the minority class—a capability that Ripper provides.
In particular, Raskutti and Kowalczyk [27[show that one class learning is particularly useful when used on extremely unbalanced data sets composed of a high dimensional noisy feature space. They argue that the one-class approach is related to aggressive feature selection methods, but is more practical since feature selection can often be too expensive to apply.

3.3 Cost-sensitive learning

As we have already mentioned changing the class distribution is not the only way to improve classifier performance when learning from imbalanced datasets. A different approach to incorporating costs in decision-making is to define fixed and unequal misclassification costs between classes [8]. Cost model takes the form of a cost matrix, where the cost of classifying a sample from a true class j to class i corresponds to the matrix entry λ_{ij}. This matrix is usually expressed in terms of average misclassification costs for the problem. The diagonal elements are usually set to zero, meaning correct classification has no cost. We define conditional risk for making a decision α_i as:

$$R(a_i \mid x) = \sum_i \lambda_{ij} P(v_j \mid x)$$

The equation states that the risk of choosing class *i* is defined by fixed misclassification costs and the uncertainty of our knowledge about the true class of *x* expressed by the posterior probabilities. The goal in cost-sensitive classification is to minimize the cost of misclassification, which can be realized by choosing the class (v_j) with the minimum conditional risk.

4 Combining methods

A mixture-of-experts approach [10] has been used to combine the results of many classifiers, each induced after over-sampling or under-sampling the data with different over/under-sampling rates. This approach recognizes the fact that it is still unclear which sampling method performs best, what sampling rate should be used—and that the proper choice is probably domain specific. Results indicate that the mixture-of experts approach performs well, generally outperforming another method (AdaBoost) with respect to precision and recall on text classification problems, and doing especially well at covering the rare, positive, examples. More detailed experiments are presented in [11].

Chan and Stolfo [4] run a set of preliminary experiments to identify a good class distribution and then sample in such a way as to generate multiple training sets with the desired class distribution. Each training set typically includes all minority-class examples and a subset of the majority-class examples; however, each majority-class example is guaranteed to occur in at least one training set, so no data is wasted. The learning algorithm is applied to each training set and meta-learning is used to form a composite learner from the resulting classifiers. This approach can be used with any learning method and Chan and Stolfo evaluate it using four different learning algorithms. The same basic approach for partitioning the data and learning multiple classifiers has been used with support vector machines. The resulting SVM ensemble [33] 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.

Another method that uses this general approach 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 [31].

MetaCost [8] is another method for making a classifier cost-sensitive. The procedure begins to learn an internal cost-sensitive model by applying a cost-sensitive procedure, which employs a base learning algorithm. Then, MetaCost procedure estimates class probabilities using bagging and then re-labels the training instances with their minimum expected cost classes, and finally relearns a model using the modified training set.

Boosting algorithms are iterative algorithms that place different weights on the training distribution each iteration. After each iteration boosting increases the weights associated with the incorrectly classified examples and decreases the weights associated with the correctly classified examples. This forces the learner to focus more on the incorrectly classified examples in the next iteration. Because rare classes/cases are more error-prone than common classes/cases, it is reasonable to believe that boosting may improve their classification performance because, overall, it will increase the weights of the examples associated with these rare cases/classes. Note that because boosting effectively alters the distribution of the training data, one could consider it a type of advanced sampling technique.

AdaBoost's weight-update rule has been made cost-sensitive, so that examples belonging to rare class(es) that are misclassified are assigned higher weights than those belonging to common class(es). The resulting system, Adacost [12], has been empirically shown to produce lower cumulative misclassification costs than AdaBoost and thus, like other cost-sensitive learning methods, can be used to address the problem with rare classes.

Rare-Boost [20] scales false-positive examples in proportion to how well they are distinguished from true-positive examples and scales false-positive examples in proportion to how well they are distinguished from true-negative examples. Another algorithm that uses boosting to address the problems with rare classes is SMOTE-Boost [6]. This algorithm recognizes that boosting may suffer from the same problems as over-sampling (e.g., overfitting), since boosting will tend to weight examples

belonging to the rare classes more than those belonging to the common classes—effectively duplicating some of the examples belonging to the rare classes. Instead of 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.

Kotsiantis and Pintelas [23] used three agents (the first learns using Naıve Bayes, the second using C4.5 and the third using 5NN) on a filtered version of training data and combine their predictions according to a voting scheme. This technique attempts to achieve diversity in the errors of the learned models by using different learning algorithms. The intuition is that the models generated using different learning biases are more likely to make errors in different ways. They also used feature selection of the training data because in small data sets the amount of class imbalance affects more the induction and thus feature selection makes the problem less difficult.

Kaizhu Huang et al. [16] presented Biased Minimax Probability Machine (BMPM) to resolve the imbalance problem. Given the reliable mean and covariance matrices of the majority and minority classes, BMPM can derive the decision hyperplane by adjusting the lower bound of the real accuracy of the testing set.

5 Evaluation metrics

Using terminology from information retrieval, the minority class has much lower precision and recall than the majority class. Many practitioners have observed that for extremely skewed class distributions the recall of the minority class is often 0—there are no classification rules generated for the minority class.

Accuracy places more weight on the common classes than on rare classes, which makes it difficult for a classifier to perform well on the rare classes. Because of this, additional metrics are coming into widespread use.

Most of the studies in imbalanced domains mainly concentrate on two-class problem as multi-class problem can be simplified to two-class problem. By convention, the class label of the minority class is positive, and the class label of the majority class is negative. Table 1 presents the most well known evaluation metrics. TP and TN denote the number of positive and negative examples that are classified correctly, while FN and FP denote the number of misclassified positive and negative examples respectively.

Accuracy = (TP+TN)/(TP+FN+FP+TN) (1)
FP rate = FP/(TN+FP) (2)
TP rate = Recall = TP/(TP+FN) (3)
Precision = TP/(TP+FP) (4)
$Fvalue = \frac{(1+\beta^2)\,\mathrm{Re}call * \mathrm{Pr}ecision}{\beta^2\,\mathrm{Re}call + \mathrm{Pr}ecision}$ (5)

Table 1. Evaluation metrics

In general, four criteria are used to evaluate the performance of classifiers in learning from imbalanced data. They are (1)Minimum Cost criterion (MC), (2) the criterion of Maximum Geometry Mean (MGM) of the accuracy on the majority class and the minority class, (3) the criterion of the Maximum Sum (MS) of the accuracy on the majority class and the minority class, and (4) the criterion of Receiver Operating Characteristic (ROC) analysis.

The MC criterion [3] minimizes the cost measured by Cost = FP * CFP + FN * CFP, where CFP is the cost of a false positive and CFn is the cost of a false negative. However, the cost of misclassification is generally unknown in real cases, this restricts the usage of this measure. The criterion of MGM maximizes the geometric mean of the accuracy [24], but it contains a nonlinear form, which is not easy to be automatically optimized. Comparatively, MS maximizing the sum of the accuracy on the positive class and the negative class (or maximizing the difference between the true-positive and false-positive probability) [13], is a linear form.

Perhaps the most common metric is ROC analysis and the associated use of the area under the ROC curve (AUC) to assess overall classification performance [3]. AUC does not place more emphasis on one class over the other, so it is not biased against the minority class. ROC curves, like precision-recall curves, can also be used to assess different tradeoffs—the number of positive examples correctly classified can be increased at the expense of introducing additional false positives. In detail, ROC curve is a two-dimensional graph in which TP rate is plotted on the y-axis and FP rate is plotted on the x-axis. FP rate (formula (2)) denotes the percentage of the misclassified negative examples, and TP rate (formula (3)) is the percentage of the correctly classified positive examples. The point (0, 1) is the ideal point of the learners. ROC curve depicts relative trade-offs between benefits (TP rate) and costs (FP rate). Furthermore, F-value (formula (5)) is also a popular evaluation metric for imbalance problem [10]. It is a kind of combination of recall (formula (3)) and precision (formula (4)), which are effective metrics for information retrieval community where the imbalance problem exist. F-value is high when both recall and precision are high, and can be adjusted through changing the value of β, where β corresponds to relative importance of precision vs. recall and it is usually set to 1.

Regardless of how we produce ROC curves - by sampling, by moving the decision threshold, or by varying the cost matrix - the problem still remains of selecting the single best method and the single best classifier for deployment in an intelligent system. If the binormal assumption holds, the variances of the two distributions are equal, and error costs are the same, then the classifier at the apex of the dominant curve is the best choice.

When applying machine learning to real-world problems, rarely would one or more of these assumptions hold, but to select a classifier, certain conditions must exist, and we may need more information. If one ROC curve dominates all others, then the best method is the one that produced the dominant curve, which is also the curve with the largest area. This was generally true of our domains, but it is not true of others. To select a classifier from the dominant curve, we need additional information, such as a target false-positive rate. On the other hand, if multiple curves dominate in different parts of the ROC space, then we can use the ROC Convex Hull method to select the optimal classifier [26].

6 Other problems related with imbalance

However, it has also been observed that in some domains, for instance the Sick data set, standard ML algorithms are capable of inducing good classifiers, even using highly imbalanced training sets. This shows that class imbalance is not the only problem responsible for the decrease in performance of learning algorithms. Class imbalances are not the only problem to contend with: the distribution of the data within each class is also relevant (between-class versus within-class imbalance) [17], [34].
Prati et al [25] developed a systematic study aiming to question whether class imbalances hinder classifier induction or whether these deficiencies might be explained in other ways. Their study was developed on a series of artificial data sets in order to fully control all the variables they wanted to analyze. The results of their experiments, using a discrimination-based inductive scheme, suggested that the problem is not solely caused by class imbalance, but is also related to the degree of data overlapping among the classes.
A number of papers discussed interaction between the class imbalance and other issues such as the small disjunct [19] and the rare cases [15] problems, data duplication [22] and overlapping classes [29]. It was found that in certain cases, addressing the small disjunct problem with no regard for the class imbalance problem was sufficient to increase performance. The method for handling rare case disjuncts was found to be similar to the m-estimation Laplace smoothing, but it requires less tuning. It was also found that data duplication is generally harmful, although for classifiers such as Naive Bayes and Perceptrons with Margins, high degrees of duplication are necessary to harm classification [22]. It was argued that the reason why class imbalances and overlapping classes are related is that misclassification often occurs near class boundaries where overlap usually occurs as well.
Jo and Japkowicz [21] experiments suggest that the problem is not directly caused by class imbalances, but rather, that class imbalances may yield small disjuncts which, in turn, will cause degradation. The resampling strategy proposed by [21] consists of clustering the training data of each class (separately) and performing random oversampling cluster by cluster. Its idea is to consider not only the between-class imbalance (the imbalance occurring between the two classes) but also the within-class imbalance (the imbalance occurring between the subclusters of each class) and to oversample the dataset by rectifying these two types of imbalances simultaneously. Before performing random oversampling, the training examples in the minority and the majority classes must be clustered. Once the training examples of each class have been clustered, oversampling starts. In the majority class, all the clusters, except for the largest one, are randomly oversampled so as to get the same number of training examples as the largest cluster. Let maxclasssize be the overall size of the large class. In the minority class, each cluster is randomly oversampled until each cluster contains maxclasssize/Nsmallclass where Nsmallclass represents the number of subclusters in the small class. Altogether, the experiments support the hypothesis that cluster based oversampling works better than simple oversampling or other methods for handling class imbalances or small disjuncts, especially when the number of training examples is small and the problem, complex. The reason is that cluster-based resampling identi-

fies rare cases and re-samples them individually, so as to avoid the creation of small disjuncts in the learned hypothesis.

7 Conclusion

Practically, it is often reported that cost-sensitive learning outperforms random re-sampling [18]. Clever re-sampling and combination methods can do quite more than cost-sensitive learning as they can provide new information or eliminate redundant information for the learning algorithm, as shown by [5], [6], [24], [14], [2].
The relationship between training set size and improper classification performance for imbalanced data sets seems to be that on small imbalanced data sets the minority class is poorly represented by an excessively reduced number of examples that might not be sufficient for learning, especially when a large degree of class overlapping exists and the class is further divided into subclusters. For larger data sets, the effect of these complicating factors seems to be reduced, as the minority class is better represented by a larger number of examples.

References

[1] Barandela, R., Sánchez, J.S., García, V., Rangel, E.: Strategies for learning in class imbalance problems, Pattern Recognition 36(3) (2003) 849-851.
[2] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard. A study of the behavior of several methods for balancing machine learning training data. SIGKDD Explorations, 6(1):20-29, 2004.
[3] Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition, 30(7): 1145-1159, 1997.
[4] P. K. Chan, and S. J. Stolfo. Toward scalable learning with non-uniform class and cost distributions: a case study in credit card fraud detection. In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, pages 164-168, 2001.
[5] N. V. Chawla, L. O. Hall, K. W. Bowyer, and W. P. Kegelmeyer. SMOTE: Synthetic Minority Oversampling TEchnique. Journal of Artificial Intelligence Research, 16:321-357, 2002.
[6] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer. Smoteboost: Improving prediction of the minority class in boosting. In Proceedings of the Seventh European Conference on Principles and Practice of Knowledge Discovery in Databases, pages 107-119, Dubrovnik, Croatia, 2003.
[7] W. W. Cohen. Fast effective rule induction. In Proceedings of the Twelfth International Conference on Machine Learning, pages 115-123, 1995.
[8] Domingos, P. (1999). "MetaCost: A general method for making classifiers cost-sensitive. Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining, pp. 155-164. ACM Press.
[9] Drummond, C., and Holte, R. C. C4.5, Class Imbalance, and Cost Sensitivity: Why Under-sampling beats Over-sampling. In Workshop on Learning from Imbalanced Data Sets II (2003).

[10] Estabrooks, and N. Japkowicz. A mixture-of-experts framework for learning from unbalanced data sets. In Proceedings of the 2001 Intelligent Data Analysis Conference, pages 34-43, 2001.
[11] Andrew Estabrooks, Taeho Jo and Nathalie Japkowicz: A Multiple Resampling Method for Learning from Imbalanced Data Sets. Computational Intelligence 20 (1) (2004) 18-36.
[12] W. Fan, S. J. Stolfo, J. Zhang, and P. K. Chan. AdaCost: misclassification cost-sensitive boosting. In Proceedings of the Sixteenth International Conference on Machine Learning, pages 99-105, 1999.
[13] J.W. Grzymala-Busse, L. K. Goodwin, and X. Zhang. Increasing sensitivity of preterm birth by changing rule strengths. Pattern Recognition Letters, (24):903–910, 2003.
[14] H. Guo and H. L. Viktor. Learning from imbalanced data sets with boosting and data generation: The DataBoost-IM approach. SIGKDD Explorations, 6(1):30-39, 2004.
[15] R. Hickey. Learning rare class footprints: the reflex algorithm. In Proceedings of the ICML'03 Workshop on Learning from Imbalanced Data Sets, 2003.
[16] Kaizhu Huang, Haiqin Yang, Irwin King, Michael R. Lyu. Learning Classifiers from Imbalanced Data Based on Biased Minimax Probability Machine. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2004)
[17] N. Japkowicz. Concept-learning in the presence of between-class and within-class imbalances. In Proceedings of the Fourteenth Conference of the Canadian Society for Computational Studies of Intelligence, pages 67-77, 2001.
[18] N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. Intelligent Data Analysis, 6(5):203-231, 2002.
[19] N. Japkowicz. Class imbalance: Are we focusing on the right issue? In Proceedings of the ICML'03 Workshop on Learning from Imbalanced Data Sets, 2003.
[20] M. V. Joshi, V. Kumar, and R. C. Agarwal. Evaluating boosting algorithms to classify rare cases: comparison and improvements. In First IEEE International Conference on Data Mining, pages 257-264, November 2001.
[21] Taeho Jo and N. Japkowicz (2004), Class Imbalances versus Small Disjuncts, Sigkdd Explorations. Volume 6, Issue 1 - Page 40-49.
[22] Kolez, A. Chowdhury, and J. Alspector. Data duplication: An imbalance problem? In Proceedings of the ICML'2003 Workshop on Learning from Imbalanced Datasets, 2003.
[23] S. Kotsiantis, P. Pintelas, Mixture of Expert Agents for Handling Imbalanced Data Sets, Annals of Mathematics, Computing & TeleInformatics, Vol 1, No 1 (46-55), 2003.
[24] M. Kubat and S. Matwin. Addressing the curse of imbalanced training sets: One sided selection. In Proceedings of the Fourteenth International Conference on Machine Learning, pages 179-186, Nashville, Tennesse, 1997. Morgan Kaufmann.
[25] Prati, R. C., Batista, G. E. A. P. A., and Monard, M. C. Class Imbalances versus Class Overlapping: an Analysis of a Learning System Behavior. In MICAI (2004), pp. 312–321. LNAI 2972.
[26] Provost, F., & Fawcett, T. (2001). Robust classification for imprecise environments. Machine Learning, 42, 203-231.
[27] B. Raskutti and A. Kowalczyk. Extreme rebalancing for svms: a case study. SIGKDD Explorations, 6(1):60-69, 2004.
[28] Veropoulos, K., Campbell, C., & Cristianini, N. (1999). Controlling the sensitivity of support vector machines. Proceedings of the International Joint Conference on AI, 55–60.
[29] S. Visa and A. Ralescu. Learning imbalanced and overlapping classes using fuzzy sets. In Proceedings of the ICML'03 Workshop on Learning from Imbalanced Data Sets, 2003.
[30] G. Weiss. Mining with rarity: A unifying framework.SIGKDD Explorations, 6(1):7-19, 2004.

[31] G. M. Weiss, and F. Provost. Learning when training data are costly: the effect of class distribution on tree induction. Journal of Artificial Intelligence Research, 19:315-354, 2003.
[32] Wu, G. & Chang, E. (2003). Class-Boundary Alignment for Imbalanced Dataset Learning. In ICML 2003 Workshop on Learning from Imbalanced Data Sets II, Washington, DC.
[33] R. Yan, Y. Liu, R. Jin, and A. Hauptmann. On predicting rare classes with SVM ensembles in scene classification. In IEEE International Conference on Acoustics, Speech and Signal Processing, 2003.
[34] B. Zadrozny and C. Elkan. Learning and making decisions when costs and probabilities are both unknown. In Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 204-213, 2001.
[35] Z. Zheng, X. Wu, and R. Srihari. Feature selection for text categorization on imbalanced data. SIGKDD Explorations, 6(1):80-89, 2004.

Biography

S. Kotsiantis received a diploma in mathematics, a Master and a Ph.D. degree in computer science from the University of Patras, Greece. His research interests are in the field of data mining and machine learning. He has more than 40 publications to his credit in international journals and conferences.	**D. Kanellopoulos** received a diploma in electrical engineering and a Ph.D. degree in electrical and computer engineering from the University of Patras, Greece. He has more than 40 publications to his credit in international journals and conferences.	**P. Pintelas** is a Professor in the Department of Mathematics, University of Patras, Greece. His research interests are in the field of educational software and machine learning. He has more than 100 publications in international journals and conferences.