Comparing SVM Ensembles for Imbalanced Datasets

Vasudha Bhatnagar
Department of Computer Science
Delhi University
Delhi, India
vbhatnagar@cs.du.ac.in

Manju Bhardwaj
Department of Computer Science
Delhi University
Delhi, India
mbhardwaj@maitreyi.du.ac.in

Ashish Mahabal
Department of Astronomy
California Institute of Technology
CA 91125, USA
aam@astro.caltech.edu

***Abstract*—Real life datasets often suffer from the problem of class imbalance, which thwarts supervised learning process. In such data sets examples of positive (minority) class are significantly less than those of negative (majority) class leading to severe class imbalance. Constructing high quality classifiers for such imbalanced training data sets is one of the major challenges in machine learning, since traditional classification algorithms tend to get biased towards majority class.**

In this paper, we compare three ensemble based approaches for handling imbalanced datasets. All the three approaches aim to overcome the under representation of minority class by learning from each of the minority class samples and a subset of majority class samples. The three approaches namely, PARTEN, UMjC and LFM were evaluated on several public datasets. Precision, recall, F- measure, g-mean and ROC space measures were used for comparison. Thread-bare discussion of the results is presented in the paper. Subsequently, we present an astronomy application, where the three methods are compared for prediction of class II, IIn and IIp supernovae.

***Keywords*-SVM, Ensembles, Classification, Supervised Learning, Class imbalance**

I. Introduction

Learning from imbalanced data sets is one of the challenging problems in supervised learning. Imbalanced datasets arise frequently in real life in both scientific and commercial domains. Fraud/intrusion detection, medical diagnosis and protein folding applications intrinsically generate imbalanced datasets either because of rare occurence of events, expensive experimentation or tedious data collection process [1]. A large number of imbalanced datasets prevail in information retrieval and filtering tasks [2].

The problem of class imbalance is quite pervasive and troublesome for data mining/machine learning community, requiring special attention. The problem was formally addressed in a workshop in the beginning of current decade [3]. A special issue of SIGKDD exploration [4] was dedicated to this problem indicating the importance of the problem and the strong interest of the community looking for solutions.

The challenge in imbalanced datasets arises because of the severe under-representation of one class[1] causing suboptimal performance of the classifier [1]. Though the class imbalance may exist in test data too, it is the one in the training data that negatively influences learning. Coupled with the fact that error costs for misclassification for two classes are often vastly different (e.g. in medical, financial and scientific domains), learning from imbalanced data sets has fuelled the quest for solutions.

Class imbalance problem has been handled either using data oriented approaches [5], [6], or algorithmically [1], [7], [8]. Data oriented approaches broadly sample the two classes in various ways to create multiple training sets, each of which is used to induce a classifier. Final prediction is done by the ensemble of induced classifiers [6], [5]. Algorithmic approaches on the other hand focus on techniques like adjusting costs or decision thresholds to counter class imbalance [9], [7]. Recognition based learning (one class learning) methods also fall in this category [8].

In this paper we focus on data oriented approach for handling class imbalance problem. Ensembles using disjunct partitioning [10], [5] (PARTEN) and undersampling of majority class (UMjC) [11], [12], [13] are two well known data oriented approaches to handle the problem. They differ in the way the training sets are created to induce component classifiers from the imbalanced data sets (Sections III-A and III-B). We present empirical comparison of these two methods with a third one which is based on learning-from-mistakes [2] (LFM) paradigm. LFM generates dependent classifiers and selects the components of the ensemble on the basis of user defined criteria of precision of the minority class (Section III-C). SVM is used as the base classifier for all the three methods. The reader is referred to [15], [16] for an introduction to SVM technique for supervised learning.

Empirically SVMs have been shown to handle the class imbalance ratio of 1:10 [5], [11]. However, for more severe class imbalance they do get overwhelmed by the majority class. Extensive experimentation on imbalanced datasets[3] shown in Table II support this view. For each dataset, the best kernel[4](linear, polynomial, RBF and sigmoid) was chosen

[1]As in the prevailing literature, we work in the setting of two class classification problem though in scientific domains, imbalance often exists in multi class classification problems.

[2]Preliminary work on this method is presented in reference [14].

[3]Class imbalance was created by adopting 1 vs. all approach in all multi-class datasets.

[4]LIBSVM [17] implementation was used.

using default parameters. Subsequent to this, for each dataset thirty pairs of training and test sets were generated and SVM was induced for each pair with the selected kernel. The performance metrics were averaged over the thirty pairs. The results in Table II demonstrate the general deterioration of SVM performance with increasing imbalance ratio in datasets. For more than half of the data sets, since not even a single object of positive class was predicted correctly, precision (See Eq. 1) and hence, F-measure are not defined (See Eq. 4). These results strengthen our motivation to work with SVM ensembles for imbalanced datasets.

Table I
DATASETS WITH THEIR CLASSES, ATTRIBUTES(ATTRIB) AND IMBALANCE RATIOS(IR)

Data Set	Min vs Maj	Instances	Attrib	IR
Ecoli-imU	imU vs all	336	7	8.6
Optdigits0	0 vs all	5564	64	9.1
Vowel0	0 vs all	990	10	10
GlassVWFP	Veh-win-float-proc vs all	214	9	10.39
Abalone11-18	11 vs 18	529	7	11.6
EcoliOM	OM vs all	336	7	13.84
Abalone11-19	11 vs 19	519	7	15.2
GlassCont	containers vs all	214	9	15.47
Abalone9-18	18 vs 9	731	7	16.68
GlassTware	tableware vs all	214	9	22.81
YeastCYT-POX	POX vs CYT	483	8	23.15
YeastME2	ME2 vs all	1484	8	28.41
YeastME1	ME1 vs all	1484	8	32.78
YeastEXC	EXC vs all	1484	8	39.16

Table II
BEST PERFORMANCE OF SVM FOR IMBALANCED DATASETS IN TABLE I

Data Set	Kernel	Acc	Prec	Recall	F-msr	gmean
Ecoli-imU	All	89.38	-	0	-	0
Optdigits0	RBF	92.12	1	0.2	0.34	0.45
Vowel0	Poly	98.61	0.98	0.87	0.92	0.93
GlassVWFP	Poly	88.06	0.18	0.11	0.19	0.32
Abalone11-18	All	92.09	-	0	-	0
Ecoli(OM)	All	93.81	-	0	-	0
Abalone11-19	All	93.68	-	0	-	0
GlassCont	Poly	93.33	0.53	0.81	0.64	0.88
Abalone9-18	All	94.26	-	0	-	0
GlassTware	Poly	98.7	0.82	0.91	0.85	0.95
YeastCYT-POX	Linear	98.27	0.96	0.63	0.74	0.79
YeastME2	All	96.57	-	0	-	0
YeastME1	All	96.97	-	0	-	0
YeastEXC	All	97.58	-	0	-	0

The major contribution of the paper is the empirical comparison of the earlier mentioned three SVM ensembles for handling imbalanced datasets. The paper is organized as follows. Section II describes the related work in the field of learning from imbalanced data sets. Section III describes in detail the three methods that are examined. Section IV details the experimental setting. Section V presents the discussion on the results on UCI datasets. An astronomy application for prediction of rare supernovae is presented in Section VI. Finally, section VII concludes the paper.

II. RELATED WORK

Joshi et al. [18] conducted a systematic study to evaluate how boosting performs for the task of mining rare classes. They empirically compared three categories of boosting algorithms and discussed their possible effect on recall and precision of the rare class. Later they demonstrated that boosting mechanism does not overcome the deficiencies of weak base learners, while predicting rare classes [19].

Chawla et al. [1] and Japkowicz [20] present an informative account of the progression of interest of machine learning community in the class imbalance problem. The problem is also known as rare class problem [21] and an excellent overview of this aspect of the problem can be found in Weiss [22]. A review of different approaches adopted in this area can be found in [21]. Data based and algorithm based approaches have been used to handle the class imbalance problem.

The data based approaches include different forms of random sampling or directed sampling. Oversampling (with replacement) the minority class or undersampling the majority class [5], [8], [11] is usually employed to overcome class imbalance. In directed sampling approach, the choice of samples to replace or eliminate is informed rather than random [1]. Some directed sampling approaches effectuate oversampling by generating new examples in an informed manner [23]. Liu et al. [5] create multiple classifiers by undersampling the majority class and oversampling (using SMOTE [23]) the minority class. Multiple training sets are generated and multiple classifiers are induced, predictions from which are then combined for each unseen instance. Yan et al. [6] use SVM ensemble to predict rare classes in scene classification.

Algorithmic mechanisms employed to mitigate class imbalance problem tend to improve a classifier's performance by working with its inherent characteristics. The misclassification costs of classes can be adjusted to counter the class imbalance [9]. While working with decision trees, the probabilistic estimate at the tree leaf can be adjusted [7]. Instead of learning from two classes together (discrimination based learning), learning can be done from each class separately (recognition based learning) [8]. Akbani et al [11] proposed a technique in which they combine oversampling with class cost adjustment using SVM.

III. METHODS EXAMINED

Ensemble methods have been examined extensively by the advocates of data oriented solution to class imbalance problem [5], [6], [24]. We present a detailed description of the three ensemble creation methods examined in this paper. In all these methods, majority voting has been used as the combining function.

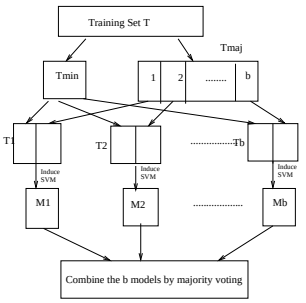

Figure 1. PARTEN approach

A. PARTEN - PARTitionining ENsemble

Given a data set T with imbalance ratio $1 : b$. The examples from minority class are denoted by T_{min} and examples from majority class denoted by T_{maj} . The majority class T_{maj} is divided into b disjunct partitions [25], [6] . With each of these b partitions, the complete minority class is combined to create T_1, T_2 $\ldots T_b$. Thus, b balanced training sets are created. This approach leads to zero data loss since each majority class record is used at least once while minority class records are used b times for learning. Further, the size of ensemble is fixed by the ratio of imbalance. We are aware that some researchers opine that perfect balance in the training set is no guarantee for best learning [21]. But to the best of authors' knowledge, there has been no theoretical work to prove this. We follow this approach because of its simplicity. Figure 1 describes the approach pictorially.

B. UMjC - Undersampling Majority Class

Undersampling (with replacement) is used for T_{maj} to generate b sets [11], [12], [13]. The size of each of the sampled set is the same as the number of instances in T_{min}. Each of the b sets is then combined with T_{min} to generate training sets $T_1 \ldots T_b$. Thus, an ensemble with b components is generated.

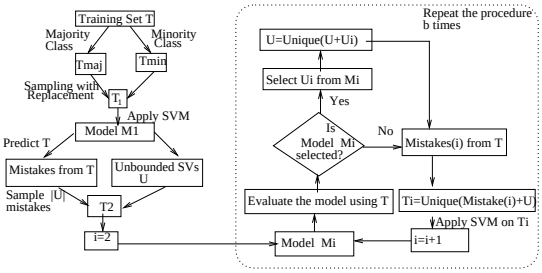

Figure 2. Ensemble creation using LFM

C. LFM Ensemble - Learning From Mistakes

This ensemble technique creates a set of dependant classifiers, each of which is created using the mistakes of the previous classifier, hence, the name "Learning from Mistakes". A balanced training set (T_1) for the first classifier of the ensemble is constructed by sampling (with replacement) the majority class and combining it with all minority class instances.

A weak SVM (M_1) is induced from the training set, by setting a high value of cost parameter. Weak classifiers are preferred in SVM ensemble approach for better predictive performance. The training set T is then predicted by M_1. Since a limited number of negative examples and all positive examples were used for learning, M_1 is expected to make a large number of mistakes. The training set for next iteration is constructed by combining the unbounded support vectors of M_1 with equal number of randomly sampled mistakes. The intuition behind using the unbounded support vectors is that they are the data points which lie exactly on the class margin boundary. These points define the boundary to which the class extends and are favourable candidates for correctly classified instances in the current training set. Note that there are equal number of correctly and incorrectly classified records in T_2.

All subsequent training sets are similarly generated by combining unbounded support vectors from selected models and the mistakes made by the classifier in the previous iteration, while maintaining uniqueness of training instances. The process is repeated b number of times to maintain comparability with the other two methods under investigation. In case there are no mistakes in an iteration, the ensemble may have $< b$ classifiers. Figure 2 describes the process of LFM ensemble creation.

Not all the classifiers that are generated may exhibit desirable performance. In each iteration the training set has equal number of correctly classified and misclassified instances thereby reducing the chances of hyperplane getting biased towards mistakes. However, a situation where the hyperplane is overwhelmed by the mistakes of the negative class, cannot be totally ruled out. Hence, only those classifiers are selected for ensemble creation which show an acceptable performance on the training set. Joshi et al. argued that boosting can fail to achieve overall good recall and precision levels in imbalanced datasets if the base learner always achieves poor recall and precision with respect to the class distribution in training data [19]. Extending this argument, we propose that only those classifiers be selected which have acceptable precision and/or recall. In case the application demands improved recognition of both classes, g-mean can be used as selection measure. In short, LFM method includes those component classifiers, which satisfy the criterion that is desired to be improved.

IV. Experimental Setup

Prior to discussion of results, we describe three important aspects of the experimental setting.

A. Evaluation metrics

Evaluation of a classifier induced by imbalanced data sets needs special attention because despite high accuracy it may not meet user requirement of recognition of minority class. For two-class confusion matrix shown in Table III, a few more measures are defined below:

Table III
TWO CLASS CONFUSION MATRIX - TP:TRUE POSITIVES, TN:TRUE NEGATIVES, FP:FALSE POSITIVES, FN:FALSE NEGATIVES

	Predicted Positive	Predicted Negative
Actual Positive	TP	FN
Actual Negative	FP	FN

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = Sensitivity = \frac{TP}{TP + FN} \tag{2}$$

$$Specificity = \frac{TN}{TN + FP} \tag{3}$$

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4}$$

$$g - mean = \sqrt{Specificity \ * \ Sensitivity} \tag{5}$$

F-measure combines recall and precision of a class giving an idea of its overall performance, while g-mean measures how well the classifier performs for both the classes. If the classifier is biased towards negative class, it will have low sensitivity and hence low g-mean. But if the classifier is able to classify both positive and negative instances well, specificity and sensitivity will be high and the g-mean will also be high.

B. Datasets Description

Fourteen datasets downloaded from UCI repository [26], with their imbalance ratios are displayed in Table I. Since very few two-class imbalanced data sets are publically available, class imbalance was created for multi-class datasets by treating one class as positive (minority), and all others as negative (majority). Similar strategy was followed in [27].

C. Training and Test Set Creation

For evaluation purposes the data set was randomly partitioned using stratified sampling into training and test set (2:1). Conclusions based on one such run may have low credibility. To beat this, thirty sets of training and test sets were created and experiments with each method were executed on each of these train-test pairs. All the measures presented in the next section are averaged over thirty runs for each method, and their variances are also computed.

V. RESULTS AND DISCUSSION

The three ensemble methods were tested on the datasets mentioned in Table I. LIBSVM implementation was used. Linear kernel with the value of cost parameter C set to 1000, was used to generate component classifiers. Each ensemble contained at most b (imbalance ratio) classifiers and majority voting was used for decision making. LFM uses an additional parameter, i.e, the selection criterion for including classifiers in the ensemble. We use precision $\geq 25\%$ as the threshold. The rationale for low value of the threshold is the lack of information in some of the data sets. Consider for example, GlassVWFP dataset which has only twelve positive examples in the training set of 140 instances. Though the imbalance is not very severe, it is the lack of information which leads to very low precision for all three methods (Table IV). In order to avoid adding another dimension to discussion by varying the selection criterion in LFM for different datasets, we chose to keep threshold for component selection uniformly low.

Consolidated results in Table IV show the measures calculated for the minority class. The following conclusions can be drawn from the results:

1) LFM improves accuracy of the classifier on all datasets. It is also evident that LFM reduces false positives significantly leading to improved precision. (with exception of GlassVWFP dataset) (Eq. 1). This improvement also positively influences the F-measure (Eq. 4). On the other hand, LFM reduces the number of true positives leading to lower TP rate or recall which also pulls down g-mean (Eq. 5). Further the extent of reduction in FP and TP varies in different datasets. We envisage that as the number of classifiers in ensemble is increased (currently it has been restricted to b), better results are achievable with LFM.
2) PARTEN reduces false negatives more effectively than UMjC and LFM, which explains higher recall of the minority class. PARTEN is also the winner in g-mean, which indicates the higher extent of recognition of minority and majority classes taken together. Usage of each majority class example for learning is attributed to better recognition of both classes.
3) UMjC performs almost as well as PARTEN in these experiments. Marginal under-performance of UMjC is due to the fact that while minority class is fully represented in each training set, some of the majority class instances may not be selected in any of the training sets.

Plotting TP rate vs FP rate in ROC space further clarifies difference in the performance of the three methods with respect to the minority class. Clustering of points in the top left corner in ROC space demonstrates effectiveness of the ensemble methods in general, for learning from imbalanced datasets. Visual inspection confirms that PARTEN and

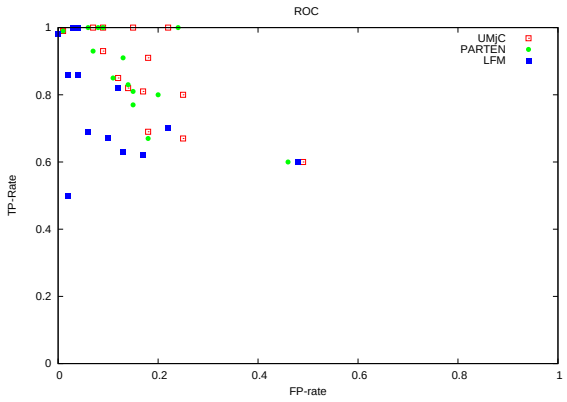

Figure 3. FP-rate vs TP-rate for three methods using 14 data sets

UMjC classifiers have higher TP rates compared to LFM classifiers which have lower FP rates. *Thus, LFM technique focuses on lowering FP-rate whereas PARTEN and UMjC give higher TP-rates for imbalanced data sets.*

Having analyzed the properties of the three ensemble methods, it is important to mention where is this tradeoff (between reduction in FP and TP) useful. Prediction of the minority (rare) class is only indicative, with confidence estimated by generalization error. In most practical applications predictions have to be inspected either manually (mostly in financial frauds, medical/bio applications). If the cost of inspection is very high (eg. as in genomic wet lab experiments), it makes sense to reduce false positives; particularly when we do not mind missing some TPs. In such cases, LFM delivers better cost effective solution. Please note that in financial fraud detection the opposite would be true. There the cost of missing TP (frauds) is much higher than the cost of detecting a FP (non-fraud). In such applications, PARTEN or UMjC should be preferred over LFM.

UMjC can be described as an approximation to PARTEN, as it produces almost the same results as PARTEN. In the cases where the size of datasets is too large, instead of generating b classifiers, lesser number of classifiers can be generated by UMjC to reduce the training as well as the prediction costs.

VI. Astronomy Applications

Astronomy is replete with datasets that contain rare objects mixed with more well-known types [28], [29]. Especially real-time transient astronomy, a still evolving field, requires that interesting objects be flagged within minutes of detection, the rarer ones be selected and followed up with larger telescopes as soon as possible [30], [31]. The training sets generally have object types forming rather imbalanced samples in terms of frequency, making them a suitable sample for techniques we are considering here.

In this section we have considered one application on artificial supernova data [32] with a large majority of the objects being SN of type Ia. We attempt detection of rarer supernovae from the spectroscopically confirmed subset of the SuperNova Challenge dataset [32]. The dataset contains observed flux for objects in 4 bands viz. g, r, i and z. We selected 270 objects that were present in all four bands.

In order to use the data we carried out the following operations: ignore points with errorbars greater than the flux values, convert the fluxes to magnitudes, obtain slopes for the curves by dividing the lightcurve into four parts, Thus, for each filter we ended up with four numbers. These 16 features were used. We separated type I SN from type II and used them. Ib and Ic were mixed with Ia due to the similarity of their light curves. IIn and IIp were tested separately as well as together. We chose to identify classes IIn (IR 22.5), IIp (IR 9) and IIp+IIn (IR 6.42).

Stratified sampling was used to create thirty training-test pairs. The three ensemble methods were applied and the results are displayed in Table V. The precision shows signicant improvement by LFM method. The improvement in precision is proportional to severity of imbalance. The improved precision is advantageous in being able to choose rarer classes, as followup is expensive. As we move into the domain of bigger surveys like LSST, it will just not be possible to follow-up all transients and such methods will be key to choice of objects. Also, a lower recall but higher value of F-measure by LFM method indicates that the LFM ensemble balances the precision and recall of the minority class better than other two methods.

VII. Conclusion

In this paper we compare three ensemble methods for learning from imbalanced datasets. Ensemble of disjunct partitions (PARTEN) and under-sampled majority class (UMjC) have been earlier shown to perform well for imbalanced datasets. In this paper Learning-from-mistakes (LFM) method has been described and empirically compared with the above mentioned two methods using public data sets. This method is shown to be effective in reducing the false positives, leading to improvement in precision of the minority(positive) class. It is found to best balance the precision and recall of the minority (positive) class. However, in applications where recognition of both classes is important it is advantageous to use PARTEN method. UMjC is close to PARTEN as far as recognition of both classes is concerned.

We further studied the performance of the three methods for detection of SNe of classes IIn, IIp and II. One of the issues in such applications is to reduce the number of false positives. LFM was found to achieve this effectively, while for effective prediction of both classes, PARTEN and UMjC performed equally well.

References

[1] N. V. Chawla, N. Japkowicz, and A. Kolcz, "Editorial : Special issue on learning from imbalanced data sets," *SIGKDD Explorations*, vol. 6, pp. 1–6, 2002.

[2] D. Lewis and J. Catlett, "Heterogeneous uncertainty sampling for supervised learning," *Proceedings of the International conference on Machine Learning (ICML'94)*, pp. 148–156, 1994.

[3] F. Provost, "Machine learning from imbalanced datasets 101/1," *AAAI Workshop on learning from Imbalanced Datasets Tech. Rep. WS-00-05*, 2000.

[4] "Special issue on learning from imbalanced data sets," *SIGKDD explorations*, vol. 6, 2002.

[5] Y. Liu, A. An, and X. Huang, "Boosting prediction accuracy on imbalanced datasets with SVM ensembles," *LNAI*, vol. 3918, pp. 107–118, 2006.

[6] R. Yan, Y. Liu, R. Jin, and A. Hauptmann, "On predicting rare classes with svm ensembles in scene classification," *IEEE international conference on acoustics, speech and signal processing(ICASSP'03)*, 2003.

[7] C. Drummond and R. Holte, "C4.5, class imbalance and cost sensitivity : Why under-sampling beats oversampling," *Workshop on Learning from Imbalanced datasets II held in conjunction with ICML'2003*, 2003.

[8] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets : One-sided selection," *Proceedings of 14th International Conference on Machine Learning*, pp. 179–186, 1997.

[9] F. Provost and T. E. Fawcett, "Robust classification for imprecise environments," *Machine Learning*, pp. 203–231, 2001.

[10] R. Barandela, J. Sanchez, V. Garcia, and E. Rangel, "Strategies for learning in class imbalance problems," *Pattern Recognition*, vol. 36, pp. 849–851, 2003.

[11] R. Akbani, S. Kwek, and N. Japkowicz, "Applying support vector machines to imbalanced datasets," *Lecture Notes in Computer Science (Springer Berlin / Heidelberg)*, vol. 3201, pp. 39–50, November 2004.

[12] V. Garcia, J. Sanchez, R. Mollineda, R. Alejo, and J. Sotoca, "The class imbalance problem in pattern classification and learning," *II Congreso Espaol de Informtica. IV Taller de Minera de Datos y Aprendizaje (TAMIDA 2007)*, September 2007.

[13] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas, "Handling imbalanced datasets : A review," *GESTS International Transactions on Computer Science and Engineering*, vol. 30, 2006.

[14] M. Bhardwaj, T. Gupta, T. Grover, and V. Bhatnagar, "An efficient classifier ensemble using SVM," *IEEE Xplore :Proceeding of International Conference on Methods and Models in Computer Science (ICM2CS 2009)*, 2009.

[15] S. Abe, *Support Vector Machines for Pattern Classification*. Springer-Verlag, 2005.

[16] C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.

[17] C. Chang and C. Lin, "Libsvm: A library for support vector machines," *Software : http://www.csie.ntu.edu.tw/ cjlin/libsvm/*.

[18] M. V. Joshi, V. Kumar, and R. C. Agarwal, "Evaluating boosting algorithms to classify rare classes: Comparison and improvements," *ICDM '01: Proceedings of the 2001 IEEE International Conference on Data Mining*, pp. 257–264, 2001.

[19] M. V. Joshi, R. C. Agarwal, and V. Kumar, "Predicting rare classes: can boosting make any weak learner strong?" *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 297–306, 2002.

[20] N. Japkowicz, "Learning from imbalanced data sets : A comparison of various strategies," *AAAI Technical Report WS-00-05*, 2005.

[21] S. Visa and A. Ralescu, "Issues in mining imbalanced data sets - a review paper," *Proceedings of the Sixteen Midwest Artificial Intelligence and Cognitive Science Conference*, 2005.

[22] G. M. Weiss, "Mining with rarity : A unifying framework," *SIGKDD Exploration Newsletter*, vol. 6(1), pp. 7–19, 2004.

[23] N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer, "Smote : Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research (JAIR)*, vol. 16, pp. 321–357, 2002.

[24] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Transactions on Systems, man and Cybernetics - Part B : Cybernetics*, vol. 39(2), pp. 539–550, 2009.

[25] Y. Dong and K. Han, "Text classification based on data partition and parameter varying ensembles," *SAC '05 : Proceedings of 2005 ACM Symposium on Applied Computing*, pp. 1044–1048, 2005.

[26] "UCI ML repository," *www.ics.uci.edu/ mlearn/MLRepository.html*.

[27] S. Garcia and F. Herrera, "Evolutionary undersampling for classification with imbalanced datasets : Proposals and taxonomy," *Evolutionary Computation*, vol. 17(3), pp. 275–306, 2009.

[28] K. D. Borne, *Scientific Data Mining in Astronomy*. Next Generation Data Mining, CRC Press, in press.

[29] N. Ball and R. Brunner, "Data mining and machine learning in astronomy," *International journal of Physics*, 2009.

[30] A. Mahabal, S. G. Djorgovski, R. Williams, A. Drake, C. Donalek, M. Graham, B. Moghaddam, M. Turmon, J. Jewell, A. Khosla, and B. Hensley, "Towards real-time classification of astronomical transients," *American Institute of Physics Conference Series*, vol. 1082, pp. 287–293, Dec. 2008.

[31] A. Mahabal, S. G. Djorgovski, M. Turmon, J. Jewell, R. R. Williams, A. J. Drake, M. G. Graham, C. Donalek, E. Glikman, and Palomar-QUEST team, "Automated probabilistic classification of transients and variables," *Astronomische Nachrichten*, vol. 329, pp. 288–291, 2008.

[32] "Supernova challenge data," *http://arxiv.org/abs/1001.5210*.

Table IV
PERFORMANCE MEASURES FOR MINORITY (POSITIVE) CLASS ON 14 DATASETS USING THE PARTEN, UMJC AND LFM. THE VALUES IN PARENTHESES ARE THE STANDARD DEVIATIONS OF THE MEASURES FROM THE AVERAGE VALUES.

Dataset	Ensemble	TP	FP	FN	TN	Acc	Precision	Recall	F-msr	gmean
Ecoli-imU	PARTEN	9.83(0.7)	14.5(3.79)	2.17(0.87)	86.5(3.79)	85.25(3.43)	0.41(0.06)	0.82(0.07)	0.55(0.06)	0.84(0.04)
	UMjC	10.67(1.15)	14.33(3.2)	1.33(1.5)	86.67(3.2)	86.14(2.67)	0.43(0.06)	0.89(0.1)	0.58(0.05)	0.87(0.05)
	LFM	9.67(1.65)	10.3(6.02)	2.33(1.65)	90.7(6.02)	88.82(4.73)	0.51(0.11)	0.81(0.14)	0.61(0.09)	0.84(0.08)
Optdigit0	PARTEN	183.33(1.27)	14.3(5.19)	1.67(1.27)	1674.7(5.19)	99.15(0.26)	0.93(0.01)	0.99(0.01)	0.96(0.01)	0.99(0)
	UMjC	184.17(1.15)	5.1(3.21)	0.83(1.15)	1683.9(3.21)	99.68(0.16)	0.97(0.02)	1(0.01)	0.98(0.01)	1(0)
	LFM	183.6(1.28)	1.83(1.18)	1.4(1.28)	1687.17(1.18)	99.83(0.08)	0.99(0.01)	0.99(0.01)	0.99(0)	1(0)
Voweldata	PARTEN	27.83(1.51)	24.63(6.28)	2.17(1.51)	275.37(6.28)	91.88(1.83)	0.54(0.06)	0.93(0.05)	0.68(0.05)	0.92(0.02)
	UMjC	29.83(0.46)	13.6(4.22)	0.17(0.46)	286.4(4.22)	95.83(1.3)	0.69(0.07)	0.99(0.02)	0.82(0.05)	0.97(0.01)
	LFM	28.5(2.36)	2.5(2.39)	1.5(2.36)	297.5(2.39)	98.79(0.74)	0.93(0.06)	0.95(0.08)	0.93(0.04)	0.97(0.04)
GlassVWFP	PARTEN	4.03(1.13)	30.63(5.74)	1.97(1.13)	35.37(5.74)	54.72(7.31)	0.12(0.03)	0.67(0.19)	0.2(0.04)	0.59(0.07)
	UMjC	4.1(1.54)	30.3(7.59)	1.9(1.54)	35.7(7.59)	55.28(10.35)	0.12(0.05)	0.68(0.26)	0.2(0.08)	0.59(0.13)
	LFM	0.6(0.93)	5.8(4.52)	5.4(0.93)	60.2(4.52)	84.44(5.98)	0.11(0.15)	0.1(0.16)	0.21(0.09)	0.18(0.24)
Abalone	PARTEN	9.7(1.39)	29.8(2.73)	4.3(1.39)	133.2(2.73)	80.73(1.52)	0.25(0.03)	0.69(0.1)	0.36(0.04)	0.75(0.05)
11-18	UMjC	8.47(1.87)	44.5(22.27)	5.53(1.87)	118.5(22.27)	71.73(11.62)	0.18(0.05)	0.6(0.13)	0.27(0.05)	0.65(0.03)
	LFM	3.7(2.78)	7.67(7.57)	10.3(2.78)	155.33(7.57)	89.85(3.06)	0.43(0.22)	0.26(0.2)	0.26(0.12)	0.46(0.17)
Ecoli	PARTEN	6.7(0.47)	13.03(2.77)	0.3(0.47)	92.97(2.77)	88.2(2.54)	0.35(0.06)	0.96(0.07)	0.51(0.06)	0.92(0.04)
OM	UMjC	7(0)	14.67(9.29)	0(0)	91.33(9.29)	87.02(8.22)	0.39(0.18)	1(0)	0.54(0.18)	0.93(0.05)
	LFM	6.37(0.61)	1.6(2.21)	0.63(0.61)	104.4(2.21)	98.02(1.84)	0.84(0.14)	0.91(0.09)	0.86(0.09)	0.95(0.04)
Abalone	PARTEN	8.63(0.93)	37.17(5.89)	2.37(0.93)	125.83(5.89)	77.28(3.13)	0.2(0.02)	0.78(0.08)	0.31(0.03)	0.78(0.04)
11-19	UMjC	6.87(2.33)	53.9(24.4)	4.13(2.33)	109.1(24.4)	66.65(12.92)	0.12(0.03)	0.62(0.21)	0.19(0.04)	0.62(0.08)
	LFM	1.47(2.4)	3.87(5.13)	9.53(2.4)	159.13(5.13)	92.3(1.91)	0.21(0.25)	0.13(0.22)	0.26(0.14)	0.23(0.27)
GlassCont	PARTEN	4.6(0.5)	7.7(2.91)	0.4(0.5)	59.3(2.91)	88.75(4.02)	0.4(0.11)	0.92(0.1)	0.55(0.11)	0.9(0.05)
	UMjC	4.7(0.47)	10.6(5.72)	0.3(0.47)	56.4(5.72)	84.86(8.05)	0.36(0.01)	0.94(0.09)	0.5(0.05)	0.89(0.07)
	LFM	3.6(0.7)	2.9(1.94)	1.4(0.7)	64.1(1.94)	94.03(2.85)	0.6(0.19)	0.72(0.13)	0.64(0.14)	0.83(0.08)
Abalone	PARTEN	11.93(1.14)	27.1(5.01)	2.07(1.14)	202.9(5.01)	88.05(1.71)	0.31(0.03)	0.85(0.08)	0.45(0.03)	0.87(0.03)
9-18	UMjC	9.77(1.65)	15.37(6.64)	4.23(1.65)	214.63(6.64)	91.97(2.37)	0.41(0.1)	0.7(0.12)	0.51(0.07)	0.8(0.06)
	LFM	5.5(2.43)	2.7(3.78)	8.5(2.43)	227.3(3.78)	95.41(1.02)	0.77(0.2)	0.39(0.17)	0.48(0.13)	0.61(0.14)
GlassTware	PARTEN	3(0)	16.23(4.58)	0(0)	52.77(4.58)	77.45(6.37)	0.17(0.05)	1(0)	0.28(0.07)	0.87(0.04)
	UMjC	2.67(0.48)	14.73(6.35)	0.33(0.48)	54.27(6.35)	79.07(9.16)	0.18(0.08)	0.89(0.16)	0.29(0.11)	0.83(0.11)
	LFM	2.87(0.35)	1.8(1.06)	0.13(0.35)	67.2(1.06)	97.31(1.26)	0.65(0.15)	0.96(0.12)	0.76(0.08)	0.96(0.06)
Yeast	PARTEN	4.6(1.19)	29(15.57)	2.4(1.19)	126(15.57)	80.62(9)	0.15(0.05)	0.66(0.17)	0.24(0.06)	0.72(0.06)
CYT-POX	UMjC	4.23(1.48)	6.97(5.56)	2.77(1.48)	148.03(5.56)	93.99(3.31)	0.51(0.28)	0.6(0.21)	0.49(0.15)	0.75(0.13)
	LFM	3.7(1.44)	0.57(0.82)	3.3(1.44)	154.43(0.82)	97.61(1.01)	0.89(0.15)	0.53(0.21)	0.64(0.18)	0.71(0.14)
YeastME2	PARTEN	12.97(1.59)	74.07(8.69)	4.03(1.59)	403.93(8.69)	84.22(1.71)	0.15(0.02)	0.76(0.09)	0.25(0.03)	0.8(0.05)
	UMjC	12.93(1.7)	62.5(13.27)	4.07(1.7)	415.5(13.27)	86.55(2.52)	0.17(0.03)	0.76(0.1)	0.28(0.04)	0.81(0.05)
	LFM	2.47(2.19)	6.5(6.13)	14.53(2.19)	471.5(6.13)	95.75(0.96)	0.32(0.26)	0.15(0.13)	0.21(0.1)	0.32(0.2)
YeastME1	PARTEN	15(0)	30.3(4.06)	0(0)	449.7(4.06)	93.88(0.82)	0.33(0.03)	1(0)	0.5(0.03)	0.97(0)
	UMjC	15(0)	39.47(7.44)	0(0)	440.53(7.44)	92.03(1.5)	0.28(0.04)	1(0)	0.44(0.04)	0.96(0.01)
	LFM	12.37(3.27)	17.6(8.96)	2.63(3.27)	462.4(8.96)	95.91(1.41)	0.45(0.14)	0.82(0.22)	0.55(0.11)	0.88(0.15)
YeastEXC	PARTEN	10.47(0.94)	65.57(11.11)	1.53(0.94)	417.43(11.11)	86.44(2.1)	0.14(0.01)	0.87(0.08)	0.24(0.02)	0.87(0.03)
	UMjC	10.3(1.12)	50.43(11.52)	1.7(1.12)	432.57(11.52)	89.47(2.27)	0.17(0.03)	0.86(0.09)	0.29(0.04)	0.88(0.05)
	LFM	6.2(2.64)	14.17(10.24)	5.8(2.64)	468.83(10.24)	95.97(1.69)	0.35(0.14)	0.52(0.22)	0.38(0.11)	0.69(0.17)

Table V
PERFORMANCE MEASURES FOR MINORITY CLASS ON SUPERNOVAE CHALLENGE DATASET USING PARTEN, UMJC AND LFM

+ vs - Class	Ensemble	TP	FP	FN	TN	Acc	Precision	Recall	F-msr	gmean
IIn vs all	PARTEN	3.6(0.67)	16.73(1.66)	0.4(0.67)	69.27(1.66)	80.96(1.72)	0.18(0.03)	0.9(0.17)	0.29(0.05)	0.85(0.08)
(IR 22.5)	UMjC	3.4(0.72)	20.07(6.14)	0.6(0.72)	65.93(6.14)	77.04(7.13)	0.16(0.05)	0.85(0.18)	0.26(0.08)	0.8(0.11)
	LFM	2.5(0.97)	2.03(1.79)	1.5(0.97)	83.97(1.79)	96.07(1.99)	0.63(0.24)	0.63(0.24)	0.62(0.17)	0.75(0.23)
IIp vs all	PARTEN	8.23(0.43)	6.93(2.26)	1.77(0.43)	73.07(2.26)	90.33(2.07)	0.55(0.06)	0.82(0.04)	0.66(0.04)	0.87(0.01)
(IR 9)	UMjC	8.67(0.66)	5.3(1.21)	1.33(0.66)	74.7(1.21)	92.63(1.18)	0.63(0.05)	0.87(0.07)	0.72(0.04)	0.9(0.03)
	LFM	8.13(0.73)	1.93(1.17)	1.87(0.73)	78.07(1.17)	95.78(1.64)	0.82(0.1)	0.81(0.07)	0.81(0.07)	0.89(0.04)
II vs all	PARTEN	13.3(0.6)	3.5(1.81)	0.7(0.6)	72.5(1.81)	95.33(2.07)	0.8(0.08)	0.95(0.04)	0.87(0.05)	0.95(0.02)
(IR 6.42)	UMjC	12.6791.06)	5.53(3.44)	1.33(1.06)	70.47(3.44)	92.37(3.81)	0.71(0.1)	0.9(0.08)	0.79(0.08)	0.91(0.04)
	LFM	11.33(1.35)	1.47(1.07)	2.67(1.35)	74.53(1.07)	95.41(1.69)	0.89(0.07)	0.81(0.1)	0.84(0.06)	0.89(0.05)