ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 9, September 2013

Conversion of Imbalanced Data Into A Stream Using SMOTE Algorithm

A.Vanitha[1], S.Niraimathi[2]

Research scholar, Computer science (Aided), NGM College, Coimbatore, India [1]

Assistant professor, Computer science, NGM College, Coimbatore, India [2]

Abstract: Machine learning approach has got major importance when distribution of data is unknown. Classification of data from the data set causes some problem when distribution of data is unknown. Characterization of raw data relates to whether the data can take on only discrete values or whether the data is continuous. In real world application data drawn from non-stationary distribution, causes the problem of "concept drift" or "non-stationary learning". Drifting of dataset is often associated with online learning scenario. The goal of intelligent machine learning algorithms is to be able to address a wide spectrum of real world scenarios, then the need for a general framework for learning from, and adapting to, a non-stationary environment that may introduce imbalanced data can be hardly overstated. This paper focus on imbalanced data that results in unequal representation of classes in a pattern recognition problem. There are typically two types on class in an imbalanced pattern recognition problem, majority (negative) and minority (positive).

Keywords: Data Stream, Class Imbalance, Imbalanced Domains, Data Level Methods, SMOTE algorithm

I. INTRODUCTION

Classification learning from a static dataset can be done easily. So, it is assumed that the dataset contain all necessary information to learn the relevant concepts. The model which is working in real world scenarios, e.g., intrusion detection, spam detection, fraud detection, loan recommendation, climate data analysis makes some prediction on previous data to detect the upcoming changes. All training data often received over time in streams of instances or batches. Arrival of data takes different ways either incrementally or in batches. Learning of model using all the information predicts new instances arriving at time step t+ 1. A learning algorithm is incremental when it produces a sequence of dependences on the training data and a limited number of previous hypotheses. A classifier can be updated incrementally from newly available data and simultaneously maintaining the performance of the classifier on old data. Stability of classifier evaluated when it is learning through the changing dataset and adaptive to the new concept. Concept change causes classification problem, as received emails changes as time. [Kuncheva L,2004] data stream is an ordered sequence of instances. Data streams can be processed by online classifiers. Those classifiers should have the following qualities Single pass through the data. The classifier reads each example only once. Limited memory and processing time. Each example should be processed very fast and in a constant period of time. Any-time learning. The classifier should provide the best answer at every moment of time. Concept drift is the fundamental of problem in learning drifting concepts is how to identify those data in the training set in a timely manner that are no longer consistent with the current concepts and hence several criteria is used to measure the concept drift. There are several approaches to track the drift from the dataset; detection of drift has got major research attention. The concept refers to the quantity to be predicted. It can also refer to other phenomenon of interest besides the target concept. Such as an input, but in the context of concept drift the term commonly refers to the target variable.

II. CLASS IMBALANCE PROBLEM IN CLASSIFICATION

We first introduce the problem of imbalanced data-sets in classification. Then, we present how to evaluate the performance of the classifiers in imbalanced domains. Finally, we recall learn++.NIE techniques to address the class imbalance problem, specifically, the data level approaches that have been Combined with ensemble learning algorithms in previous works. Prior to the introduction of the problem of class imbalance, we should formally state the concept of supervised classification. In machine learning, the aim of classification is to learn a system capable of the prediction of the unknown output class of a previously unseen instance with a good generalization ability.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 9, September 2013

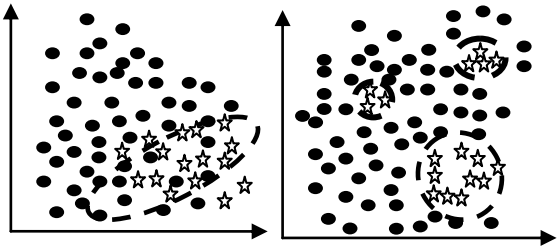

Fig 1. Example of difficulties in imbalanced data sets. (a) Class overlapping (b) Small disjuncts

The learning task, i.e., the knowledge extraction, is carried out by a set of n input instances $x_1,...,x_n$ characterized by i features $a_1,...,a_i \in A$, which includes numerical or nominal values, whose desired output class labels $y_j \in C = \{c_1,...,c_m\}$,in the case of supervised classification, are known before to the learning stage. In such a way, the system that is generated by the learning algorithm is a mapping function that is defined over the patterns $A^i \rightarrow C$, and it is called classifiers.

A. ***The Problem of Imbalanced Data-Sets***

In classification, a data-set is said to be imbalanced when the number of instances which represents one class is smaller than the ones from other classes. Furthermore, the class with the lowest number of instances is usually the class of interest from the point of view of the learning task [N. V. Chawla,et al,2004]. This problem is of great interest because it turns up in many real world classification problems, such as remote sensing, pollution detection [W.-Z. Lu and D.Wang, 2008], risk management [Y.-M. Huang,et al,2006], fraud detection, and especially medical diagnosis[X. Peng and I. King,2008].In these cases, standard classifier learning algorithms have a bias toward the classes with greater number of instances, since rules that correctly predict those instances are positively weighted in favour of the accuracy metric, whereas specific rules that predict examples from the minority class are usually ignored (Treating them as noise), because more general rules are preferred. In such a way, minority class instances are more often misclassified than those from the other classes. Anyway, skewed data distribution does not hinder the learning task by itself. The issue is that usually a series of difficulties related to this problem turn up.

a) *Small sample size*

Generally imbalanced data-sets do not have enough minority class examples. In [6], the authors reported that the error rate caused by imbalanced class distribution decreases when the number of examples of the minority class is representative (fixing the ratio of imbalance). This way, patterns that are defined by positive instances can be better learned despite the uneven class distribution. However, this fact is usually unreachable in real-world problems.

b) *Overlapping* or *class separability*

[See Fig. 1(a)]: When it occurs, discriminative rules are hard to induce. As a consequence, more general rules are induced that misclassify a low number of instances (minority class instances) [7]. If there is no over lapping between classes, any simple classifier could learn an appropriate classifier regardless of the class distribution.

c) *Small disjuncts*

[See Fig. 1(b)]: The presence of small disjuncts in a data-set occurs when the concept represented by

Table .1
Confusion Matrix for a Two-Class Problem

Class	Positive prediction	Negative prediction
Positive class	True Positive	False Negative
Negative class	False Positive	True Negative

The minority class is formed of sub concepts [8]. The majority class corresponds to a class or set of classes that is the large majority of the instances in a dataset. The minority class is under-represented in the training data. The minority class is typically of greater importance than the majority class to the pattern recognition problem. Besides, small disjuncts are implicit in most of the problems. The existence of sub concepts also increases the complexity of the problem because the amount of instances among them is not usually balanced. In this paper, we focus on two-class imbalanced data-sets, where there is a positive (minority) class, with the lowest number of instances, and a negative (majority) class, with the highest number of instances. We also consider the imbalance ratio (IR) defined as the number of negative class examples that are divided by the number of positive class examples, to organize the different data-sets.

B. Performance Evaluation in Imbalanced Domains:

The evaluation criterion is a key factor both in the assessment of the classification performance and guidance of the classifier modelling. In a two class problem, the confusion matrix (shown in Table I) records the results of correctly and incorrectly recognized examples of each class. Traditionally, the accuracy rate (1) has been the most commonly used empirical measure. However, in the framework of imbalanced data-sets, accuracy is no longer a proper measure, since it does not distinguish between the numbers of correctly classified examples of different classes. Hence, it may lead to erroneous conclusions, i.e., a classifier that achieves an accuracy of 90% in a data-set with an IR value of 9, is not accurate if it classifies all examples as negatives.

$$Acc=\frac{TP+TN}{TP+FN+FP+TN}$$

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 9, September 2013

For this reason, when working in imbalanced domains, there are more appropriate metrics to be considered instead of accuracy. Specifically, we can obtain four metrics from Table I to measure the classification performance of both, positive and negative, classes independently.

i. True positive rate $TP_{rate}= \frac{TP}{TP+FN}$ is the percentage of positive instances correctly classified.

ii. True negative rate $TN_{rate}= \frac{TN}{FB+TN}$ is the percentage of negative instances correctly classified.

iii. False positive rate $FP_{rate} = \frac{FP}{FP+TN}$ is the percentage of negative instances misclassified.

iv. False negative rate $FN_{rate}= \frac{FN}{TP+FN}$ is the percentage of positive instances misclassified.

Clearly, since classification intends to achieve good quality results for both classes, none of these measures alone is adequate by itself. One way to combine these measures and produce an evaluation criterion is to use the receiver operating characteristic (ROC) graphic [9]. This graphic allows the visualization of the trade-off between the benefits (TP_{rate}) and costs (FP_{rate}); thus, it evidences that any classifier cannot increase the number of true positives without the increment of the false positives. The area under the ROC curve (AUC) corresponds to the probability of correctly identifying which one of the two stimuli is noise and which one is signal plus noise. AUC provides a single measure of a classifier's performance for the evaluation that which model is better on average.

Fig. 2 shows how to build the ROC space plotting on a two dimensional chart, the TP_{rate} (Y-axis) against the FP_{rate} (X-axis). Points in $(0, 0)$ and $(1, 1)$ are trivial classifiers where the predicted class is always the negative and positive, respectively.

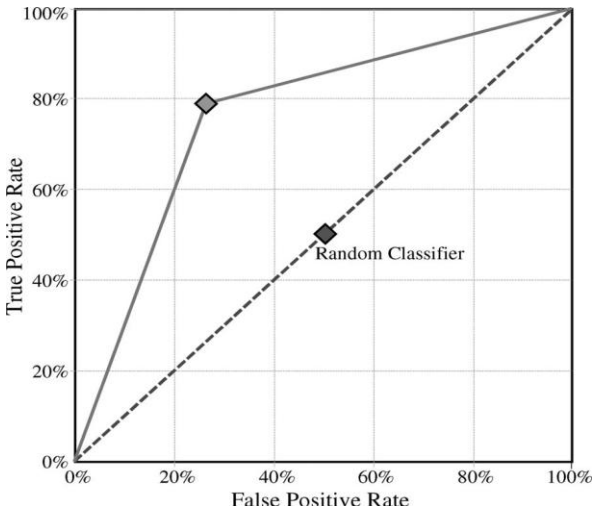

Fig 2: Graph Representation of True Positive Rate and False Positive Rate

On the contrary, $(0, 1)$ point represents the perfect classification. The AUC measure is computed just by obtaining the area of the graphic:

$$AUC=\frac{1+TP_{rate}FP_{rate}}{2}$$

C) Dealing with the Class Imbalance Problem

On account of the importance of the imbalanced data-sets problem, a large amount of techniques have been developed to address this problem. As stated in the introduction, these approaches can be categorized into three groups, depending on how they deal with the problem.

a) *Algorithm level* approaches (also called *internal*) try to adapt existing classifier learning algorithms to bias the learning toward the minority class [10]. These methods require special knowledge of both the corresponding classifier and the application domain, comprehending why the classifier fails when the class distribution is uneven.

b) *Data level* (or *external*) approaches rebalance the class distribution by resampling the data space [11].This way, they avoid the modification of the learning algorithm by trying to decrease the effect caused by imbalance with a pre-processing step. Therefore, they are independent of the classifier used, and for this reason, usually more versatile.

c) *Cost-sensitive learning* framework falls between data and algorithm level approaches. It incorporates both data level transformations (by adding costs to instances) and algorithm level modifications (by modifying the learning process to accept costs). It biases the classifier toward the minority class the assumption higher misclassification costs for this class and seeking to minimize the total cost errors of both classes. The major drawback of these approaches is the need to define misclassification costs, which are not usually available in the data-sets. Aside from those three categories, ensemble-based methods can be classified into a new category. These techniques usually consist in a combination between an ensemble learning algorithm and one of the techniques above, specifically, data level and cost sensitive ones. By the addition of a data level approach to the ensemble learning algorithm, the new hybrid method usually pre-processes the data before training each classifier. On the other hand, cost-sensitive ensembles instead of modifying the base classifier in order to accept costs in the learning process guide the cost minimization via the ensemble learning algorithm. This way, the modification of the base learner is avoided, but the major drawback in the cost.

III. DATA LEVEL METHODS FOR HANDLING IMBALANCE DATA

An easy Data level method for balancing the classes consists of resampling the original data set, either by oversampling the minority class or by under-sampling the majority class, until the classes are approximately equally

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 9, September 2013

represented. Both strategies can be applied in any learning system, since they act as a pre-processing phase, allowing the learning system to receive the training instances as if they belonged to a well-balanced data set. Thus, any bias of the system towards the majority class due to the different proportion of examples per class would be expected to be suppressed. Hulse et.al.[12] Suggest that the utility of the resampling methods depends on a number of factors, including the ratio between positive and negative examples, other characteristics of data, and the nature of the classifier. However, resampling methods 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.

A) Undersampling and Oversampling

Under-sampling and over-sampling change the training sets by sampling a smaller majority training set and repeating instances in the minority training set. The level of imbalance is reduced in both methods, with the hope that a more balanced training set can give better results. Both sampling methods are easy to implement and have been shown to be helpful in imbalanced problems. Under sampling requires shorter training time, at the cost of ignoring potentially useful data. Over-sampling increases the training set size, and thus requires longer training time. Furthermore, it tends to lead to overfitting since it repeats minority class examples [13]. Besides the basic under-sampling and over-sampling methods, there are also methods that sample in more complex ways. SMOTE added new synthetic minority class examples by randomly interpolating pairs of closest neighbors in the minority class. The one-sided selection procedures [14] tried to find a representative subset of majority class examples by only removing 'borderline 'and 'noisy' majority examples. Some other methods combine different sampling strategies to achieve further improvement. Also, researchers have studied the effect of varying the level of imbalance and how to find the best ratio when a C4.5 tree classifier was used [15].

B) Synthetic Minority Oversampling Technique: [SMOTE]

Over-sampling by replication can lead to similar but more specific regions in the feature space as the decision region for the minority class. This can potentially lead to over fitting on the multiple copies of minority class examples. To overcome the over fitting and broaden the decision region of minority class examples, we introduced a novel technique to generate synthetic examples by operating in "feature space" rather than "data space" (Chawla et al., 2002). The minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbors. Depending upon the amount of over-sampling required, neighbors from the k nearest neighbors are randomly chosen. Synthetic samples are generated in the following way: Take the difference between the feature vector (sample) under consideration and its nearest neighbor. Multiply this difference by a random number between 0 and l, and add it to the feature vector under consideration. This causes the selection of a random point along the line segment between two specific features. This approach effectively forces the decision region of the minority class to become more general. For the nominal cases, we take the majority vote for the nominal value amongst the nearest neighbors. We use the modification of Value Distance Metric (VDM) (Cost and Salzberg, 1993) to compute the nearest neighbors for the nominal valued features. The synthetic examples cause the classifier to create larger and less specific decision regions, rather than smaller and more specific regions, as typically caused by over-sampling with replication. More general regions are now learned for the minority class rather than being subsumed by the majority class samples around them. The effect is that decision trees generalize better. SMOTE was tested on a variety of datasets, with varying degrees of imbalance and varying amounts of data in the training set, thus providing a diverse tested. SMOTE forces focused learning and introduces a bias towards the minority class. On most of the experiments, SMOTE using C4.5 (Quinlan, 1992) and Ripper (Cohen, 1995a) as underlying classifiers outperformed other methods including sampling strategies, Ripper's Loss Ratio, and even Naive Bayes by varying the class priors.

C) Overcoming Class Imbalance In Concept Drifting Data Streams

In the previous sections we focused on strategies for over- coming concept drift in balanced class distributions. While this research is valuable, a large number of concept drifting data sources also suffer from class imbalance (e.g., credit card fraud, network intrusion detection, etc.). In this section we outline various methods which seek to overcome both issues simultaneously, and note the relative paucity of research into such methods. In addition to being the most commonly applied technique when dealing only with concept drift, ensemble methods have also been the de facto standard for combating class imbalance. Gao et al proposed a framework based on collecting positive class examples. In their ensemble algorithm, they break each incoming chunk into a set of positive (P) and negative (Q) class instances. One then selects all seen positive class instances (AP), and a subset of the negative class instances (O) which is determined randomly based on a distribution ratio. These two sets are then combined to form a complete dataset to train the new ensemble classifier Ci. By accumulating all positive class instances, this approach implicitly assumes, however, that the minority class is not drifting. Building on this concept,

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 9, September 2013

Chen and He propose SERA, which is similar to the proposal of Gao et al. however, instead of using all past instances, the algorithm selects the
"Best" n minority class instances as defined by the Mahalanobis distance. Given these instances, the algorithm then uses all majority class instances and uses bagging to build an ensemble of classifiers. Thus SERA suffers from a similar, albeit less severe, concern as the method proposed by Gao et al., as the algorithm may not be able to track drift in minority instances depending on the parameter n. Similarly, Lichtenwalter and Chawla propose an extension of Gao et al.'s work where instead of propagating all minority class examples, they also propagate misclassified majority class instances. In this way, they seek to better defined the boundary between the classes, thereby increasing the performance of the ensemble members. Additionally, they propose to use a combination of Hellinger distance and information gain to measure the similarity of the cur- rent batch to the batch that each ensemble member was built on. The more similar the batches, the more likely that they describe the same concept. Thus each ensemble member's probability estimate is weighted by the similarity measure in order to obtain a more accurate prediction. Finally, Ditzler and Polikar outline a method for extending their Learn++.NSE algorithm for the case of class imbalance. In these papers, the authors propose Learn++.NIE (for learning in non-stationary and imbalanced environments). In Learn++.NIE, the authors apply the logic of the Learn++.NSE algorithm, with an additional step of using bagging instead of a single base classifier. In this way, the authors claim they can both reduce error via bagging, and, more importantly, learn on a less imbalanced dataset by under-sampling the majority class when creating each bag.

IV CONCLUSION

The paper provides an overview of the classification of imbalanced data sets. At data level, sampling is the most common approach to deal with imbalanced data. Oversampling clearly appears as better than under-sampling for local classifiers, whereas some under-sampling strategies outperform over-sampling when employing classifiers with global learning.

REFERENCE

[1] Kuncheva L. I., Classifier Ensembles for Changing Environments, Proceedings of 5th International Workshop on Multiple Classifier Systems, MCS 04, LNCS, vol.3077, p. 1-15, Springer-Verlag, 2004.
[2] N. V. Chawla, N. Japkowicz, and A. Kolcz, Eds., Special Issue Learning Imbalanced Datasets, SIGKDD Explor. Newsl. vol. 6, no. 1, 2004.
[3] W.-Z. Lu and D.Wang, "Ground-level ozone prediction by support vector machine approach with a cost-sensitive classification scheme," Sci. Total.Enviro., vol. 395, no. 2-3, pp. 109–116, 2008.
[4] Y.-M. Huang, C.-M. Hung, and H. C. Jiau, "Evaluation of neural networks and data mining methods on a credit assessment task for class imbalance problem," Nonlinear Anal. R. World Appl., vol. 7, no. 4, pp. 720–747, 2006.
[5] X. Peng and I. King, "Robust BMPM training based on second-order cone programming and its application in medical diagnosis," Neural Netw., vol. 21, no. 2–3, pp. 450–457, 2008.
[6] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," Intell. Data Anal., vol. 6, pp. 429–449, 2002.
[7] V. Garc´ıa, R. Mollineda, and J. S´anchez, "On the k-nn performance in a challenging scenario of imbalance and overlapping," Pattern Anal. App., vol. 11, pp. 269–280, 2008.
[8] G. M. Weiss and F. Provost, "Learning when training data are costly: The effect of class distribution on tree induction," J. Artif. Intell. Res., vol. 19, pp. 315–354, 2003.
[9] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," Pattern Recog., vol. 30, no. 7, pp. 1145–1159, 1997.
[10] R. Barandela, J. S. S´anchez, V. Garc´ıa, and E. Rangel, "Strategies for learning in class imbalance problems," Pattern Recog., vol. 36, no. 3,pp. 849–851, 2003.
[11] 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 Expl. Newslett, vol. 6, pp. 20–29, 2004.
[12] Hulse, J., Khoshgoftaar, T., Napolitano, an Experimental perspectives on learning from imbalanced datal In: Proceedings of the 24th International Conference on Machine learning, pp. 935–942 (2007).
[13] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, 2002.
[14] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: One-sided selection," in Proceedings of the 14th International Conference on Machine Learning, Nashville, TN, 1997, pp. 179–186.
[15] G. M. Weiss and F. Provost, "Learning when training data are costly: The effect of class distributions on tree induction," Journal of Artificial Intelligence Research, vol. 19, pp. 315–354, 2003.