A multiple expert approach to the class imbalance problem using Inverse Random Under Sampling

M. A. Tahir, J. Kittler, K. Mikolajczyk and F. Yan

Centre for Vision, Speech and Signal Processing
University of Surrey
Guildford, GU2 7XH, UK
{m.tahir,j.kittler,k.mikolajczyk,f.yan}@surrey.ac.uk

Abstract. In this paper, a novel inverse random under sampling (IRUS) method is proposed for class imbalance problem. The main idea is to severely under sample the negative class (majority class), thus creating a large number of distinct negative training sets. For each training set we then find a linear discriminant which separates the positive class from the negative class. By combining the multiple designs through voting, we construct a composite between the positive class and the negative class. The proposed methodology is applied on 11 UCI data sets and experimental results indicate a significant increase in Area Under Curve (AUC) when compared with many existing class-imbalance learning methods.

1 Introduction

Many real world classification problems are represented by highly imbalance data sets, that is, the number of samples from one class is much smaller than from another. This is known as class imbalance problem and is often reported as an obstacle to construct a model that can successfully discriminate minority samples from majority samples. Generally, the problem of imbalanced data sets occurs when one class represents a rare or uncommon concept while the other class represents the anti-concept, so that the examples from the anti-concept class outnumber the examples from the concept class. This type of data is found, for example, in the image retrieval concept detection problem where only few images belong to the concept class; in medical record databases for rare diseases where a small number of patients would have a particular disease.

There is a great deal of research on learning from imbalanced data sets reported in the literature [1, 8, 6]. The most commonly used methods to handle imbalanced data sets involve under sampling or over sampling of the original data set. Over sampling aims to balance class populations through replicating the minority class examples while under sampling aims to balance the class populations through the elimination of majority class examples.

In this paper, a novel inverse random under sampling (IRUS) method is proposed for the class imbalance problem in which the ratio of the respective training set cardinalities is inversed. The idea is to severely under sample the negative class (majority class), thus creating a large number of distinct negative training sets. For each training set we then find a linear discriminant which separates the positive class from the negative samples. As the number of positive samples in each training set is greater than the number of negative samples, the focus in machine learning is on the positive class and consequently it can invariably be successfully separated from the negative training samples. Thus each training set yields one classifier design. By combining the multiple designs through voting, we construct a composite between the positive class and the negative class. We shall argue that this boundary has the capacity to delineate the positive class more effectively than the solutions obtained by conventional learning. We shall show experimentally on standard benchmarking data that the proposed method leads to significant improvements in performance.

This paper is organized as follows. Section 2 provides briefly review several class imbalance methods followed by proposed inverse random under sampling method (IRUS) in section 3. Section 4 describes the experimental setup followed by results and discussion in Section 5. The paper is drawn to conclusion in Section 6.

2 Related Work

As discussed in Section 1, the most commonly used methods to handle imbalanced data sets involve under sampling or over sampling of the original data sets. Random over sampling and random under sampling are the most popular non-heuristic methods that balance class representation through random replication of the minority class and random elimination of majority class examples respectively. There are some limitations of both random under sampling and random over sampling. For instance, under-sampling can discard potentially useful data while over-sampling can increase the likelihood of overfitting [1]. Despite these limitations, random over sampling in general is among the most popular sampling techniques and provides competitive results when compared with most complex methods [1, 12].

Several heuristic methods are proposed to overcome these limitations including Tomek links [13], Condensed Nearest Neighbour Rule (CNN) [7], One-sided selection [10] and Neighbourhood Cleaning rule (NCL) [11] are several well-known methods for under-sampling while Synthetic Minority Over-Sampling Technique (SMOTE) is a well-known method for over-sampling technique [5]. The main idea in SMOTE is to generate synthetic examples by operating in the "feature space" rather than the "data space" [5]. The minority class is oversampled by interpolating between several minority class examples that lie together. Depending upon the amount of over-sampling required, neighbours from the k nearest neighbours are randomly chosen. Thus, the overfitting problem is avoided and the decision boundaries for the minority class are spread further into the majority class space [1].

Liu et al [12] and Chan et al [3] examine the class imbalance problem by combining classfiers built from multiple under-sampled training sets. In both approaches, several subsets from the majority class with each subset having approximately the same number of samples as the minority class are created. One classifier is trained from each of these subsets and the minority class and then the classifiers are combined. Both these approaches differ in grouping multiple classifiers and in creating subsets from the majority class.

3 Inverse Random Under Sampling

In this section, we will discuss the proposed inverse random under sampling (IRUS) method. For convenience, we refer to the minority class as the concept class and the majority class as the anti-concept class. A conventional training of a concept detector using a data set containing representative proportions of samples from the concept and anti concept classes will tend to find a solution that will be biased towards the larger class. In other words, the probability of misclassifying samples from the anti-concept class will be lower than the probability of error for the concept class. However, the actual performance will be determined by the underlying overlap of the two classes and the class prior probabilities. Thus, we need to control the probability of misclassification of samples from the anti-concept class to achieve the required target performance objectives. This may require setting the operating point of the detector so as to achieve false positive rate that is lower than what would be yielded by conventional training. This could be achieved by biasing the decision boundary in favour of the anti concept sample error rates using threshold (off set) manipulation. Alternatively, we could increase the imbalance between the number of samples from the two classes artificially by eliminating some of them. The latter solution is not very sensible, as we would be depleting the class which is naturally underrepresented even further. The former solution would lead to a substantial increase in the false negative rate.

The problem of learning decision functions in situations involving highly imbalanced class sizes is sometimes mitigated by stratified sampling. This aims to create a training set containing a comparable numbers of samples from all the classes. Clearly, in stratified sampling the training set size would be determined by the number of samples in the underrepresented class. This would lead to a drastic subsampling of the anti-concept class with the resultant reduction in the accuracy of the estimated class boundary. This loss of accuracy can be recovered by means of multiple classifier methodology. By drawing randomly multiple subsets from the anti-concept class data set, each adhering to the stratified sampling criteria, we can design several detectors and fuse their opinions. For a typical imbalance of priors of say 100 : 1, the number of the designs would be too low to allow an alternative approach to controlling false positive error rate and one would have to resort to the biasing methods discussed earlier.

Suppose we take the data set manipulation to the extreme and inverse the imbalance between the two classes. Effectively we would have to draw sample sets from the anti-concept class of size proportional to P^2 where P is the prior probability of the concept class. This would lead to very small sample sets for the anti-concept class and therefore, a poor definition of the boundary between the two classes. Nevertheless, the boundary would favour the concept class. Also, as the number of samples from the negative class is very small in relation to the dimensionality of the feature space, the capacity of each boundary to separate the classes fully is high. Moreover, as the number of samples drawn is proportional to P^2, the number of independent sets that can be drawn will be of the order of $\frac{1}{P^2}$. This large number of designs could then be used for controlling the false positive rate using a completely different mechanism. By combining the designed detectors using voting, we can control the threshold on the number of votes needed to accept the concept hypothesis, thus controlling the false positive error rate. This contrasts with the complex task of biasing a decision boundary in high dimensional space.

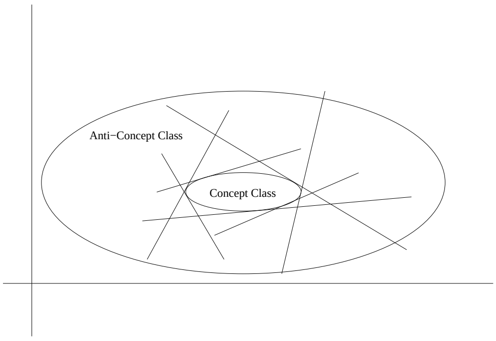

Fig. 1. Schematic diagram showing each boundary partitions the training data set by a hyperplane tangent to the surface of the volume occupied by the concept class.

Interestingly, there is another important benefit of the the IRUS method. As the number of samples forming the negative class is very small, each detector design will be significantly different. This will produce highly diverse detectors which are required for effective classifier fusion. The fused decision rule achieves better class separation than a single boundary, albeit estimated using more samples. This is conveyed schematically in Figure 1. Each boundary partitions the training data set by a hyperplane tangent to the surface of the volume occupied by the concept class. It is the union of these tangent hyperplanes created by fusion, which constitutes a complex boundary to the concept class. Such boundary could not easily be found by a single linear discriminant function. If one resorted to nonlinear functions, the small sample set training would most likely lead to a over fitting and, consequently, to poor generalisation on the test set. Figure 2 provides supporting evidence for the above conjecture. The histogram of discriminant function values (i.e. distance from the decision boundary) generated by one thousand classifiers designed using the inverse imbalance sampling principle for a single negative class test sample (blue bar) shows many of the classifiers scoring positive values which lie on the concept class side of the boundary. This is expected for more than half of the classifiers, as the negative sample will lie beyond the concept class, but nevertheless on the same side as the concept class. In contrast, discriminant function values for a single positive class test sample show that most of the classifiers scoring positive values lie on the concept side of the boundary.

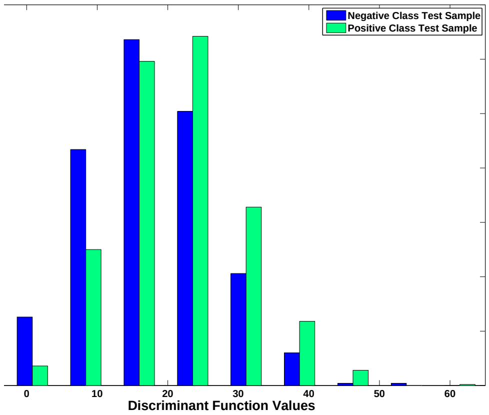

Fig. 2. Histogram of Discriminant Function generated by one thousand classifiers.

In summary, we propose a classifier design approach which is based on an inverse imbalance sampling strategy. This is accomplished by setting the appropriate threshold in the fusion stage combining the outputs of the multiple concept detectors. It allows a very accurate definition of the boundary between the concept class and the negative class.

The pseudo code of IRUS is shown in Algorithm 1. S and $Sets$ are user specified parameters. S controls the number of negative samples drawn at random in each model while $Sets$ determine the number of models or classifiers. For each set Ξ'_a paired with Ξ_c we learn a model h_i. For each model h_i, the probability of unseen instances belonging to concept class D_c is calculated. The probabilities from all models are added. The output is a probability set Ξ_p of the test instances belonging to concept class. Ξ_p is then used to calculate the performance measure discussed in Section 4.3.

4 Experiments

4.1 Experimental Setup

To evaluate the effectiveness of the proposed method, extensive experiments were carried out on 11 public data sets from UCI repository which have different

Algorithm 1 PseudoCode for Inverse Random Under Sampling (IRUS)

```
Require: $\Xi_c$: Training set of concept patterns with cardinality $N_c$
  $\Xi_a$: Training set of anti-concept patterns with cardinality $N_a$
  $\Xi_t$: Test set with cardinality $N_t$
  $S$: Number of samples from $\Xi_a$ for each Model
  $Sets$: Number of classifiers, default: $ceil(\frac{N_c}{S})$
Ensure: $\Xi_p$: Probability set of Test instances belonging to concept class
  $\Xi_p \Leftarrow 0$
  for $i = 1$ to $Sets$ do
    $\Xi'_a \Leftarrow$ Randomly pick $S$ samples without replacement from $\Xi_a$
    $T_s \Leftarrow \Xi'_a + \Xi_c$
    Train base classifier $h_i$ using $T_s$ samples
    for $j = 1$ to $N_t$ do
      $D_c \Leftarrow$ Probability distribution of Test Sample $\Xi_{tj}$ belonging to concept class
      from $h_i$
      $\Xi_{pj} \Leftarrow \Xi_{pj} + D_c$
    end for
  end for
```

degrees of imbalance [2]. Table 1 describes the data sets used in this study. For each data set, it shows the number of attributes (A), number of samples (N_s), number of majority samples (N_a) and number of minority samples (N_c). As in [1, 12], for more than two classes, the class with fewer samples is chosen as the positive class and the remaining as the negative class.

For every data set, we perform a 10-fold stratified cross validation. The whole cross validation is repeated 10 times, and the final values are the averages of these 10 cross validation runs.

Table 1. Description of Data sets. Ratio is the size of majority class divided by that of minority class.

Data set	Samples	Attributes	Concept/Anti-Concept	#min/#maj	Ratio
	N_s	A		N_a/N_c	
Flag	194	28	White/Remainder	17/177	10.42
German	1000	20	Bad/Good	300/700	2.33
Glass	214	9	Ve-win-float-proc/Remainder	17/197	11.59
Haberman	306	3	Die/Survive	81/225	2.78
Mf-Mor	2000	6	10/Remainder	200/1800	9.0
Mf-Zer	2000	47	10/Remainder	200/1800	9.0
Nursery	12960	8	Not-recom/Remainder	328/12632	38.51
Phoneme	5404	5	1/0	1586/3818	2.41
Pima	768	8	1/0	268/500	1.87
Satimage	6435	36	4/Remainder	626/5809	9.28
Vehicle	846	18	Van/Remainder	199/647	3.25

4.2 Benchmark Methods

Decision tree (C45) is used as the base classifier for the proposed inverse random under sampling technique (IRUS). The IRUS method is compared with the following class imbalance techniques: Random Under Sampling (RUS), Random Over Sampling (ROS) and SMOTE. The WEKA [14] implementation is used for C45 and SMOTE and the k nearest neighbour parameter is set to 5 in SMOTE. Further, since pruning and unpruned trees can have different effects on learning from imbalanced data sets, all methods are evaluated using both pruned (25% confidence lavel)/unpruned decision trees. The presented method is also compared with Chan and Stolfo's method [3] (ChSt). The only difference is that the number of majority class examples sampled by ChSt method is equal to the number of minority class examples, while the number of majority class examples sampled in this paper is smaller than the number of minority class examples.

4.3 Performance Measure

The area under the receiver operating characteristic curve (AUC) is most commonly used measure for class imbalance data sets [9, 12] and is adopted here. The AUC represents the expected performance as a singular scalar. It integrates performance of the learning method over all possible values of false positive rate. The Mann Witney statistic is used to calculate the AUC and is implemented in WEKA [14].

5 Results and Discussion

Table 2 shows the AUC for various data sets using different methods. It is clear from Table 2 that AUC using unpruned tree is higher than AUC using pruned tree. This is due to the fact that pruning can reduce the minority class coverage in the decision trees [4]. On Nursery and Vehicle data sets, all methods have achieved very high AUC (> 0.95) for both pruned and unpruned decision trees. Overall, our proposed IRUS method has increased performance in 7 out of 11 data sets. There is an increase in the performance in all data sets except phoneme, mf-mor, nursery and satimage. For mf-mor, nursery and satimage, the difference is not significant. However, for phoneme, there is a significant decrease in performance when compared with all other methods. This is explained by the fact that number of positive samples is quite high in this data set (1586 out of 3818) and since only few samples from negative class are used to learn a model, some negative samples are always on the wrong side of the boundary. Overall, the average AUC for IRUS is approximately 10.1%, 4.6%, 2.8%, 2.7%, 0.63% better than J48, RUS, ROS, SMOTE, ChSt respectively when unpruned decision tree is used. It should be noted that the minority class is over-sampled at different values for SMOTE and the highest mean AUC is obtained when minority class is over-sampled at 400%. For IRUS, again after experimenting with different run-time paramters, the paramters used are $S = 15$ and $Sets = 1.5 \times ceil(\frac{N_c}{S})$.

Table 2. AUC of the compared methods.

Data set	Pruning	J48	RUS	ROS	SMOTE	ChSt	IRUS
Flag	yes	0.5000	0.7354	0.7424	0.6592	0.7891	0.7852
	no	0.7089	0.7581	0.7289	0.6926	0.7921	**0.7949**
German	yes	0.7061	0.6969	0.7058	0.7164	0.7254	0.5365
	no	0.7021	0.6950	0.7047	0.7141	0.7234	**0.7668**
Glass	yes	0.5894	0.7036	0.7635	0.7818	0.7899	0.8148
	no	0.6432	0.7078	0.7656	0.7820	0.8121	**0.8169**
Haberman	yes	0.5851	0.6167	0.6320	0.6693	0.6454	0.6555
	no	0.6182	0.6100	0.6367	0.6726	0.6545	**0.6877**
Mf-Mor	yes	0.500	**0.9294**	0.9234	0.9264	0.9286	0.9275
	no	0.5000	0.9284	0.9227	0.9269	0.9281	0.9262
Mf-Zer	yes	0.5980	0.8660	0.8771	0.8754	0.9006	**0.9072**
	no	0.8667	0.8660	0.8771	0.8752	0.9007	0.9065
Nursery	yes	0.9940	0.9606	0.9975	0.9944	0.9898	0.9850
	no	0.9975	0.9743	**0.9982**	0.9973	0.9965	0.9978
Phoneme	yes	0.9127	0.8931	0.9251	0.9174	0.9146	0.8429
	no	0.9151	0.8960	**0.9254**	0.9195	0.9238	0.8596
Pima	yes	0.7756	0.7572	0.7763	0.7717	0.7671	0.8110
	no	0.7788	0.7626	0.7781	0.7747	0.7689	**0.8167**
Satimage	yes	0.9084	0.9095	0.9214	0.9202	0.9454	0.9289
	no	0.9162	0.9109	0.9213	0.9208	**0.9486**	0.9405
Vehicle	yes	0.9770	0.9649	0.9768	0.9740	0.9810	0.9810
	no	0.9769	0.9679	0.9779	0.9758	**0.9850**	**0.9850**
Average	yes	0.7319	0.8197	0.8405	0.8369	**0.8524**	0.8341
	no	0.7840	0.8252	0.8397	0.8411	0.8581	**0.8635**

Table 3 shows the results of t-test (significance level 0.05) of AUC. The t-test is shown separately for pruned and unpruned trees in the upper and lower triangles respectively. The table clearly indicates that IRUS achieves significant performance gains when compared with other methods. For unpruned decision tree, the t-test reveals that IRUS performs significantly better in 8 out of 11 data sets when compared with ROS and SMOTE and 5 out of 11 when compared with ChSt. IRUS is significantly lower in only 1 data set (phoneme) when compared with RUS and SMOTE while only in 2 data sets when compared with ROS and ChSt. For pruned decision tree, IRUS performs significantly better in 8 and 6 data sets when compared with ROS and SMOTE respectively, although the overall average AUC for IRUS is less than ROS and SMOTE (see Table 2).

Figure 3 shows the different values of run-time parameter S vs AUC in glass, haberman and pima data sets. This parameter effectively controls the number of anti-concept samples drawn at random in each model (classifier). It is observed that IRUS performs best for values in the range $[5 - 20]$. Parameter S also effects the training time. For low value of S, more sets or classifiers (See Algorithm 1) are trained while for high value of S, less classifiers are required. We

Table 3. Summary of t-test with significance level at 0.05. The upper triangle shows the results with pruned decision tree and the lower triangle shows the results with unpruned decision trees. Each tabular shows the amount of WIN-TIE-LOSE of a method in a row comparing with the method in a column.

	J48	RUS	ROS	SMOTE	ChSt	IRUS
J48	-	4-2-5	0-3-8	1-2-8	1-2-8	3-0-8
RUS	3-3-5	-	1-3-7	2-0-9	1-1-9	2-1-8
ROS	7-4-0	8-2-1	-	4-5-4	2-3-6	3-0-8
SMOTE	7-4-0	9-1-1	2-7-2	-	3-4-4	3-2-6
ChST	9-1-1	9-2-0	8-2-1	5-4-2	-	4-4-3
IRUS	9-1-1	9-1-1	8-1-2	8-2-1	5-4-2	-

have also experimented with different values of other run-time parameter $Sets$. This parameter is important to make sure that almost all anti-concept samples are selected during different models. After some experiments, it is observed that the mean AUC is almost identical when $Sets > 1.5 \times ceil(\frac{N_c}{S})$.

6 Conclusion

A novel inverse random under sampling (IRUS) method is proposed in this paper to solve the class imbalance problem. The main idea is to use disproportionate training set sizes, but by inversing the training set cardinalities. By the proposed method of inverse under sampling of the majority class, we can construct a large number of minority class detectors which in the fusion stage has the capacity to realise a complex decision boundary. The distinctiveness of IRUS is assessed experimentally using 11 public UCI data sets. The results indicate significant performance gains when compared with other class imbalance methods.

In this paper, C4.5 is used as a base classifier. It would be interesting to see how other well-known classifiers like NaiveBayes, SVM, KNN, LDA behave when used as a base classifier in our proposed inverse under sampling method.

Acknowledgements This work was supported by the EU VIDI-Video Project.

References

1. G. Batista and R. C. Prati amd M. C. Monard. A study of the bahavior of several methods for balancing machine learning training data. *SIGKDD Explorations*, 6(20–29), 2004.
2. C. Blake, E. Keogh, and C. J. Merz. UCI repository of machine learning databases.
3. 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 4th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 164–168, New York, NY, 1998.

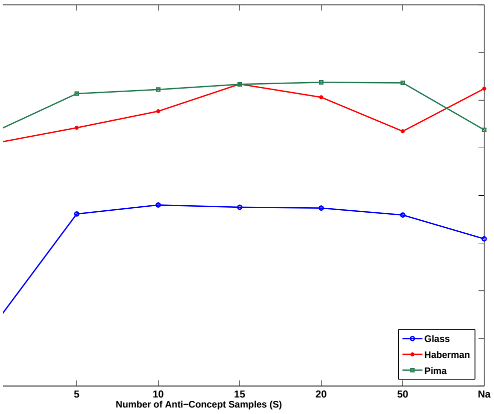

Fig. 3. Run time parameter S vs AUC. N_a = Total number of samples in Concept Class.

4. N. V .Chawla. C4.5 and imbalanced data sets: Investigating the effect of sampling method, probabilistic estimate, and decision tree structure. In *Proceedings of the International Conference on Machine Learning (ICML 2003) Workshop on Learning from Imbalanced Data Sets II*, 2003.
5. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Synthetic minority over-sampling technique. *Journal of Artificial Inetelligence Review*, (16):321–357, 2002.
6. C. P. de Souto Marcilio, V. G. Bittencourt, and A. F. C. Jose. An empirical analysis of under-sampling techniques to balance a protein structural class dataset. *Neural Information Processing, Lecture Notes in Computer Science*, pages 21–29, 2006.
7. P. E. Hart. Condensed nearest neighbor rule. *IEEE Transactions on Information Theory*, 14:515–516, 1968.
8. M. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, (6):429–449, 2002.
9. S. B. Kotsiantis and P. E.Pintelas. Mixture of expert agents for handling imbalanced data sets. *Annals of Mathematics, Computing and Teleinformatics*, 1(1):46–55, 2003.
10. M. Kubat and S. Matwin. Addressing the course of imbalanced training sets: One-sided selection. In *Proceedings of International Conference of Machine Learning*, pages 179–186, 1997.
11. J. Laurikkala. *Improving Identification of Difficult Small Classes by Balancing Class Distribution*, chapter Artificial Intelligence in Medicine, Lecture Notes in Computer Science. Springer, 2001.
12. X. Y. Liu, J. Wu, and Z. H. Zhou. Exploratory under-sampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 2009.
13. I. Tomek. Two modifications of CNN. *IEEE Transactions on Systems, Man and Cybernatics*, (6):769–772, 1976.
14. I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques.* Morgan Kaufmann, 2005.