1-1-2009

Learning pattern classification tasks with imbalanced data sets

Son Lam Phung
University of Wollongong, phung@uow.edu.au

Abdesselam Bouzerdoum
University of Wollongong, bouzer@uow.edu.au

Giang Hoang Nguyen
University of Wollongong, giang_nguyen@uow.edu.au

Recommended Citation
Phung, Son Lam; Bouzerdoum, Abdesselam; and Nguyen, Giang Hoang: Learning pattern classification tasks with imbalanced data sets 2009, 193-208.
https://ro.uow.edu.au/infopapers/792

Learning pattern classification tasks with imbalanced data sets

Abstract

This chapter is concerned with the class imbalance problem, which has been recognised as a crucial problem in machine learning and data mining. The problem occurs when there are significantly fewer training instances of one class compared to another class.

Keywords

pattern, classification, sets, tasks, learning, imbalanced, data

Disciplines

Physical Sciences and Mathematics

Publication Details

Nguyen, G. Hoang., Bouzerdoum, A. & Phung, S. (2009). Learning pattern classification tasks with imbalanced data sets. In P. Yin (Eds.), Pattern recognition (pp. 193-208). Vukovar, Croatia: In-Teh.

10

Learning Pattern Classification Tasks with Imbalanced Data Sets

Giang Hoang Nguyen, Abdesselam Bouzerdoum and Son Lam Phung
University of Wollongong
Australia

1. Introduction

This chapter is concerned with the class imbalance problem, which has been recognized as a crucial problem in machine learning and data mining. The problem occurs when there are significantly fewer training instances of one class compared to another class. Most machine learning algorithms work well with balanced data sets since they aim to optimize the overall classification accuracy or a related measure. For imbalanced data sets, the decision boundary established by standard machine learning algorithms tends to be biased towards the majority class; therefore, the minority class instances are more likely to be misclassified.

There are many problems that arise from learning with imbalanced data sets. The first problem concerns measures of performance. Evaluation metrics are known to play a vital role in machine learning. They are used to guide the learning algorithm towards the desired solution. Therefore, if the evaluation metric does not take the minority class into consideration, the learning algorithm will not be able to cope with class imbalance very well. With standard evaluation metrics, such as the overall classification accuracy, the minority class has less impact compared to the majority class. The second problem is related to lack of data. In an imbalanced training set, a class may have very few samples. As a result, it is difficult to construct accurate decision boundaries between classes. For a class consisting of multiple clusters, some clusters may contain a small number of samples compared to other clusters; therefore, the lack of data can occur within the class itself. The third problem in learning from imbalanced data is noise. Noisy data have a serious impact on minority classes than on majority classes. Furthermore, standard machine learning algorithms tend to treat samples from a minority class as noise.

In this chapter, we review the existing approaches for solving the class imbalance problem, and discuss the various metrics used to evaluate the performance of classifiers. Furthermore, we introduce a new approach to dealing with the class imbalance problem by combining both unsupervised and supervised learning. The rest of the chapter is organized as follows. Section 2 describes the problems caused by class imbalance. Section 3 reviews current state-of-the-art techniques for tackling these problems. Section 4 describes existing classification performance measures for imbalanced data. Section 5 describes our proposed learning approach to handle the class imbalance problem. Section 6 presents experimental results, and Section 7 gives concluding remarks.

2. Class Imbalance Problems

Class imbalance occurs when there are significantly fewer training instances of one class compared to other classes. In some applications, class imbalance is an intrinsic property. For example, in credit card usage data there are very few cases of fraud transactions as compared to the number of normal transactions. However, imbalanced data can also occur in areas that do not have an inherent imbalance problem. Instead, the imbalance is mainly caused by limitations in collecting data, such as cost, privacy, and the large effort required to obtain a representative data set. Class imbalance presents several difficulties in learning, including imbalanced in class distribution, lack of data, and concept complexity. These factors are explained in more detail in the following subsections.

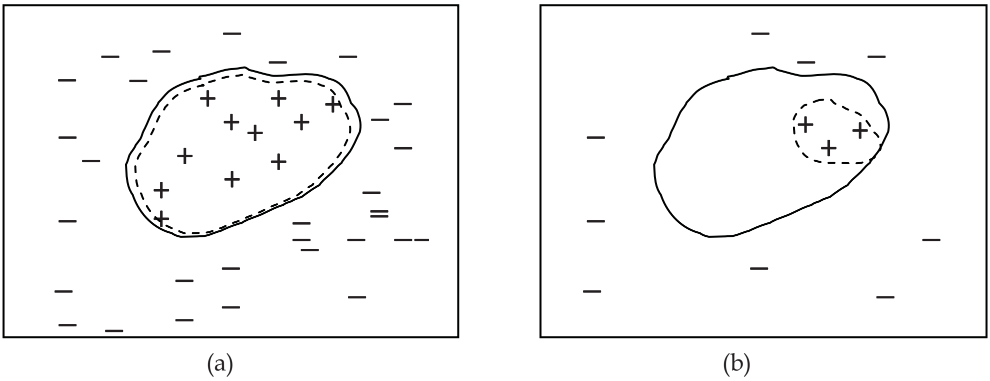
2.1 Imbalance in class distribution

The class imbalance problems can arise either from between classes (inter-class) or within a single class (intra-class). We first discuss issues related with inter-class imbalance, where the number of examples of one class is much larger than the number of examples of another class, namely the minority class. The degree of imbalance can be represented by the ratio of sample size of the minority class to that of the majority class. Most classification techniques such as decision tree, discriminant analysis and neural networks assume that the training samples are evenly distributed amongst different classes. However, in real-world applications, the ratio of minority to majority samples can be as low as 1 to 100, 1 to 1000, or 1 to 10,000 (Chawla et al., 2004). Hence, the standard classifiers are affected by the prevalent classes and tend to ignore or treat the small classes as noise. Weiss and Provost investigated the relationship between the imbalance ratio of training samples in each class and classifier performances, in terms of overall accuracy and area under the ROC curve (AUC) (Weiss and Provost, 2003). They used a decision-tree classifier and tested it on a number of data sets from the UCI Repository (Asuncion and Newman, 2007). Their experimental results indicated that the ratio of samples in each class depends on the evaluation metrics used. When the performance is measured using classification accuracy, the best ratio is near the natural ratio; on the other hand, when the AUC measure is used, the best ratio is near the balanced ratio. Visa and Ralescu also reported similar results using fuzzy classifiers (Visa and Ralescu, 2005). However, we should note that the imbalance ratio between classes is not the only factor that reduces classification performance; other factors such as training size and concept complexity also affect performance.

In tasks that involve learning a concept or detecting an event, data imbalance can appear within a single class. The within-class imbalance problem occurs when a class consists of several sub-clusters or sub-concepts and these sub-clusters do not have the same number of samples (Japkowicz, 2001). The within-class and between-class imbalances together are known as the problem of small disjuncts (Holte et al., 1989), in which classifiers are biased towards recognizing large disjuncts correctly, but overfitting and misclassifying samples represented by small disjucts. In most classification tasks, the presence of within-class imbalance is implicit. It is known to have negative effects on the performance of standard classifiers and increases the complexity of concept learning (Yoon and Kwek, 2007). However, most existing methods for class imbalance focus mainly on rectifying the between-class imbalance, and ignore the case where imbalance occurs within each class.

2.2 Lack of data

One of the primary problem when learning with imbalanced data sets is the associated lack of data where the number of samples is small (Weiss, 2004). In a given classification task, the size of data set has an important role in building a good classifier. Lack of examples, therefore, makes it difficult to uncover regularities within the small classes. Fig. 1 illustrates an example of the problem that can be caused by lack of data. Fig. 1 (a) shows the decision boundary (dashed line) obtained when using sufficient data for training, whereas Fig. 1 (b) shows the result when using a small number of samples. When there is sufficient data, the estimated decision boundary (dashed line) approximates well the true decision boundary (solid line); whereas, if there is a lack of data, the estimated decision boundary can be very far from the true boundary. In fact, it has been shown that as the size of training set increases, the error rate caused by imbalanced training data decreases (Japkowicz and Stephen, 2002). Weiss and Provost conducted experiments on twenty six data sets, taken from the UCI repository, to investigate the relationship between the degree of class imbalance and training set sizes (Weiss and Provost, 2003). They showed that when more training data become available, the classifiers are less sensitive to the level of imbalance between classes. This suggests that with sufficient amount of training data, the classification system may not be affected by high imbalance ratio.


(a)

(b)

Fig. 1. The effect of lack of data on class imbalance problem; the solid line represents the true decision boundary and dashed line represents the estimated decision boundary.

2.3 Concept complexity

Concept complexity is an important factor in a classifier ability to deal with imbalanced problems. Concept complexity in data corresponds to the level of separability of classes within the data. Japkowicz and Stephen reported that for simple data sets that are linearly separable, classifier performances are not susceptible to any amount of imbalance (Japkowicz and Stephen, 2002). Indeed, as the degree of data complexity increases, the class imbalance factor starts impacting the classifier generalization ability. High complexity refers to inseparable data sets with highly overlapped classes, complex boundaries and high noise level. When samples of different classes overlap in the feature space, finding the optimum class boundary becomes hard. In fact, most accuracy-driven algorithms bias toward the

prevalent class. That is, they improve the overall accuracy by assigning the overlapped area to the majority class, and ignore or treat the small class as noise (Murphey et al., 2007).
The class imbalance problem is more significant when the data sets have a high level of noise. Noise in data sets can emerge from various sources, such as data samples are poorly acquired or incorrectly labeled, or extracted features are not sufficient for classification. It is known that noisy data affect many machine learning algorithms; however, Weiss showed that noise has even more serious impact when learning with imbalanced data (Weiss, 2004). The problem occurs when samples from the small class are mistakenly included in the training data for the majority class, and vice versa. For the prevalent class, noise samples have less impact on the learning process. In contrast, for the small class it takes only a few noise samples to influence the learned sub-concept. For a given data set that is complex and imbalanced, the challenge is how to train a classifier that correctly recognizes samples of different classes with high accuracy.

3. Existing approaches

To address the problems associated with imbalanced data sets, many studies have been conducted to improve traditional learning algorithms. In this section, we review various approaches, which have been proposed both at the data level, such as re-sampling and combinations, and at the algorithmic level, such as recognition-based approach, cost-sensitive learning and boosting.

3.1 Recognition-based approach

As discussed previously, certain discriminative learners such as neural networks, decision trees, support vector machines and fuzzy classifiers tend to recognize the majority class instances since they are trained to achieve the overall accuracy, to which the minority class contributes very little. A recognition-based or one-class learning approach is another alternative solution where the classifier is modeled on the examples of the target class (the small class) in the absence of examples of the non-target class. One of the early systems that utilize this recognition-based approach was proposed in (Japkowicz et al., 1995). It uses neural networks and attempts to learn only from the target class examples and thus recognizing the target concept, rather than to differentiating between majority and minority instances of a concept. One-class learning approach is also applied to autoencoder-based classifiers (Eavis and Japkowicz, 2000), SVMs (Raskutti and Kowalczyk, 2004), and ensemble one-class classifiers (Spinosa and Carvalho, 2005). Here similar patterns from positive instances of a concept are learnt, classifiers are then presented with unseen samples, classification is accomplished by imposing a threshold on the similarity value. A too high threshold will result in misclassifying positive samples, while a too low threshold will include more negative samples. Since threshold draws the boundaries that separate the two classes, choosing an effective threshold is crucial in one-class learning. Japkowicz shows that one-class learning approach to solving the imbalanced class problem is better than discriminative (two-class learning) approach (Japkowicz, 2001). However, recognition-based approach cannot apply to many machine learning algorithms such as, decision tree, Naive Bayes, and associative classifications. These classifiers are not constructed from only samples of one-class.

3.2 Cost-sensitive learning

In many applications such as medical diagnosis, fraud detection, intrusion prevention and risk management, the primary interest is in fact in the small classes. In these applications, it is not only the data distributions that are skewed, but so are the misclassification costs. Most classical learning algorithms assume that all misclassification errors cost equally, and ignore the difference between types of misclassification errors. One practical solution to this problem is to use cost-sensitive learning methods (Elkan, 2001).

A cost-sensitive learning technique takes costs, such as misclassification cost, into consideration during model construction and produces a classifier that has the lowest cost. Let $C(i, j)$ denote the cost of estimating an example from class i as class j. In a two class problem, $C(+, -)$ signifies the cost of misclassifying a positive sample as the negative sample, and $C(-, +)$ denotes the cost of the contrary case. Cost-sensitive learning methods take advantage of the fact that it is more expensive to misclassify a true positive instance than a true negative instance, that is $C(+, -) > C(-, +)$. For a two-class problem, a cost-sensitive learning method assigns a greater cost to false negatives than to false positives, hence resulting in a performance improvement with respect to the positive class.

Existing cost-sensitive learning for dealing with imbalanced data sets can be divided into two different categories. The first category consists of learning algorithms that are designed to optimize a cost-sensitive function directly. One example is cost-sensitive decision tree, proposed in (Ling and Li, 2004) that directly takes costs into model building. The misclassification costs are used to choose the best attribute as a root of the tree. The second category is a collection of existing cost-insensitive learning algorithms that are converted into cost-sensitive ones. This category, also known as cost-sensitive meta-learning, can be further divided into sampling, weighting, thresholding, and ensemble learning. Methods in the weighting group (Alejo et al., 2007), convert sample-dependent costs into sample weights; in other words, they assign heavier weights to the minority training instances. Different weighting strategies have been reported: Nguyen and Ho proposed to weight samples of the minority class based on the local data distributions (Nguyen and Ho, 2005), and others suggested to weight training samples based on posterior probability (Tao et al., 2005). Zhou and Lui conducted a rigorous comparison on the effects of oversampling, and under-sampling, threshold-moving and ensemble classifiers in training cost-sensitive neural networks (Zhou and Liu, 2006). They find that in training cost-sensitive neural networks, threshold-moving and ensemble learning are relatively good choices in both two-class and multi-class tasks. However, like many other solutions, they also have some drawbacks. Cost-sensitive learning approach assumes the misclassification costs are known. In practice, specific cost information is often unavailable because costs often depend on a number of factors that are not easily compared. Moreover, Weiss found that cost-sensitive classifiers may lead to over fitting during training (Weiss, 2004).

3.3 Sampling

One of the common approaches to class imbalance problem is sampling. The key idea is to pre-process training data to minimize any discrepancy between the classes. In other words, sampling methods modify the prior distributions of the majority and minority class in the training set to obtain a more balanced number of instances in each class.

Basic sampling methods. The two basic methods of reducing class imbalance in training data are under-sampling and over-sampling. Under-sampling extracts a smaller set of majority instances while preserving all the minority instances. Under-sampling is suitable for large-scale applications where the number of majority samples is very large and lessening the training instances reduces the training time and storage. However, a drawback with under-sampling is that discarding instances may lead to loss of informative majority class instances and degrade classifier performance.

In contrast, over-sampling increases the number of minority instances by replicating them (Chawla et al., 2002, Japkowicz and Stephen, 2002). The advantage is that no information is lost, all instances are employed. However, over-sampling also has its own drawbacks. By creating additional training instances, over-sampling leads to a higher computational cost. Moreover, if some of the small class samples contain labeling errors, adding them will actually deteriorate the classification performance on the small class (Chawla et al., 2004). Lastly, over-sampling duplicates majority instances rather than introducing new data, so it does not address the underlying lack of data.

Despite the fact that sampling methods are widely used for tackling class imbalance problems, there is no established way to determine the suitable class distribution for a given data set (Weiss and Provost, 2003). The optimal class distribution is dependent on the performance measures and varies from one dataset to another. However, effectively sampling training instances can improve and overcome some of the weaknesses discussed above. Next, we describe some of the advanced sampling methods that are reported to be superior to random over-sampling and under-sampling.

Advanced sampling methods

In advanced sampling, instances are added or removed adaptively. Advanced sampling methods may also combine under-sampling and over-sampling techniques. One of the popular over-sampling approaches is SMOTE (Synthetic Minority Over-sampling TEchnique), which attempts to add information to the training set by introducing new, non-replicated minority class examples (Chawla et al., 2002). Generative over-sampling, proposed in (Liu et al., 2007), is a variation of SMOTE. It creates new data points by learning from available training data. In other words, a probability distribution is selected to model the available minority class examples. Then new data points are generated from this model. A drawback of this method is that when the number of examples of the minority class is not adequate, the probability distribution estimates that model the actual data distributions may not be accurate.

In an under-sampling scheme, instead of eliminating instances randomly, Yu and co-workers proposed a different method to re-sampling the majority class instances (Yu et al., 2007). The authors proposed to use vector quantization, which is a lossy compression method, on the majority class to build a set of representative local models and use them for training the SVM. Another informative re-sampling technique is cluster-based under-sampling (Yen and Lee, 2009). In this technique, clustering is employed for selecting the representative training samples to improve the predictive accuracy for the minority class. Yen and Lee reported that this approach empirically outperforms other under-sampling techniques. Yoon and Kwek also proposed to use clustering to reduce the imbalanced ratio, called Class Purity Maximization (CPM) (Yoon and Kwek, 2005). CPM partitions the data space into clusters, and filters out regions in the data space that consist of high majority class purity. Hence, only regions containing minority samples are used to build a predictive model. CPM reduces the imbalance ratio and makes the learning task more tractable.

Active learning is also another solution to class imbalance problem. Ertekin et al. proposed using active learning to select informative samples of the training set (Ertekin et al., 2007). Similarly to re-sampling, active learning query technique creates balanced training sets at the early stages of the learning process. This technique focuses on query instances near the classification boundary rather than selecting randomly any instance. Active learning gives the learners the ability to select examples adaptively. Furthermore, the risk of losing important information is reduced, compared with the under-sampling approach. Active learning does not create extra data as in oversampling.

3.4 Ensemble-learning methods

Another alternative solution for the class imbalance problem is ensemble-learning, in which multiple classifiers are trained from the original data and their predictions are combined to classify new instances. Boosting (Freund and Schapire 1996) and bagging (Breiman, 1996) are two widely known ensemble-based approaches. Boosting algorithms, such as AdaBoost (Leskovec and Shawe-Taylor, 2003), improve performance of weak classifiers by forcing the learners to focus more on the difficult examples. Boosting algorithms have been adapted to address the problem with small classes. At each boosting iteration, the distribution of training data is altered by updating the weight associated with each sample. Examples of algorithms that use boosting to address the class imbalance problems are SMOTEBoost (Chawla et al., 2002), DataBoost-IM (Guo and Viktor, 2004), and cost-sensitive booting (Sun et al., 2007). Both DataBoost-IM and SMOTEBoost improve boosting by combining data generation and boosting procedures. To avoid over fitting, SMOTEBoost alters the data distribution by adding new minority class samples using the SMOTE algorithm (Chawla et al., 2002).

DataBoot-IM, proposed by (Guo and Viktor, 2004), generates data to balance not only the class distribution but also the total weight within the class. Through experiments on seventeen data sets, the authors showed that DataBoost method does not sacrifice one class over the other but improve the predictive accuracies of both majority and minority classes. A cost-sensitive booting algorithm for classification of imbalanced data was proposed in (Sun et al., 2007), in which misclassification costs are integrated into AdaBoost learning. The AdaBoost weight-update strategy is altered so that the weights of misclassified samples from the small class increase at a higher rate compared to those of the prevalent class. The weights of correctly classified samples from the small class reduce at a lower rate, compared to those from the prevalent class.

Bagging is one of the ensemble-based meta-learning algorithms. Most current bagging methods use a similar learning procedure: re-sampling subsets from a given training set, building multiple base classifiers on those subsets, and combining their predictions to make final prediction (Breiman, 1996). Several algorithms based on a variety of sampling strategies are proposed, for example roughly balanced (RB) bagging (Hido and Kashima, 2008), underBagging (Liu et al., 2006), overBagging and SMOTEBagging (Wang and Yao, 2009). In underBagging, each subset from the training set is created by under-sampling the majority classes randomly to build a classifier. RB bagging is a variation to underBagging: it makes use of both minority samples and under-sampling majority samples. However, RB bagging uses an effective under-sampling technique based on negative binomial distributions. In comparison, overBagging forms subsets simply by over-sampling the minority classes randomly. SMOTEBagging (Wang and Yao, 2009) differs from underBagging and overBagging in that it involves generating synthetic instances during subset construction. The main advantage of bagging is that it maintains the class distribution of the training set on which bagging is applied. However, bagging relies on a simple strategy that is very limited for dealing with class imbalance problem, except from changing the bag size and sampling step.

4. Classifier performance measures

Evaluation metrics play an important role in machine learning. They are used to evaluate and guide the learning algorithms. If the choices of metrics do not value the minority class, then the learning algorithms will not be able to handle the imbalance problem very well. The commonly used metric for these purposes is the overall classification rate (i.e. accuracy). However, on an imbalanced data set, the overall classification rate is no longer a suitable metric, since the small class has less effect on accuracy as compared to the prevalent class. Weiss and Provost conducted an empirical study on twenty-six data sets, and showed that using the overall accuracy measure leads to poor performance for the minority class (Weiss and Provost, 2003). Therefore, other metrics have been developed to assess classifiers performance for imbalanced data sets. A variety of common metrics are defined based on the *confusion matrix* (also called a contingency table). A two-by-two confusion matrix is shown in Table 1.

		True class	
		Positive	Negative
Prediction class	Positive	True Positives (TP)	False Positives (FP)
	Negative	False Negatives (FN)	True Negatives (TN)

Table 1. Confusion matrix for a two-class classification task.

Among the various evaluation criteria, the measures that are most relevant to imbalanced data are precision, recall, F-measure, sensitivity, specificity, geometric mean, ROC curve, AUC, and precision-recall curve. These metrics share a commonality in that they are all class-independent measures.

Precision, recall and F-measure. These metrics arise from the fields of information retrieval. They are used when performance of positive class (the minority class) is considered, since both precision and recall are defined with respect to the positive class.

- *Precision* of a classifier is the percentage of positive predictions made by the classifier that are correct.

$$\text{Precison} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

- *Recall* is the percentage of true positive patterns that are correctly detected by the classifier.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

- *F-measure* is defined as the harmonic mean of recall and precision (Fawcett, 2006). A high F-measure value signifies a high value for both precision and recall.

$$\text{F-measure} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$

Sensitivity, Specificity and Geometric mean. These measures are utilized when performance of both classes is concerned and expected to be high simultaneously. The geometric mean (G-mean) metric was suggested in (Kubat and Matwin, 1997) and has been used by several researchers for evaluating classifiers on imbalanced data sets (Ertekin et al., 2007, Karagiannopoulos et al., 2007, Su and Hsiao, 2007). G-mean indicates the balance between classification performances on the majority and minority class. This metric takes into account both the *sensitivity*, (the accuracy on the positive examples) and the *specificity* (the accuracy on the negative examples):

$$\text{Sensitivity} = \text{Recall}$$

$$\text{Specificity} = 1 - \frac{\text{FP}}{\text{Total Negatives}}$$

$$\text{G-means} = \sqrt{\text{Sensitivity} \times \text{Specificity}}$$

ROC and AUC. The receiver operating characteristic (ROC) and the area under the ROC curve (AUC) are the two most common measures for assessing the overall classification performance (Weiss, 2004). The ROC is a graph showing the relationship between benefits (correct detection rate or true positive rate) and costs (false detection rate or false positive rate) as the decision threshold varies. The ROC curve shows that for any classifier, the true positive rate cannot increase without also increasing the false positive rate. The true positive rate is the same as recall, and the false detection rate is equal to

$$\text{FDR} = \frac{\text{FP}}{\text{Total Negatives}} \quad .$$

A ROC curve gives a visual indication if a classifier is superior to another classifier, over a wide range of operating points. However, a single metric is sometimes preferred when comparing different classifiers. The area under the ROC curve (AUC) is employed to summarize the performance of a classifier into a single metric. The AUC does not place more weight on one class over the another. The larger the AUC, the better is the classifier performance.

Precision-Recall (PR) curve. Precision-recall curve is used in information retrieval in a similar fashion as the ROC curve. The PR curve depicts the relationship between precision and recall as the classification threshold varies.

Apart from the above evaluation metrics, a number of new evaluation metrics have been proposed to take small class size into account when evaluating the end result. For imbalanced data sets, not only the class distribution but also the misclassification costs are skewed. Hence, Weng and Poon introduced a new metric, weighted-AUC, that can take into account the cost bias when evaluating classifier performance (Weng and Poon, 2008). Some other authors had suggested that the ROC curve alone is not sufficient, and the effect of imbalance class distribution should be analyzed when comparing different learning algorithms (Landgrebe et al., 2004). Therefore, they proposed to use costs that are dependent on class distribution such as positive fraction together with ROC curve. The positive fraction is defined as the fraction of objects that are positively labeled.

Other metrics such as rank metrics, rank prop and soft ranks are proposed for training and model selection (Caruanan, 2000). These metrics prevent learners from mainly optimizing classification performance on the dominant class. Comparisons of several evaluation metrics were conducted by Liu and Shriberg and found that a single measure such as precision, recall, F-measure, sensitivity, specificity, G-mean or AUC provide limited information, since each measure is designed to assess one particular property or decision point (Liu and Shriberg, 2007). Hence, to analyze and compare learning algorithms involving class imbalance, it is necessary to combine different metrics and performance curves such as ROC and PR.

5. The proposed learning approach for imbalanced data set

In this chapter, we introduce a new learning approach that aim to tackling the class imbalance problem. In our approach, we first propose a new under-sampling method based on clustering. Here, a clustering technique is employed to partition the training instances of each class independently into a smaller set of training prototype patterns. Then a weight is assigned to each training prototype to address the class imbalance problem.. The weighting strategy is introduced in the cost function such that the class distributions become roughly even. In the extreme imbalance cases, where the number of minority instances is small, we apply unsupervised learning to resample only the majority instances, and select cluster centers as prototype samples, and keep all the small class samples.

The proposed learning approach, which combines unsupervised and supervised learning to deal with the class imbalance problem, can be applied on any classifier model. In this chapter, we apply the proposed learning approach to train feed-forward neural networks, which is a classification model that has been used extensively in pattern recognition. Based on the proposed learning approach, we derive and analyze the resilient back-propagation training algorithm for feed-forward neural networks. The algorithm is implemented and tested on some benchmark data sets.

5.1 Under-sampling based on clustering

Suppose that a multi-layer feed-forward neural network is to be trained on a given training data set D of size M

$$D_M = \{(x_i, \mathbf{d}_i) | x_i \in \Re^N, i = 1, 2, \ldots, M\}$$

where x_i is the *i*-th input pattern and $\mathbf{d}_i$ is the corresponding desired output vector. Let $\mathbf{w}$ be a vector consisting of all free network parameters, including weights and biases. The objective of supervised learning is to find a vector $\mathbf{w}_o$ that minimizes a cost function. A common objective function is the *mean square error* (MSE), defined as

$$E(\mathbf{w}) = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (y_{ij} - d_{ij})^2, \tag{1}$$

where *N* is the number of neurons in the output layer, and y_{ij} is the network output.

When the numbers of training instances of different classes are uneven, the contribution from each class to the objective function is not equal. In a two-class problem, the majority class has a significant effect in the optimization process. Hence, we propose a more efficient algorithm for training feed-forward neural networks. In this approach, a pre-processing step is introduced to obtain a more balanced number of samples in each class. To this end, unsupervised *clustering* is applied to training samples to extract cluster centers that yield a compact representation of the majority classes.

Here, clustering is applied independently to each class. Therefore, each cluster contains samples from the same class, and each class can have several clusters. We deal with imbalanced data sets by assigning the same number of clusters to each class. When the number of minority samples is small, we only apply unsupervised clustering to resample the majority instances, and retain all the minority samples. After clustering, the data set is reduced to *K* exemplars; each is represented by a cluster *centroid* c_k and cluster *size* z_k. Here, the cluster size z_k is simply the number of training samples in the cluster. Next, we present the resilient back-propagation training algorithm that integrates the cluster centroids and sizes into the learning rule.

5.2 Modified training algorithm

In the supervised learning stage, training samples are replaced by a set of cluster centroids, which is then presented to the network along with the target outputs. To compensate for the information lost during the clustering process, weights for each class are introduced in the cost function, which is modified as follows.

$$E(\mathbf{w}) = \frac{1}{N} \sum_{k=1}^{K} \sum_{i=1}^{N} (y_{ki} - d_{ki})^2 \times p_k \, , \tag{2}$$

where d_{ki} is the *i*-th element of the target or desired output vector $\mathbf{d}_k$, and p_k is the cluster weight. The cluster weight is defined as follow,

$$p_k = \frac{z_k}{\sum_{i=1}^{N_{cl}} w_i \gamma_{ki}}, \qquad k = 1, ..., K \tag{3}$$

where N_{cl} is number of classes in the training set, w_i is the size of class *i*, and γ_{ki} is the degree of membership of cluster *k* in class *i*:

$$\gamma_{ki} = \begin{cases} 1 & \text{if } c_k \in \text{class } i \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Numerous optimization algorithms for minimizing *E* can be derived to train feed-forward neural networks, such as gradient descent (GD), gradient descent with momentum and variable learning rate (GDMV), resilient back-propagation (RPROP), and Levenberg-Marwquardt (LM). We have implemented and analyzed these algorithms based on our proposed learning approach (Nguyen et al., 2008). In this chapter, we only perform the analysis on the resilient back-propagation method (RPROP) to train the feed-forward neural network, refer to (Nguyen et al., 2008) for other training methods. The RPROP training algorithm updates network weights and biases according to $w(t+1) = w(t) + \Delta w(t)$. Because details of the standard RPROP algorithms can be found in (Riedmiller and Braun, 1993), we only summarize its main characteristics here.

Resilient back-propagation: Weight update depends only on the sign of the gradient

$$\Delta w_i(t) = -\text{sign}\left\{ \frac{\partial E}{\partial w_i}(t) \right\} \times \Delta_i(t), \tag{1}$$

where $\Delta_i(t)$ is on adaptive step specific to weight $w_i(t)$.

6. Experiments

In this section, we apply the proposed learning approach to four benchmark problems, taken from UCI database repository (Asuncion and Newman, 2007). The benchmarks used are the liver disorder, hepatitis, Wisconsin diagnostic breast cancer, and Pima Indian diabetes data sets. These data sets are summarized in Table 2. Our aim is to study the generalization capability of the proposed approach in tackling the class imbalance problem, compared to the standard approach for training feed-forward neural networks.

The comparison is based on a five-fold cross validation in the classification tasks. For each fold, the data set is partitioned into 60% for training set, 20% for validation set and 20% for test set. Several networks are trained and the best performing network on the validation set is selected to be evaluated on the test set. The average classification rate on the test set, over the five folds, is used as an estimate of generalization performance. Since the overall classification rate is not the most suitable tool for imbalanced data, other measures are also used, including the geometric mean and F-measure.

Data sets	**Size**	**Features**	**Class distribution**	**Imbalanced ratio (Majority/Minority)**
Liver	345	6	145/200	1.38
Hepatitis	155	19	32/123	3.84
Pima diabetes	768	8	268/500	1.87
Wisconsin Breast cancer	699	10	241/458	1.90

Table 2. Summary of data sets used in the experiments.

The comparison results of different training algorithms over all data sets are shown in Table 3. The modified training (Mod-RPROP) and the standard training (RPROP) achieve almost similar classification rates (CRs). For examples, in the hepatitis data set, CRs of RPROP and Mod-RPROP are 92.00% and 92.67%, respectively. However, the modified algorithm has higher values of G-mean and F-measure than its counterpart. In the hepatitis data set, the G-mean values of RPROP and Mod-RPROP are 90.80% and 91.65%, and the F-measure value of RPROP and Mod-RPROP are 80.48% and 82.57%, respectively. This finding suggests that the modified training algorithm exhibits good classification rates for all classes.

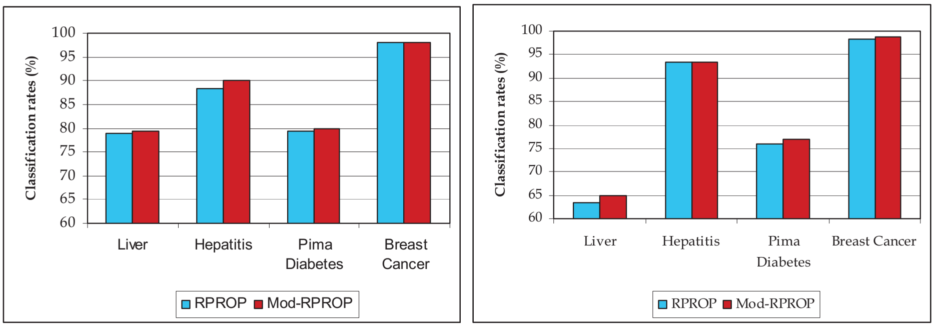
Fig. 2 shows the classification rates of each class over all data sets. The classification rates of positive class (or the sensitivity) as well as classification rates of negative class (or the specificity) are increased. For example, the sensitivity of Mod-RPROP increases by 1.38% in liver data set, and 1.13% in Pima data set, compared to the standard RPROP. In terms of specificity, the Mod-RPROP maintains the accuracies and in some occasions improves them.

Data sets	Overall CR		F-measure		G-means	
	RPROP	Mod-RPROP	RPROP	Mod-RPROP	RPROP	Mod-RPROP
Liver	73.91	74.78	66.53	67.66	70.80	71.79
Hepatitis	92.00	92.67	80.48	82.57	90.80	91.65
Pima Diabetes	80.65	82.22	70.70	72.11	77.60	78.38
Breast Cancer	98.13	98.27	97.35	97.56	98.18	98.39

Table 3. Comparision of standard and reduced training algorithms on benchmark data sets.

Data sets	Specificity		Sensitivity	
	RPROP	Mod-RPROP	RPROP	Mod-RPROP
Liver	79.00	79.50	63.45	64.83
Hepatitis	88.33	90.00	93.33	93.33
Pima Diabetes	79.40	79.80	75.85	76.98
Breast Cancer	98.02	98.02	98.33	98.75

Table 4. Classification rates of each class on benchmark data sets.



(a) (b)

Fig. 2. Comparison of the standard RPROP and Mod-RPROP trainging algorithms on four data sets in terms of (a) classification rates of negative class and (b) classification rates of positive class.

7. Conclusion

In this chapter, we discussed the problems that arise when learning with imbalanced data sets, including between classes imbalance, within-class imbalance, the lack of data, and concept complexity. Then we reviewed various methods and techniques that address the class imbalance problems, both at the data level (re-sampling and combinations) and the algorithmic level (recognition-based approach, cost-sensitive learning and boosting). We also discussed a number of evaluation metrics that have been developed to assess classifier performance on imbalanced data sets. Then we presented a new approach that combines unsupervised clustering and supervised learning to handle imbalanced data set and applied this learning approach for training feed-forward neural networks. The proposed approach can be applied to existing training algorithms. Experimental results show that the proposed approach can effectively improve the classification accuracy of the minority classes, while maintaining the overall classification performance.

8. References

Alejo, R., García, V., Sotoca, J., Mollineda, R. & Sánchez, J. (2007). Improving the Performance of the RBF Neural Networks Trained with Imbalanced Samples, *In Proceedings of Computational and Ambient Intelligence, 9th International Work-Conference on Artificial Neural Networks*, pp. 162-169, San Sebastian, Spain, 2007.

Asuncion, A. & Newman, D. J. (2007) UCI Machine Learning Repository.

Breiman, L. (1996) Bagging Predictors. *Machine learning,* 24**,** 123-140.

Caruanan, R. (2000). Learning from Imbalanced Data: Rank Metrics and Extra Tasks, *The AAAI Workshop on Learning from Imbalanced Data Sets*, pp. 51-57, 2000.

Chawla, N., Japkowicz, N. & Kotez, A. (2004) Editorial: Special Issue on Learning from Imbalanced Data. *Sigkdd Explorations,* 6**,** 1-6.

Chawla, N. V., Bowyer, K., Hall, L. & Kegelmeyer, W. P. (2002) SMOTE: Synthetic Minority Over-sampling TEchnique. *Artificial Intelligence Research,* 16**,** 321-357.

Chong, E. K. P. & Zak, S. H. (1996) *An Introduction to Optimization,* New York, John Wiley and Sons, Inc.

Eavis, T. & Japkowicz, N. (2000). A Recognition-Based Alternative to Discrimination-Based Multi-layer Perceptrons, *The 13th Biennial Conference of the Canadian Society on Computational Studies of Intelligence*, pp. 280-292, London, UK, 2000, Springer-Verlag.

Elkan, C. (2001). The Foundations of Cost-sensitive Learning, *In Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pp. 73-978, 2001.

Ertekin, S., Huang, J., Bottou, L. & Giles, C. L. (2007). Learning on the Border: Active Learning in Imbalanced Data Classification, *In Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management*, pp. 127-136, Lisbon, Portugal, 2007, ACM Press.

Fawcett, T. (2006) An Introduction to ROC Analysis. *Pattern Recognition Letters,* 27**,** 861-874.

Guo, H. & Viktor, H. L. (2004) Learning from Imbalanced Data Dets with Boosting and Data Generation: the DataBoost-IM Approach. *ACM SIGKDD Explorations Newsletter,* 6**,** 30-39.

Hagan, M. T. & Menhaj, M. B. (1994) Training Feedforward Networks with the Marquardt Algorithm. *IEEE Transactions on Neural Networks,* 5**,** 989-993.

Hido, S. & Kashima, H. (2008). Roughly Balanced Bagging for Imbalanced Data., *In Proceedings of the SIAM International Conference on Data Mining*, pp. 143-152, Atlanta, Georgia, USA, 2008.

Holte, R., Acker, L. & Porter, B. (1989). Concept Learning and the Problem of Small Disjuncts, *In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pp., Austin, TX, USA, 1989, Morgan Kaufmann.

Japkowicz, N. (2001). Concept-Learning in the Presence of Between-Class and Within-Class Imbalances, *In Proceedings of the 14th Biennial Conference of the Canadian Society on Computational Studies of Intelligence*, pp., London, UK, 2001, Springer-Verlag.

Japkowicz, N., Mayers, C. & Gluck, M. (1995). A Novelty Detection Approach to Classification *In Proceedings of the Fourteenth Joint Conference on Artificial Intelligence*, pp. 518-523, 1995.

Japkowicz, N. & Stephen, S. (2002) The Class Imbalance Problem: A Systematic Study. *Intelligent Data Analysis,* 6**,** 429-449.

Karagiannopoulos, M. G., Anyfantis, D. S., Kotsiantis, S. B. & Pintelas, P. E. (2007). Local Cost Sensitive Learning for Handling Imbalanced Data Sets, *In Proceedings of Mediterranean Conference on Control and Automation*, pp. 1-6, 2007.

Kubat, M. & Matwin, S. (1997). Addressing the Curse of Imbalanced Training Set: One-sides Selection, *In Proceedings of the Fourteenth International Conference on Machine Learning*, pp. 179-186, 1997, Morgan Kaufmann.

Landgrebe, T., Pacli'k, P., Tax, D. J. M., Verzakov, S. & Duin, R. P. W. (2004). Cost-based Classifier Evaluation for Imbalanced Problems, *In Proceedings of The 10th International Workshop on Structural and Syntactic Pattern Recognition and the 5th International Workshop on Statistical Techniques in Pattern Recognition*, pp. 762-770, Lisbon, Portugal, 2004, Springer Verlag, Berlin.

Leskovec, J. & Shawe-Taylor, J. (2003). Linear Programming Boosting for Uneven Datasets, *In Proceedings of The twentieth International Conference on Machine Learning* pp. 456-463, 2003, AAI Press.

Ling, C. X. & Li, C. (2004). Decision Trees with Minimal Costs, *In Proceedings of the Twenty-first International Conference on Machine Learning*, pp. 69, Banff, Alberta, Canada, 2004, ACM.

Liu, A., Ghosh, J. & Martin, C. (2007). Generative Oversampling for Mining Imbalanced Datasets, *In Proceedings of The 2007 International Conference on Data Mining*, pp., Las Vegas, Nevada, USA, 2007, CSREA Press.

Liu, Y., Chawla, N. V., Harper, M., Shriberg, E. & Stolcke , A. (2006) A Study in Machine Learning from Imbalanced Data for Sentence Boundary Detection in Speech. *Computer Speech and Language,* 20**,** 468-494.

Liu, Y. & Shriberg, E. (2007). Comparing Evaluation Metrics for Sentence Boundary Detection, *In Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 185-188, 2007.

Murphey, Y. L., Wang, H., Ou, G. & Feldkamp, L. (2007). OAHO: and Effective Algorithm for Multi-class Learning from Imbalanced Data, *Proceedings of the International Joint Conference on Neural Networks*, pp. 406-411, Orlando, Florida, USA, 2007, IEEE.

Nguyen, C. & Ho, T. (2005). An Imbalanced Data Rule Learner, *In Proceedings of The 9th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pp. 617-624, Porto, Portugal, 2005.

Nguyen, G. H., Bouzerdoum, A. & Phung, S. L. (2008). A Supervised Learning Approach for Imbalanced Data Sets, *In Proceeding of International Conference on Pattern Recognition*, pp. 1-4, Florida, USA, 2008.

Raskutti, B. & Kowalczyk, A. (2004) Extreme Re-balancing for SVMs: a Case Study. *ACM Sigkdd Explorations Newsletter,* 6**,** 60-69.

Riedmiller, M. & Braun, H. (1993). A Direct Adaptive Method of Faster Backpropagation Learning: The RPROP Algorithm, *IEEE International Conference on Neural Networks*, pp. 586-591, San Francisco, 1993.

Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1986) Learning Internal Representations by Error Propagation. *Parallel Distributed Processing: Explorations in The Microstructure of Cognition,* I**,** 318 – 362.

Spinosa, E. J. & Carvalho, A. (2005) Combining one-class classifiers for robust novelty detection in gene expression. *Advances in bioinformatics and computational biology,* 3549**,** 54-64.

Su, C. & Hsiao, Y. (2007) An Evaluation of Robustness of MTS for Imbalanced Data. *IEEE Transaction on Knowledge and Data Engineering,* 19**,** 1321-1332.

Sun, Y., Kamel, M. S., Wong, A. K. C. & Wang, Y. (2007) Cost-sensitive Boosting for Classification of Imbalanced Data. *Pattern Recognition,* 40**,** 3358-3378.

Tao, Q., Gao Wei Wu, Fei Yue Wang & Wang, J. (2005) Posterior Probability Support Vector Machnies for Unbalanced Data. *IEEE Transaction on Neural Networks,* 16**,** 1561-1573.

Visa, S. & Ralescu, A. (2005). The Effect of Imbalanced Data Class Distribution on Fuzzy Classifiers - Experimental Study, *In Proceedings of the IEEE Conference on Fuzzy Systems*, pp. 749-754, 2005.

Wang, S. & Yao, X. (2009). Diversity Analysis on Imbalanced Data Sets by Using Ensemble Models, *In Proceedings of The IEEE Symposium on Computational Intelligence and Data Mining*, pp. 324-331, 2009, IEEE.

Weiss, G. M. (2004) Mining with Rarity: a Unifying Framework. *SIGKDD Explorations and Newsletters,* 6**,** 7-19.

Weiss, G. M. & 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.

Weng, C. G. & Poon, J. (2008). A New Evaluation Measure for Imbalanced Datasets, *In Proceedings of The Seventh Australasian Data Mining Conference* pp. 27-32, Glenelg, South Australia, 2008, ACS.

Yen, S.-J. & Lee, Y.-S. (2009) Cluster-based Under-sampling Approaches for Imbalanced Data Distributions. *Expert Systems with Applications,* 36**,** 5718-5727.

Yoon, K. & Kwek, S. (2005). An Unsupervised Learning Approach to Resolving the Data Imbalanced Issue in Supervised Learning Problems in Functional Genomics, *In Proceedings of The Fifth International Conference on Hybrid Intelligent Systems*, pp. 303-308, Washington, DC, USA, 2005, IEEE Computer Society.

Yoon, K. & Kwek, S. (2007) A Data Reduction Approach for Resolving The Imbalanced Data Issue in Functional Genomics. *Neural Computing and Applications,* 16**,** 295-306.

Yu, T., Jan, T., Simoff, S. & Debenham, J. (2007) A Hierarchical VQSVM for Imbalanced Data Sets. *In Proceedings of The International Joint Conference on Neural Networks.*

Zhou, Z.-H. & Liu, X.-Y. (2006) Training Cost-sensitive Neural Networks with Methods Addressing The Class Imbalance Problem. *IEEE Transaction on Knowledge and Dada Engineering,* 18**,** 63-77.