Pattern Recognition 45 (2012) 3738–3750

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Inverse random under sampling for class imbalance problem and its application to multi-label classification

Muhammad Atif Tahir [a,b,*], Josef Kittler [a], Fei Yan [a]

[a] Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford GU2 7XH, UK
[b] School of Computing, Engineering and Information Science, Northumbria University, Newcastle Upon Tyne, UK

ARTICLE INFO

Article history:
Received 21 October 2010
Received in revised form
8 March 2012
Accepted 21 March 2012
Available online 13 April 2012

Keywords:
Class imbalance problem
Multi-label classification
Inverse random under sampling

ABSTRACT

In this paper, a novel inverse random under sampling (IRUS) method is proposed for the class imbalance problem. The main idea is to severely under sample the majority class thus creating a large number of distinct training sets. For each training set we then find a decision boundary which separates the minority class from the majority class. By combining the multiple designs through fusion, we construct a composite boundary between the majority class and the minority class. The proposed methodology is applied on 22 UCI data sets and experimental results indicate a significant increase in performance when compared with many existing class-imbalance learning methods. We also present promising results for multi-label classification, a challenging research problem in many modern applications such as music, text and image categorization.

1. Introduction

Many real world classification problems are represented by highly unbalanced data sets, in which, 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 constructing a model that can successfully discriminate the minority samples from the majority samples. Generally, the problem of unbalanced data sets occurs when one class represents a rare or positive concept while the other class represents the negative concept, so that the examples from the negative class outnumber the examples from the positive class. This type of data is found, for example, in the multi-label classification problem where only few samples belong to the positive 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 unbalanced data sets reported in the literature [1–8]. The most commonly used methods to handle unbalanced data sets involve under sampling or over sampling 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. The most common strategy of class imbalance learning aims at improving the true positive rate (tpr) with the sacrifice of higher false positive rate (fpr).

In this paper, a novel inverse random under sampling (IRUS) method is proposed for the class imbalance problem in which the ratio of the majority and minority class cardinalities is inversed. This paper is an extension of shorter version of our work [9]. The main idea is to severely under sample the majority class multiple times with each subset having fewer examples than the minority class. For each training set we then find a decision boundary which separates the majority class from the minority class. 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 class training samples. Thus, each training set yields one classifier design. By combining the multiple designs through fusion, we construct a composite boundary between the majority class and the minority class. We shall argue that this boundary has the capacity to delineate the majority class more effectively than the solutions obtained by conventional learning. The proposed methodology is applied on 22 UCI data sets and experimental results indicate a significant increase in performance when compared with many existing class-imbalance learning methods. We also present promising results for multi-label classification, a challenging research problem in many modern applications such as music, text and image categorization.

This paper is organized as follows. Section 2 reviews several class imbalance methods. This is followed by introducing the proposed inverse random under sampling method (IRUS) in

* Corresponding author at: School of Computing, Engineering and Information Science, Northumbria University, Newcastle Upon Tyne, UK. Tel.: +44 1912437633.
E-mail addresses: m.tahir@surrey.ac.uk, muhammad.tahir@northumbria.ac.uk (M.A. Tahir), j.kittler@surrey.ac.uk (J. Kittler), f.yan@surrey.ac.uk (F. Yan).

0031-3203/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2012.03.014

Section 3. Section 4 describes the experimental setup followed by the presentation and discussion of the results in Section 5. Section 6 presents promising results for multi-label classification, a challenging research problem in many modern applications such as music, text and image categorization. The paper is drawn to conclusion in Section 7.

2. Related work

The most commonly used methods to handle unbalanced data sets involve under sampling or over sampling of the original data sets. He and Garcia [10] and Galar et al. [11] give a good summary for sampling methods. Random over sampling and random under sampling are the most popular non-heuristic methods that attempt to balance the class representation through random replication of the minority class and random elimination of majority class samples respectively. Both of these approaches have limitations. For instance, under-sampling can discard potentially useful data while over-sampling can increase the likelihood of over fitting [4]. Despite these limitations, random over sampling in general is among the most popular sampling techniques and provides competitive results when compared with more complex methods [4,8].

Several heuristic methods are proposed to overcome these limitations including Tomek's links [12], Condensed Nearest Neighbor Rule (CNN) [13], one-sided selection [1] and Neighborhood Cleaning rule (NCL) [14] are several well-known methods for under-sampling while Synthetic Minority Over-Sampling Technique (SMOTE) is a well-known method for over-sampling [2]. The main idea in SMOTE is to generate synthetic examples by operating in the "feature space" rather than the "data space" [2]. The minority class is oversampled by interpolating between several minority class examples that lie together. Depending upon the amount of over-sampling required, neighbors from the k nearest neighbors are randomly chosen. Thus, the over fitting problem is avoided and the decision boundaries for the minority class are spread further into the majority class space [4].

Liu et al. [8] and Chan et al. [15] examine the class imbalance problem by combining classifiers 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.

In addition to techniques involving over/under sampling of the original data sets, cost-sensitive learning methods are also quite popular to handle the class imbalance problem. The aim of the cost-sensitive learning is to bias existing classifier learning algorithms towards the minority class [3,16–18]. Ensemble methods such as bagging and boosting are also applied frequently for unbalanced data sets and good results are reported in [19–23,11].

3. Novel inverse random under sampling technique for class imbalance problem

In this section, we discuss the proposed inverse random under sampling (IRUS) method. We first briefly discuss Bagging since this is an important concept used in IRUS.

3.1. Bagging

Bagging proposed by Brieman [24] is the one of the most popular resampling ensemble method. It is a relatively simple idea: given a training set, bagging generates many bootstrap samples or training subsets. Each bootstrap sample is drawn by randomly generating subsets of samples where each sample is selected with replacement and equal probability. A prediction method e.g. decision tree or neural network is applied to each bootstrap sample to get base model. A bagged ensemble predicts a new sample by having each of its base models classify that example. The final prediction of the class is normally obtained by majority voting [25,26]. However, other combining rules such as mean, product and average are also used in bagging [27,21,23]. Diversity also plays an important role in improving the performance of ensemble classifier. The main idea in bagging is that the base models generated from the different bootstrapped training sets disagree often enough that the ensemble performs better than the base models and with the variance being reduced due to aggregation.

Friedman and Hall [28] proposed a version of bagging named subagging in which the bootstrap samples are generated without replacement. It has been shown that subagging can a viewed as a computationally cheaper version of bagging. In the proposed method, training subsets are generated without replacement. For simplicity, subagging is refereed as bagging throughout this paper.

3.2. Inverse random under sampling

A conventional training of a positive class using a data set containing representative proportions of samples from the positive and negative 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 negative class will be lower than the probability of error for the positive class. However, as we wish to retrieve samples from the positive class, we need to reverse the impact of the priors.

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 negative class of size proportional to P^2 where P is the prior probability of the positive class. This would lead to very small sample sets for the negative class and therefore, a poor definition of the boundary between the two classes. Nevertheless, the boundary would favor the positive class with high true positive rate (tpr) since the number of samples from the negative class are much less than the number of samples from the positive class. Further, since 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 $1/P^2$. This huge number of designs could then be used for controlling the false positive rate using a completely different mechanism. By combining the designed detectors using fusion, we can control the false positive error rate. In summary, the main idea behind IRUS is

- maintain a very high true positive rate (tpr) by *imbalance inversion* i.e. by making the majority (negative) class subsets have fewer examples than the minority (positive) class;
- then, control the false positive rate (fpr) by *classifier bagging* [24,29,28] i.e. by creating various subsets with each subset having all examples from the positive class and very few samples from the negative class.

Fig. 1 provides supporting evidence for the above idea. This graph shows the number of component classifiers vs {true positive rate (*tpr*), false positive rate (*fpr*)} during training in a real German data set (see Section 4 for details about this data set). It is clear from the graph that IRUS maintains very high and robust *tpr* in all iterations. False positive rate is then improved by creating various subsets with each subset having all samples from the minority class and few distinct samples from the majority class.

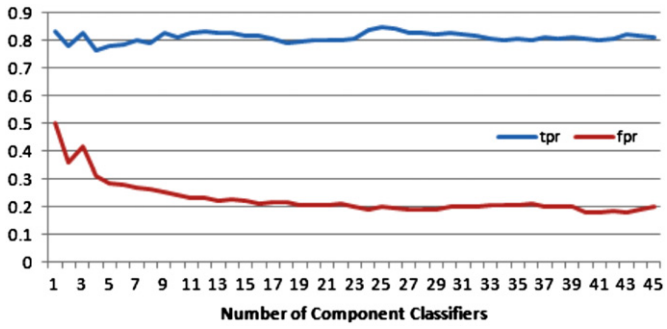

Fig. 1. True positive rate and false positive rate as a function of the number of component classifiers.

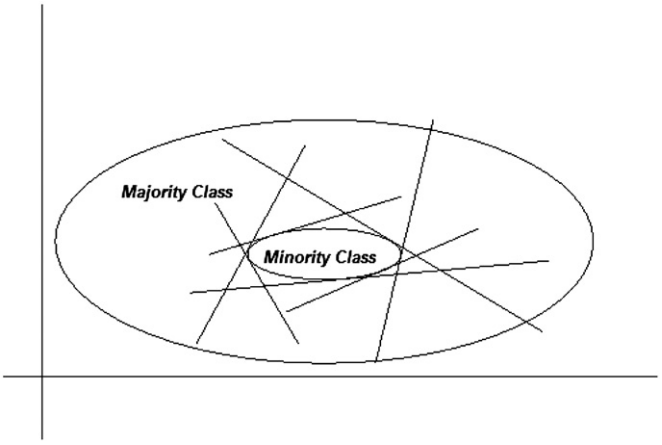

Fig. 2. Schematic diagram showing each boundary partitioning the training data set by a hyperplane tangent to the surface of the volume occupied by the minority class.

Interestingly, there is another important benefit of the IRUS method. As the number of samples forming the majority 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 Fig. 2. Each boundary partitions the training data set by a hyperplane tangent to the surface of the volume occupied by the positive class. It is the union of these tangent hyperplanes created by fusion, which constitutes a complex boundary to the positive class. Such boundary could not easily be found by a conventional learning. If one resorted to non-linear functions, the small sample set training would most likely lead to a over fitting and, consequently, to poor generalization on the test set.

In summary, we propose a classifier design approach which is based on an inverse imbalance sampling strategy. This is accomplished by combining the outputs of the multiple classifiers in the fusion stage. This method allows a very accurate definition of the boundary between the majority and the minority class.

The pseudo code of IRUS is shown in Algorithm 1. N_{min} is the number of minority examples and N_{maj} is the number of majority examples. S and *Sets* are user specified parameters. S controls the number of majority samples drawn at random for each model while *Sets* determines the number of models or classifiers. It is important to note that the positive class is now majority class since $S < N_{min}$. For each set $X'_{N_{maj}}$ paired with $X_{N_{min}}$, we learn a model h_i. From this model, probability/score is obtained for test sample t belonging to the positive class. This score is then normalized using z-score normalization. The purpose of the normalization is to produce classifier scores with the same mean and variance to maximize the effectiveness of classifier aggregation by a simple score averaging. Let A_i be the set of scores obtained from the training data of model i. A score a is normalized to a_{norm} by computing

$$a_{norm} = \frac{a - \mu(A_i)}{\sigma(A_i)} \tag{1}$$

where $\mu(A_i)$ and $\sigma(A_i)$ are the mean and standard deviation of A_i. The aggregation of the generated classifiers is then implemented by soft voting (MEAN rule) [30]. It should be noted that although many aggregation models have been proposed in the literature, we only focus on the MEAN rule due to its simplicity and good performance in many pattern classification problems. The output is the confidence score of test sample t belonging to positive class. Similarly, confidence scores are obtained for other test samples. These confidence scores are then directly used to evaluate performance measures like area under the ROC curve (AUC). For performance measures like F_1 and G-mean where predicted class is required, the decision is made in favor of positive class if $conf(t) > 0$ or $1/(1+\exp^{-conf(t)}) > 0.5$.

An important factor in determining whether IRUS will improve performance is the stability of the classifier for constructing base models in bagging. It has been shown in [29] that instability i.e. responsiveness to changes in the training data is quite important for bagging to be effective. In other words, improvement will occur for unstable methods where a small change in the training set can result in large changes in model. Brieman [29] mentioned that the classification trees, neural nets are unstable methods while k-nearest neighbor and linear discriminant analysis are stable methods. The unstable classifiers are distinguished by a high variance although they can have a low bias. On the contrary, stable classifiers have a low variance, but they can have a high bias. Since in IRUS, false positive rate (*fpr*) is controlled by bagging, it is recommended to use unstable methods such as neural nets, classification trees etc as base classifier.

Algorithm 1. Pseudo-code for inverse random under sampling (IRUS).

Require: $X_{N_{min}}$: Training set of minority patterns with cardinality N_{min}
$X_{N_{maj}}$: Training set of majority patterns with cardinality N_{maj}
S: Number of samples from $X_{N_{maj}}$ for each Model, $S < N_{min}$
Sets: Number of classifiers, Default:$1.5 \times ceil(N_{maj}/S)$
t: Test sample
Ensure: Confidence score of t, $conf(t)$
$conf(t) = 0$
for $i = 1$ to *Sets* **do**
$X'_{N_{maj}} \Leftarrow$ Randomly pick S samples without replacement from $X_{N_{maj}}$
$T_s \Leftarrow X'_{N_{maj}} \cup X_{N_{min}}$
Train base classifier h_i using T_s samples
D=Probability of positive class assigned by h_i to the test sample t
D_{norm}=z-score normalization of D (Eq. (1))
$conf(t) = conf(t) + D_{norm}$
end for
$conf(t) = conf(t)/Sets$

4. Experiments

4.1. Experimental setup

To evaluate the effectiveness of the proposed method, extensive experiments were carried out on 22 public data sets[1] from

[1] In shorter version of this work [9], categorical features were used to train the decision tree (C4.5) in Nursery data set. However, it was later observed that an

Table 1
A description of the data sets. Ratio is the size of the majority class divided by that of the minority class (N_{maj}/N_{min}).

Data sets	Samples N_s	Attributes A	Minority/majority	# min/# maj	Ratio
Abalone	4177	8	Age > 20/age < 20	36/4141	115.0
Arrhythmia	452	280	Class5/rest	13/439	33.8
Balance-scale	625	4	Balance/rest	49/576	11.8
Breast-cancer	286	10	Recurrence/no-recurrence	85/201	2.4
CMC	1473	9	Class2/rest	333/1140	3.4
Flag	194	28	White/rest	17/177	10.4
German	1000	20	Bad/good	300/700	2.33
Glass	214	9	Ve-win-float-proc/rest	17/197	11.6
Housing	506	13	[20–23]/rest	106/400	3.8
Haberman	306	3	Die/survive	81/225	2.8
Heart-statlog	270	14	Present/absent	120/150	1.3
Hepatitis	155	20	Die/live	32/123	3.8
Ionosphere	351	35	Bad/good	126/225	1.8
Mf-Mor	2000	6	10/rest	200/1800	9.0
Mf-Zer	2000	47	10/rest	200/1800	9.0
Nursery	12,960	8	Very-recom/rest	328/12,632	38.5
Phoneme	5404	5	1/0	1586/3818	2.4
Pima	768	8	1/0	268/500	1.9
Satimage	6435	36	4/rest	626/5809	9.3
Solar-Flare	1066	13	*F*/rest	43/1023	23.8
Vehicle	846	18	Opel/rest	212/634	3.0
Wpdc	198	33	Rec/non-rec	47/151	3.2

the UCI repository which have different degrees of imbalance [31]. Table 1 describes the data sets used in this study. For each data set, it shows the number of attributes (A), the number of samples (N_s), the number of majority samples (N_{maj}) and the number of minority samples (N_{min}). All the reported results are obtained by 10×10-Fold Cross Validation and the paired t-test is then used to determine their significance under a value of 0.05.

4.2. Benchmark methods

The proposed IRUS technique is compared with several class imbalance techniques including RUS, ROS, SMOTE, Chan, Easy-Ensemble and Asymmetric Bagging. In all methods, Decision tree (C4.5) is used as the base classifier. Since pruning can reduce the minority class coverage in the decision trees in highly unbalanced data sets [32,9], all the results reported are without pruning. Laplace smoothing which is often used to smooth the frequency-based estimates in C4.5 is used to estimate probabilities [32]. The WEKA [33] implementation (J48) is used for C4.5.

4.2.1. C4.5

C4.5 classifier [34]. It uses the entire data set to train a single classifier.

4.2.2. Random under sampling (RUS)

In this approach, the class representation is balanced in the training set through the random elimination of the majority class. C4.5 is then as classifier on under-sampled training subset.

4.2.3. Random over sampling (ROS)

The class representation is balanced in the training set through the random replication of the minority class. C4.5 is then used as classifier on over-sampled training subset.

4.2.4. SMOTHE (SM)

The minority class is oversampled by interpolating between several minority class examples that lie together [2]. The WEKA [33] implementation is used for SMOTE. k in the k nearest neighbor method is set to 5.

4.2.5. Chan and Stolfo's method (Chan)

This method splits majority class into several non-overlapping subsets [15]. C4.5 classifier is then trained from each of these subsets and samples from minority class. The aggregation of the generated classifiers is then implemented by soft voting (MEAN rule).

4.2.6. EasyEnsemble (Easy)

This method splits majority class into several independent subsets [8].[2] AdaBoost classifier is then trained from each of these subsets and samples from minority class. All generated classifiers are then combined for the final decision using Adaboost. C4.5 is used as weak classifier. Parameters T and s_i are set to 4 and 10 in EasyEnsemble as suggested in [8].

4.2.7. Asymmetric Bagging (Asym)

This method splits majority class into several independent subsets [21,22] but without replacement as in IRUS. C4.5 classifier is then trained from each of these subsets and samples from minority class. The aggregation of the generated classifiers is then implemented by soft voting (MEAN rule).

4.3. Performance measure

The area under the receiver operating characteristic curve (AUC), F_1 and G-mean are the most commonly used measures for evaluating the classification performance on class imbalance data sets [35,8] and are adopted here. The AUC represents the expected performance as a single scalar. It integrates the performance of the learning method over all possible values of the false positive rate. The Mann–Witney statistic is used to calculate AUC

(*footnote continued*)
almost perfect separation can be obtained if features are treated as continuous and using a single base classifier (C4.5). In the revised version, all results are reported by treating features as continuous instead of categorical in Nursery.

[2] Source code of EasyEnsemble is downloaded from http://lamda.nju.edu.cn/code_EasyEnsemble.ashx. Instead of CART, the decision trees were constructed using J48 (WEKA implementation of C4.5).

and is implemented in WEKA [33]. F_1 and G-mean are defined as follows:

$$\text{False positive rate } (fpr) = \frac{\text{false positive}}{\text{false positive} + \text{true negative}}$$

$$\text{True positiverate } (tpr, Acc^{+}, \text{recall}) = \frac{\text{true positive}}{\text{true positive} + \text{false negative}}$$

$$\text{True negative rate } (tnr, Acc^{-}) = \frac{\text{true negative}}{\text{true negative} + \text{false positive}}$$

$$G\text{-mean} = \sqrt{Acc^{+} \times Acc^{-}}$$

$$\text{Precision} = \frac{\text{true positive}}{\text{truepositive} + \text{false positive}}$$

$$F_1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

5. Results and discussion

This section presents the results obtained with our proposed method. The first part evaluates the sensitivity of the performance with respect to the parameters of the method and the second part compares the performance of IRUS to other techniques for class imbalance problems.

5.1. Evaluation of parameters

In this section, we discuss and evaluate two parameters, S and *Sets*, used in IRUS. S controls the number of majority samples drawn at random for each model while *Sets* determines the number of models. In order to make IRUS as general as possible, we suggest to use the following formulas to obtain S and *Sets*.

$$S = \max(2, ceil(pow(N_{min}, X))) \tag{2}$$

$$Sets = ceil\left(1.5 \times \frac{N_{maj}}{S}\right) \tag{3}$$

where N_{min} and N_{maj} are the number of minority and majority examples respectively. $X \in \{0.05, 0.10, \ldots, 0.95\}$ is selected via a 5-Fold Cross Validation of the training data. Thus, effectively, only one parameter, namely X, needs to be tuned from the training data while the main parameters S and *Sets* are directly calculated from X, N_{min} and N_{maj}. For $X=1$, the IRUS method is equivalent to Asymmetric Bagging.

Table 2 shows the different values of parameter X used with the various data sets. For demonstration purposes, this table also shows the optimal value of AUC obtained by selecting X directly from the test data instead of 5-Fold Cross Validation of the training data. It is clear from this table that for the majority of the data sets, AUC obtained is not significantly different from the optimal AUC while for nine data sets (Abalone, Arrhythmia, Ionosphere, Nursery, Phoneme, Satimage, Solar-Flare, Vehicle and Wpdc), the optimal AUC is achieved. Parameter S also affects the training time. For a low value of S, more sets or classifiers are needed (For example, 39 classifiers are used in Mfeat-Mor) while for a high value of S, a fewer classifiers are required (for example, only six classifiers are used for Phoneme).

5.2. Empirical evaluation of IRUS

In this section, we examine the performance of IRUS compared to other imbalance classifiers. Table 3 shows the results along with the standard deviation when AUC is used as an evaluation criterion. It is cleared from the table that there is a significant increase in performance when the proposed method is directly compared with C4.5 in majority of data sets. For example, approx. 46% improvement is observed in Mf-Mor. When IRUS is compared with existing imbalance techniques including state-of-the-art EasyEnsemble, the proposed method also exhibits good performance in many data sets. For example, approx. 7% improvement in Flag when compared with EasyEnsemble. However, EasyEnsemble also achieves good performance on data sets like German, Housing, Ionosphere, Phoneme, and Satimage when compared with IRUS. This can be explained by the fact that EasyEnsemble benefits from the combination of boosting and bagging while IRUS focuses on bagging only. Future work aims to further enhance the performance of IRUS by investigating techniques for combining boosting and bagging. Overall, the average AUC for IRUS is approximately 10% better than its base classifier C4.5 and 0.6% better when compared with state-of-art method EasyEnsemble classifier.

Table 2
Evaluation of parameter X. For demonstration purposes only, $X_{test\text{-}set} \in \{0.05, 0.10, \ldots, 0.95\}$ is selected directly using test-data.

Data sets	N_{maj}	N_{min}	X	S	Sets	AUC	$X_{test\text{-}set}$	$AUC_{optimal}$
Abalone	4141	36	0.95	31	201	**0.855**	0.95	**0.855**
Arrhythmia	439	13	0.95	12	55	**0.977**	0.95	**0.977**
Balance-scale	576	49	0.25	3	288	0.588	0.40	0.621
Breast-cancer	201	85	0.55	12	26	0.661	0.60	0.662
Cmc	1140	333	0.60	33	52	0.736	0.65	0.736
Flag	177	17	0.65	7	38	0.804	0.60	0.804
German	700	300	0.55	24	44	0.766	0.50	0.769
Glass	197	17	0.90	13	23	0.803	0.95	0.808
Haberman	225	81	0.50	9	38	0.673	0.60	0.697
Heart-statlog	150	120	0.45	9	25	0.888	0.30	0.892
Hepatitis	123	32	0.80	16	12	0.838	0.70	0.845
Housing	400	106	0.50	11	55	0.811	0.55	0.813
Ionosphere	225	126	0.80	48	8	**0.954**	0.80	**0.954**
Mf-Mor	1800	200	0.80	70	39	0.930	0.75	0.931
Mf-Zer	1800	200	0.45	11	246	0.907	0.50	0.907
Nursery	12,632	328	0.95	246	78	**0.999**	0.95	**0.999**
Phoneme	3818	1586	0.95	1098	6	**0.923**	0.95	**0.923**
Pima	500	268	0.65	38	20	0.812	0.60	0.816
Satimage	5809	626	0.85	239	37	**0.951**	0.85	**0.951**
Solar-Flare	1023	43	0.95	36	43	**0.898**	0.95	**0.898**
Vehicle	647	199	0.85	90	11	**0.853**	0.85	**0.853**
Wpdc	151	47	0.25	3	76	**0.732**	0.25	**0.732**

Table 3
Comparison of IRUS with other methods when AUC is used as an evaluation measure. ∗ means significantly better than all other methods except those which are marked as +. Number in brackets show the standard deviation.

Data sets	C4.5	RUS	ROS	SM	Chan	Easy	Asym	IRUS
Abalone	0.710	0.735	0.795	0.794	0.856	**0.861**∗	0.858+	0.855+
	(0.023)	(0.037)	(0.020)	(0.016)	(0.007)	(0.009)	(0.008)	(0.008)
Arrhythmia	0.895	0.874	0.934	0.901	0.973	0.969+	0.974+	**0.977**∗
	(0.018)	(0.058)	(0.022)	(0.027)	(0.005)	(0.010)	(0.006)	(0.006)
Balance-scale	0.500	0.512	**0.624**∗	0.540	0.543	0.607+	0.555	0.588
	(0.000)	(0.030)	0.026)	(0.026)	(0.048)	(0.018)	(0.024)	(0.018)
Breast-cancer	0.614	0.602	0.613	0.572	0.644+	0.652	0.636	**0.661**∗
	(0.024)	(0.036)	(0.021)	(0.026)	(0.022)	(0.010)	(0.015)	(0.011)
Cmc	0.680	0.669	0.664	0.698	0.711	0.706	0.714	**0.736**∗
	(0.010)	(0.017)	(0.013)	(0.007)	(0.010)	(0.005)	(0.006)	(0.005)
Flag	0.709	0.758+	0.729	0.693	**0.805**∗	0.751	0.792+	0.804+
	(0.050)	(0.060)	(0.042)	(0.049)	(0.029)	(0.048)	(0.025)	(0.028)
German	0.702	0.695	0.705	0.714	0.723	**0.781**∗	0.720	0.766
	(0.007)	(0.009)	(0.017)	(0.011)	(0.014)	(0.009)	(0.013)	(0.006)
Glass	0.643	0.708	0.766	0.782+	0.796+	0.739	**0.805**∗	0.803+
	(0.024)	(0.060)	(0.033)	(0.033)	(0.033)	(0.032)	(0.029)	(0.027)
Haberman	0.618	0.610	0.637	0.673+	0.669+	**0.681**∗	0.665+	0.673
	(0.010)	(0.029)	(0.021)	(0.013)	(0.014)	(0.024)	(0.014)	(0.025)
Heart-statlog	0.848	0.842	0.844	0.852	0.851	0.884+	0.837	**0.888**∗
	(0.015)	(0.008)	(0.014)	(0.015)	(0.012)	(0.011)	(0.019)	(0.009)
Housing	0.749	0.741	0.761	0.766	0.793	**0.817**∗	0.786	0.811+
	(0.011)	(0.020)	(0.009)	(0.011)	(0.010)	(0.013)	(0.015)	(0.005)
Hepatitis	0.794	0.788	0.771	0.774	0.825	**0.848**∗	0.835+	0.838
	(0.036)	(0.044)	(0.031)	(0.027)	(0.015)	(0.024)	(0.021)	(0.022)
Ionosphere	0.925	0.930	0.938	0.930	0.944	**0.974**∗	0.928	0.954
	(0.012)	(0.010)	(0.012)	(0.014)	(0.011)	(0.003)	(0.009)	(0.006)
Mf-Mor	0.500	0.928	0.923	0.927	0.928	0.919	0.928	**0.930**∗
	(0.000)	(0.001)	(0.004)	(0.003)	(0.001)	(0.004)	(0.001)	(0.002)
Mf-Zer	0.867	0.866	0.877	0.875	0.900	0.902	0.899	**0.907**∗
	(0.007)	(0.007)	(0.005)	(0.006)	(0.004)	(0.005)	(0.003)	(0.002)
Nursery	**1.000**∗	0.981	0.998	**1.000**∗	0.999	0.999	0.999	0.999
	(0.000)	(0.004)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)
Phoneme	0.915	0.896	0.925	0.920	0.926	**0.956**∗	0.926	0.923
	(0.003)	(0.006)	(0.002)	(0.003)	(0.002)	(0.001)	(0.001)	(0.002)
Pima	0.779	0.763	0.778	0.777	0.797	0.809+	0.769	**0.812**∗
	(0.014)	(0.011)	(0.011)	(0.011)	(0.011)	(0.010)	(0.014)	(0.007)
Satimage	0.916	0.911	0.921	0.921	0.947	**0.956**∗	0.949	0.951
	(0.003)	(0.003)	(0.002)	(0.004)	(0.002)	(0.001)	(0.001)	(0.001)
Solar-Flare	0.839	0.858	0.856	0.880	0.900+	0.896+	**0.901**∗	0.898+
	(0.025)	(0.009)	(0.011)	(0.008)	(0.004)	(0.010)	(0.004)	(0.003)
Vehicle	0.820	0.785	0.823	0.819	0.834	**0.860**∗	0.833	0.853+
	(0.010)	(0.015)	(0.008)	(0.016)	(0.006)	(0.007)	(0.007)	(0.006)
Wpdc	0.633	0.642	0.656	0.682	0.698	0.699	0.695+	**0.732**∗
	(0.043)	(0.059)	(0.031)	(0.028)	(0.033)	(0.040)	(0.038)	(0.022)
Average	0.757	0.777	0.797	0.795	0.821	0.830	0.819	**0.835**

Tables 4 and 5 show the comparison of IRUS when F_1 and G-mean are used as evaluation criteria. Again it is observed that the proposed IRUS method compares favorably with other methods. Overall, the average F_1 and G-mean for IRUS are approximately 23.6% and 38.7% respectively better than its base classifier C4.5 and 1.0% and 1.3% better when compared with state-of-the-art EasyEnsemble classifier. However, it is observed that the proposed method consistently performs poorly in Phoneme and Nursery even with its base classifier when F_1 is used as the evaluation measure. This can be explained by the fact that although IRUS has maintained a very high true positive rate (tpr) in these data sets, this is achieved only at the cost of high false positive rate (fpr) and a low precision when compared with its base classifier as shown in Fig. 3.

Tables 6–8 show the results of t-test (significance level 0.05) using AUC, F_1 and G-mean respectively indicating WIN–TIE–LOSE. The paired t-test reveals that IRUS is superior in the majority of data sets when compared with other methods including state-of-the-art EasyEnsemble. For example, when IRUS is compared with EasyEnsemble with AUC as evaluation measure (Table 6), the number of WINs is 7, the number of TIEs is 10 and the number of LOSSes is 5. However, in some papers, these counts can also be used for further statistical test such as sign test [36]. Sign test indicates the probability of obtaining the observed record of wins to losses, or more extreme, by chance. If the sign test is significantly low then it is sufficient to conclude that it is likely that the outcome was not obtained by chance. The comparison of various methods using sign test reveals that IRUS is able to outperform all the algorithms with a confidence level of 10% with the exception of EasyEnsemble in all evaluation criteria and Asym for G-mean. However, IRUS offers a potential computational advantage over EasyEnsemble since all base classifiers in IRUS can be trained independently and thus parallel execution is applicable. We have also used another comparative statistic, the geometric mean of the performance ratio of every pair of method which should be considered to give a general view of the trends of the relative performance of the classifiers [36]. Lets assume that the performance measure is AUC, then for two methods m_1 and m_2, with AUCs $a_1^1, a_2^1, \ldots, a_n^1$ and $a_1^2, a_2^2, \ldots, a_n^2$ respectively for n data sets, the geometric mean of the AUC ratio is $\dot{r} = (\prod_{i=1}^{n} a_i^1/a_i^2)^{1/n}$.

Table 4
Comparison of IRUS with other methods when F_1 is used as an evaluation measure. ∗ means significantly better than all other methods except those which are marked as +.

Data sets	C4.5	RUS	ROS	SM	Chan	Easy	Asym	IRUS
Abalone	0.045$^+$	0.040	0.058$^+$	**0.066***	0.053$^+$	0.057$^+$	0.055$^+$	0.055$^+$
	(0.039)	(0.006)	(0.028)	(0.020)	(0.002)	(0.004)	(0.002)	(0.002)
Arrhythmia	0.283	0.314	**0.400***	0.274	0.352	0.288	0.353$^+$	0.373$^+$
	(0.081)	(0.084)	(0.056)	(0.130)	(0.015)	(0.041)	(0.025)	(0.027)
Balance-scale	0.000	0.143	0.147	0.012	0.154$^+$	**0.172***	0.155	0.164$^+$
	(0.000)	(0.018)	(0.012)	(0.012)	(0.021)	(0.010)	(0.014)	(0.015)
Breast-cancer	0.395	0.434	0.425	0.386	0.468$^+$	**0.472***	0.457$^+$	0.471$^+$
	(0.024)	(0.031)	(0.026)	(0.044)	(0.025)	(0.019)	(0.020)	(0.023)
Cmc	0.370	0.428	0.397	0.421	0.461	0.452	0.464	**0.484***
	(0.022)	(0.016)	(0.017)	(0.014)	(0.007)	(0.008)	(0.006)	(0.006)
Flag	0.215	0.267$^+$	0.218	0.155	0.267$^+$	0.274$^+$	0.278$^+$	**0.289***
	(0.083)	(0.049)	(0.047)	(0.047)	(0.029)	(0.026)	(0.041)	(0.034)
German	0.457	0.503	0.474	0.489	0.536	**0.590***	0.530	0.589$^+$
	(0.012)	(0.007)	(0.026)	(0.013)	(0.016)	(0.018)	(0.019)	(0.010)
Glass	0.048	0.228	0.164	0.233$^+$	0.284$^+$	0.237	**0.289***	0.282$^+$
	(0.050)	(0.046)	(0.066)	(0.069)	(0.023)	(0.042)	(0.023)	(0.019)
Haberman	0.402	0.441$^+$	0.458	0.477$^+$	0.478$^+$	0.469$^+$	**0.482***	0.472$^+$
	(0.019)	(0.046)	(0.020)	(0.016)	(0.017)	(0.027)	(0.025)	(0.023)
Heart-statlog	0.737	0.736	0.729	0.746	0.750	**0.783***	0.726	0.772
	(0.020)	(0.017)	(0.009)	(0.017)	(0.020)	(0.016)	(0.025)	(0.008)
Housing	0.218)	0.498	0.453	0.486	0.528	0.537	0.526	**0.550***
	(0.051)	(0.022)	(0.033)	(0.020)	(0.010)	(0.013)	(0.014)	(0.014)
Hepatitis	0.440	0.526	0.549	0.450	0.562	**0.611***	0.591$^+$	0.600$^+$
	(0.082)	(0.032)	(0.049)	(0.074)	(0.029)	(0.019)	(0.044)	(0.034)
Ionosphere	0.851	0.848	0.859	0.825	0.862	**0.902***	0.852	0.875
	(0.019)	(0.008)	(0.017)	(0.014)	(0.012)	(0.008)	(0.017)	(0.009)
Mf-Mor	0.000	0.641	0.623	0.635	0.642	0.628	0.643$^+$	**0.644***
	(0.000)	(0.003)	(0.005)	(0.005)	(0.002)	(0.005)	(0.002)	(0.002)
Mf-Zer	0.180	0.465	0.454	0.464	0.514	0.553	0.518	**0.566***
	(0.027)	(0.015)	(0.019)	(0.020)	(0.007)	(0.011)	(0.006)	(0.006)
Nursery	0.983	0.559	0.890	**0.990***	0.691	0.770	0.682	0.643
	(0.004)	(0.048)	(0.004)	(0.003)	(0.007)	(0.017)	(0.015)	(0.019)
Phoneme	0.774	0.748	0.788	0.774	0.773	**0.819***	0.775	0.769
	(0.004)	(0.007)	(0.004)	(0.004)	(0.004)	(0.003)	(0.003)	(0.004)
Pima	0.617	0.634	0.625	0.639	0.642	0.656$^+$	0.631	**0.667***
	(0.017)	(0.017)	(0.016)	(0.015)	(0.019)	(0.012)	(0.015)	(0.010)
Satimage	0.557	0.475	0.558	**0.573***	0.544	0.568$^+$	0.547	0.567$^+$
	(0.012)	(0.009)	(0.010)	(0.012)	(0.004)	(0.005)	(0.005)	(0.003)
Solar-Flare	0.150	0.285	0.281	0.310$^+$	0.319$^+$	0.256	**0.320***	0.314
	(0.047)	(0.016)	(0.014)	(0.040)	(0.005)	(0.010)	(0.007)	(0.005)
Vehicle	0.534	0.564	0.550	0.578	0.608	**0.640***	0.607	0.636$^+$
	(0.022)	(0.019)	(0.018)	(0.023)	(0.018)	(0.012)	(0.011)	(0.007)
Wpdc	0.337	0.426	0.389	0.457	0.467	0.438	0.457	**0.505***
	(0.051)	(0.048)	(0.050)	(0.020)	(0.023)	(0.024)	(0.044)	(0.028)
Average	0.391	0.464	0.477	0.475	0.498	0.508	0.497	**0.513**

Considering this statistical measure, Tables 6–8 show that IRUS also performs better when compared with all other methods as values of $\dot{r}$ are above 1 for all the evaluations.

Fig. 4 shows the different values of run-time parameter X vs AUC in the haberman and Wpdc data sets. It is observed that optimal performance is obtained when $X < 1$. This supports our earlier argument that severely under sampling the majority class can help to delineate the majority class very effectively.

6. Case study: multi-label classification

The most extensive application for our proposed technique is multi-label classification. In contrast to conventional multi-class classification systems where each instance only belongs to single label from a set of disjoint labels, in multi-label classification, each instance may belong to more than one class. There is a considerable amount of research concerned with the development of "good" multi-label learning methods [37–42]. Despite the extensive research effort, there exist many scientific challenges. They include highly unbalanced training sets, as for some labels, only limited data is available, and capturing correlation among classes. Interestingly, most state-of-the-art multi-label methods are designed to focus mainly on the second problem and a very limited effort has been devoted to handling unbalanced data populations [43,44]. In this paper, we focus on the first problem of multi-label learning, and tackle highly unbalanced data distributions using the proposed inverse random under sampling method.

The sparse literature on multi-label classification, driven by problems in text classification, bioinformatics, music categorization, and image/video classification, has recently been summarized by Tsoumakas et al. [37]. Multi-label classification can be reduced to the conventional binary classification problem. This approach is referred to as *binary relevance* (BR) learning in the literature. In BR learning, the original data set is divided into $|Y|$ data sets where $Y = \{1, 2, \ldots, N\}$ is the finite set of labels. BR learns one binary classifier $h_a : X \rightarrow \{\neg a, a\}$ for each class $a \in Y$. BR learning is theoretically simple and has a linear complexity with respect to the number of labels. Its assumption of label independence makes it attractive to situations where new examples may

Table 5
Comparison of IRUS with other methods when *G*-mean is used as an evaluation measure. ∗ means significantly better than all other methods except those which are marked as +.

Data sets	C4.5	RUS	ROS	SM	Chan	Easy	Asym	IRUS
Abalone	0.065	0.686	0.139	0.192	0.762	0.773	**0.779***	0.774+
	(0.053)	(0.052)	(0.058)	(0.064)	(0.015)	(0.032)	(0.017)	(0.016)
Arrhythmia	0.338	0.815	0.519	0.370	**0.939***	0.857	0.928+	0.931+
	(0.105)	(0.105)	(0.083)	(0.151)	(0.002)	(0.076)	(0.027)	(0.027)
Balance-scale	0.000	0.169	0.351	0.031	0.505	**0.552***	0.514	0.536
	(0.000)	(0.145)	(0.053)	(0.030)	(0.047)	(0.026)	(0.030)	(0.029)
Breast-cancer	0.528	0.562	0.559	0.526	0.589+	0.594+	0.580+	**0.599***
	(0.021)	(0.029)	(0.024)	(0.039)	(0.024)	(0.018)	(0.018)	(0.021)
Cmc	0.541	0.620	0.590	0.599	0.652	0.644	0.655	**0.673***
	(0.021)	(0.015)	(0.015)	(0.012)	(0.007)	(0.008)	(0.005)	(0.006)
Flag	0.267	0.615+	0.345	0.227	**0.641***	0.627+	0.640+	0.637+
	(0.106)	(0.077)	(0.064)	(0.072)	(0.065)	(0.057)	(0.089)	(0.077)
German	0.589	0.627	0.605	0.618	0.655	**0.704***	0.650	0.702+
	(0.010)	(0.007)	(0.022)	(0.010)	(0.015)	(0.016)	(0.017)	(0.009)
Glass	0.064	0.583	0.235	0.398	0.711+	0.601	**0.737***	0.707+
	(0.062)	(0.103)	(0.098)	(0.109)	(0.061)	(0.078)	(0.035)	(0.057)
Haberman	0.544	0.581	0.602	0.619+	0.626+	0.615+	**0.631***	0.613+
	(0.026)	(0.052)	(0.018)	(0.016)	(0.016)	(0.025)	(0.022)	(0.027)
Heart-statlog	0.759	0.755	0.749	0.765	0.768	**0.800***	0.744	0.795+
	(0.020)	(0.018)	(0.008)	(0.016)	(0.018)	(0.016)	(0.022)	(0.007)
Housing	0.335	0.700	0.629	0.677	0.729	0.733+	0.727	**0.744***
	(0.057)	(0.021)	(0.030)	(0.020)	(0.010)	(0.012)	(0.013)	(0.012)
Hepatitis	0.554	0.712	0.694	0.574	0.732+	**0.771***	0.760+	0.764
	(0.102)	(0.027)	(0.048)	(0.081)	(0.032)	(0.019)	(0.034)	(0.028)
Ionosphere	0.877	0.880	0.887	0.863	0.885	**0.919***	0.881	0.896
	(0.014)	(0.006)	(0.014)	(0.011)	(0.011)	(0.007)	(0.013)	(0.007)
Mf-Mor	0.000	0.923	0.908	0.921	0.924+	0.917	**0.925***	**0.925***
	(0.000)	(0.002)	(0.005)	(0.003)	(0.001)	(0.005)	(0.000)	(0.000)
Mf-Zer	0.315	0.828	0.730	0.749	0.859+	**0.866***	0.861+	0.861+
	(0.043)	(0.011)	(0.020)	(0.020)	(0.005)	(0.007)	(0.004)	(0.005)
Nursery	0.989	0.977	0.997	**0.999***	0.988	0.992	0.988	0.985
	(0.003)	(0.005)	(0.000)	(0.002)	(0.000)	(0.001)	(0.001)	(0.001)
Phoneme	0.840	0.840	0.859	0.860	0.859	**0.890***	0.861	0.857
	(0.004)	(0.005)	(0.003)	(0.003)	(0.004)	(0.002)	(0.002)	(0.003)
Pima	0.697	0.707	0.704	0.708	0.717	0.729+	0.706	**0.738***
	(0.014)	(0.015)	(0.013)	(0.014)	(0.016)	(0.010)	(0.015)	(0.009)
Satimage	0.726	0.822	0.733	0.770	0.873	**0.886***	0.875	0.876
	(0.007)	(0.007)	(0.008)	(0.008)	(0.003)	(0.003)	(0.003)	(0.003)
Solar-Flare	0.212	0.799	0.677	0.512	0.859+	0.824	**0.862***	0.851
	(0.063)	(0.031)	(0.024)	(0.055)	(0.007)	(0.014)	(0.009)	(0.006)
Vehicle	0.667	0.719	0.687	0.724	0.757	**0.783***	0.758	0.782+
	(0.022)	(0.016)	(0.016)	(0.018)	(0.015)	(0.010)	(0.009)	(0.007)
Wpdc	0.459	0.583	0.536	0.623	0.630	0.601	0.621	**0.659***
	(0.058)	(0.047)	(0.061)	(0.017)	(0.024)	(0.031)	(0.040)	(0.027)
Average	0.471	0.705	0.624	0.606	0.757	0.758	0.758	**0.768**

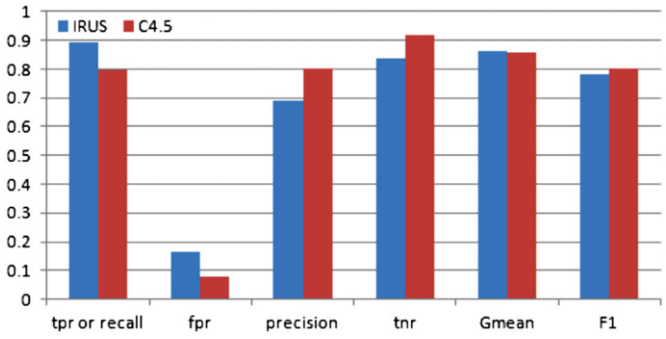

Fig. 3. Graph showing {tpr, fpr, precision, tnr, *G*-mean, F_1} for IRUS and C4.5 in Phoneme.

not be relevant to any known subset of labels or where label relationships may change over the test data [45]. Due to its simplicity, BR relevance approach has been adopted in this paper and a binary classifier for each class is trained using the proposed IRUS method and also other class imbalance methods.

Multi-label classification can also be reduced to the conventional classification problem by considering each unique set of labels as one of the classes. This approach is referred to as *label powerset* (LP) in the literature. However, this approach leads to a large number of label subsets with the majority of them with a very few examples and it is also computationally expensive. Like *one-vs-all* approach in conventional BR learning, the binary pairwise *one-vs-one* approach has also been employed for multi-label classification, therefore requiring $|Y|^2$ classifiers as opposed to $|Y|$. Calibrated label ranking (CLR) [40] is an efficient pairwise approach to multi-label classification.

The proposed BR-IRUS approach is applied to three publicly available multi-label data sets (emotions, scene, and yeast). The scene data set is concerned with semantic indexing of images of still scenes [38]. For Scene, the feature vector consists of 294 dimensions computed as spatial color moments in the *LUV* space. The image is divided into 49 blocks using a 7×7 grid. The first and second moment (mean and variance) are computed for each band. The end result is a $49 \times 2 \times 3 = 294$ dimensional feature vector. The "yeast" data set contains 14 functional classes of 2417 genes of yeast *Saccharomyces cerevisiae* [46] where each gene is

Table 6
Statistical comparison of algorithms using AUC as evaluation measure. *s* indicates the frequency of WIN–TIE–LOSE of the method in a row compared to the method in a column, *p* shows the *p*-value of the sign test and $\dot{r}$ demonstrates the geometric mean of the AUC ratio.

	C4.5	RUS	ROS	SMOTE	Chan	Easy	Asym	IRUS
Mean	0.757	0.777	0.797	0.795	0.821	0.830	0.819	**0.835**
RUS								
s	2-15-5							
p	0.453							
$\dot{r}$	1.031							
ROS								
s	11-9-2	12-9-1	–	–	–	–	–	–
p	0.022	0.003						
$\dot{r}$	1.060	1.029						
SMOTE								
s	12-9-1	14-6-2	5-13-4	–	–	–	–	–
p	0.003	0.004	1.000					
$\dot{r}$	1.056	1.024	0.996					
Chan								
s	20-1-1	18-4-0	18-3-1	13-8-1	–	–	–	–
p	0.000	0.000	0.000	0.002				
$\dot{r}$	1.092	1.059	1.030	1.034				
Easy								
s	21-0-1	18-3-1	18-3-1	17-2-3	10-9-3	–	–	–
p	0.000	0.000	0.000	0.003	0.002			
$\dot{r}$	1.105	1.072	1.043	1.047	1.012			
Asym								
s	18-2-2	16-6-0	17-3-2	13-8-1	1-18-3	4-8-10	–	–
p	0.000	0.000	0.001	0.002	0.625	0.180		
$\dot{r}$	1.088	1.056	1.027	1.031	0.997	0.985		
IRUS								
s	21-0-1	21-1-0	20-0-2	19-2-1	13-8-1	7-10-5	12-8-2	–
p	0.000	0.000	0.000	0.000	0.002	0.774	0.013	
$\dot{r}$	1.111	1.079	1.049	1.053	1.018	1.006	1.021	

Table 7
Statistical comparison of algorithms using F_1 as evaluation measure. *s* indicates the frequency of WIN–TIE–LOSE of the method in a row compared to the method in a column, *p* shows the *p*-value of the sign test and $\dot{r}$ demonstrates the geometric mean of the F_1 ratio. For C4.5 vs other methods, $*$ means $\dot{r}$ is calculated without considering scores from balance-scale and Mf-Mor (F_1 is 0 for C4.5 in these data sets).

	C4.5*	RUS	ROS	SMOTE	Chan	Easy	Asym	IRUS
Mean	0.391	0.464	0.477	0.475	0.498	0.508	0.497	**0.513**
RUS								
s	13-6-3		–	–	–	–	–	–
p	0.021							
$\dot{r}$	1.242							
ROS								
s	12-9-1	4-13-5	–	–	–	–	–	–
p	0.003	1.000						
$\dot{r}$	1.269	1.020						
SMOTE								
s	14-7-1	4-11-7	9-7-6	–	–	–	–	–
p	0.000	0.549	0.607					
$\dot{r}$	1.276	0.914	0.896					
Chan								
s	15-5-2	15-7-0	12-6-4	11-9-2	–	–	–	–
p	0.002	0.000	0.077	0.022				
$\dot{r}$	1.362	1.091	1.070	1.193				
Easy								
s	18-3-1	15-5-2	17-2-3	13-6-3	9-7-6	–	–	–
p	0.000	0.002	0.003	0.021	0.607			
$\dot{r}$	1.365	1.097	1.076	1.201	1.006			
Asym								
s	16-5-1	14-8-0	12-7-3	11-9-2	1-20-1	5-7-10	–	–
p	0.000	0.000	0.035	0.022	1.000	0.302		
$\dot{r}$	1.366	1.094	1.073	1.197	1.003	0.997		
IRUS								
s	18-2-2	19-3-0	16-4-2	14-6-2	13-7-2	9-10-3	11-8-3	–
p	0.000	0.000	0.001	0.004	0.007	0.146	0.057	
$\dot{r}$	1.408	1.128	1.106	1.234	1.034	1.028	1.031	

Table 8
Statistical comparison of algorithms using *G*-mean as evaluation measure. *s* indicates the frequency of WIN–TIE–LOSE of the method in a row compared to the method in a column, *p* shows the *p*-value of the sign test and $\dot{r}$ demonstrates the geometric mean of the *G*-mean ratio. For C4.5 vs other methods, $*$ means $\dot{r}$ is calculated without considering scores from balance-scale and Mf-Mor (*G*-mean is 0 for C4.5 in these data sets).

	C4.5	RUS	ROS	SMOTE	Chan	Easy	Asym	IRUS
Mean	0.471	0.705	0.624	0.606	0.757	0.758	0.758	**0.768**
RUS								
s	16-5-1		–	–	–	–	–	–
p	0.000							
$\dot{r}$	1.676							
ROS								
s	15-7-0	3-7-12	–	–	–	–	–	–
p	0.000	0.035						
$\dot{r}$	1.357	0.853						
SMOTE								
s	15-7-0	2-6-14	9-7-6	–	–	–	–	–
p	0.000	0.004	0.607					
$\dot{r}$	1.356	0.763	0.894					
Chan								
s	19-3-0	15-7-0	18-3-1	16-5-1	–	–	–	–
p	0.000	0.000	0.000	0.000				
$\dot{r}$	1.774	1.107	1.298	1.451				
Easy								
s	22-0-0	15-6-1	20-1-1	18-3-1	10-6-6	–	–	–
p	0.000	0.000	0.000	0.000	0.455			
$\dot{r}$	1.769	1.108	1.299	1.453	1.001			
Asym								
s	18-4-0	15-7-0	17-4-1	16-4-2	1-20-1	5-8-9	–	–
p	0.000	0.000	0.000	0.001	1.000	0.424		
$\dot{r}$	1.775	1.109	1.300	1.453	1.001	1.000		
IRUS								
s	21-0-1	20-2-0	18-3-1	19-1-2	9-11-2	8-10-4	8-11-3	–
p	0.000	0.000	0.000	0.000	0.065	0.388	0.227	
$\dot{r}$	1.801	1.125	1.319	1.475	1.016	1.015	1.015	

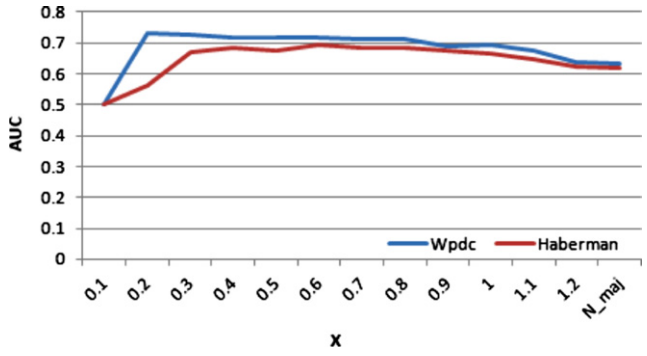

Fig. 4. Parameter *X* vs AUC for Wpdc and Haberman.

represented by a 103-dimensional feature vector. The "emotions" consists of 100 songs from each of the following seven different genres: Classical, Reggae, Rock, Pop, Hip-Hop, Techno and Jazz. The collection was created from 233 musical albums choosing three songs from each album [47]. For emotions, the feature vector consists of 72 dimensions. Table 9 shows certain standard statistics of these data sets.

6.1. Comparison with other multi-label and class imbalance algorithms

The BR-IRUS is compared with the following state-of-the art multi-label methods along with class imbalance methods described in Section 4.2 with micro/macro F_1 used as an evaluation criterion. Micro/macro F_1 are frequently used to assess the average performance of a binary classifier over multiple categories and are also

Table 9
Standard and multi-label statistics for the data sets used in the experiments. LCard (Label Cardinality) is the standard measure of "multi-labeled-ness" [37]. It is the average number of labels relevant to each instance.

Data sets	Domain	Samples	Features	Labels	LCard
Emotions	Music	593	72	6	1.87
Scene	Vision	2407	294	6	1.07
Yeast	Biology	2417	103	14	4.24

frequently used in multi-label classification. The two averaging procedures (micro and macro) bias the results differently. The micro-averaging tends to over-emphasize the performance for the largest categories, while macro-averaging over-emphasizes the performance on the smallest categories. All reported results are estimated from 10×10 fold cross validation and the paired *t*-test is then used to determine their significance under a value of 0.05. In all methods, C4.5 is used as a base classifier.

- BR: Traditional binary relevance approach.
- LP: Traditional label powerset approach.
- RAkEL [48]: An ensemble of label powerset (LP) classifiers where each LP classifier is trained using a different small random subset of the set of labels.
- CLR [40]: An efficient pairwise approach for multi-label classification.
- MAX–MIN [43]: An efficient classification of multi-label and imbalanced data using min–max modular classifiers.

Figs. 5–7 compare the performance of IRUS with the state of the art multi-label classifiers along with class imbalance methods

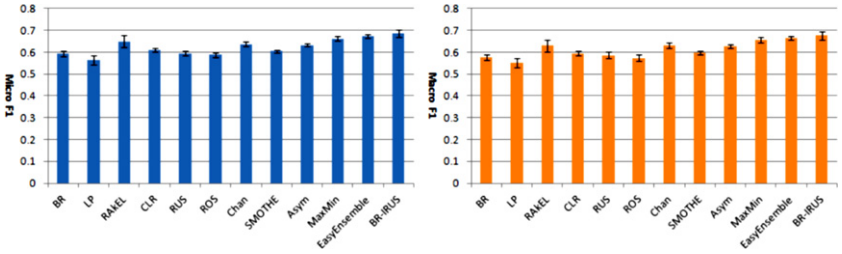

Fig. 5. A comparison of the proposed method with state-of-the-art multi-label classifiers for emotions.

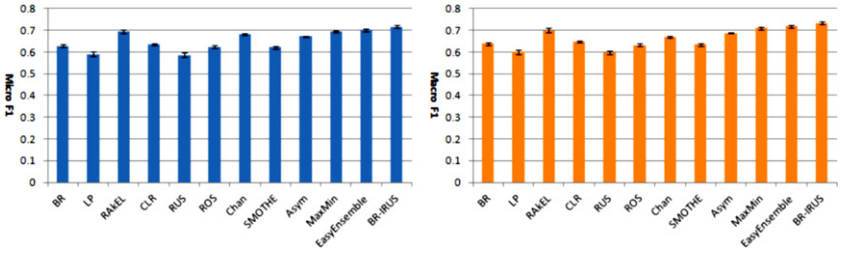

Fig. 6. A comparison of the proposed method with state-of-the-art multi-label classifiers for scene.

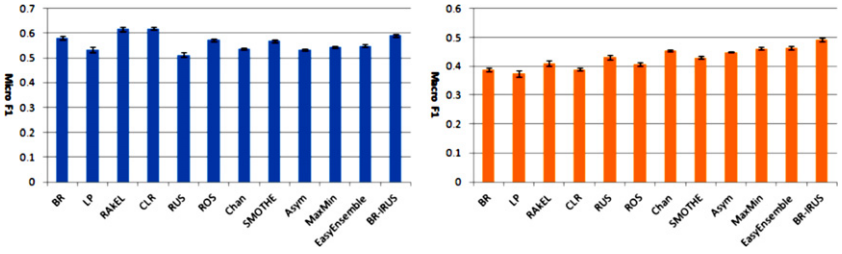

Fig. 7. A comparison of the proposed method with state-of-the-art multi-label classifiers for yeast.

on the emotions, scene and yeast data sets respectively. It is observed that using the proposed technique, significant performance gains have been achieved in all data sets. For emotions, there is an improvement of approx. 5% and 7% for micro/macro F_1 respectively when compared to the nearest best multi-label method RAkEL and an improvement as high as 13% and 15% respectively for micro/macro F_1 when compared with the conventional BR technique. The only exception is the micro F_1 measure in yeast where multi-label methods (CLR and RaKEL) perform significantly better than IRUS. This can be explained by the fact that both CLR and RaKEL consider correlation among labels in multi-label classification while our proposed method uses simple BR technique to learn individual classifiers. Our future work will aim to use the proposed technique as a base classifier in both CLR and RAkEL to improve the performance further as both class imbalance and correlation among labels problems in multi-label methods will then be handled simultaneously. When the proposed approach in compared with other class imbalance techniques, IRUS also performs superior to all methods. The only exception is the comparison of IRUS and EasyEnsemble in emotions where both methods have similar performance as indicated by *t*-test. Overall, class imbalance techniques such as IRUS, EasyEnsemble, MAX–MIN have performed significantly better than multi-label methods (CLR and RaKEL) in scene and emotions and using macro F_1 measure in yeast, which clearly indicates our earlier statement that solving imbalance problem in multi-label learning is very important and even more important than label correlation when using problem transformation methods.

In order to demonstrate the effectiveness of the proposed approach in some individual categories, Figs. 8–10 show the

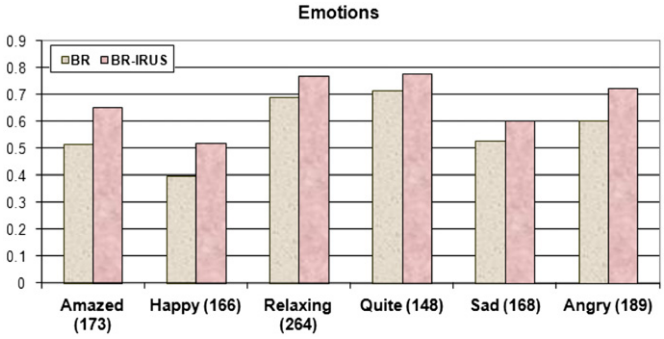

Fig. 8. The performance of the individual concepts measured using F_1 which has been achieved by BR and BR-IRUS for Emotions. The numbers in bracket show the total number of samples belonging to the minority class.

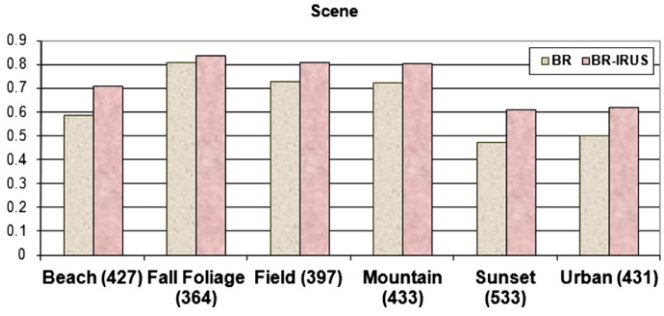

Fig. 9. The performance of the individual categories measured using F_1, which has been achieved by BR and BR-IRUS for scene.

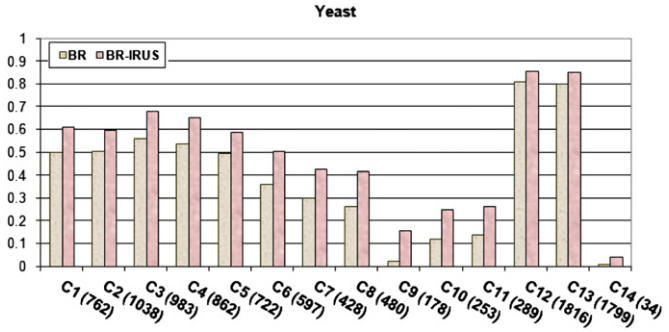

Fig. 10. The performance of the individual categories measured using F_1, which has been achieved by BR and BR-IRUS for yeast.

performance using F_1 on the emotions, scene, and yeast data sets respectively. As expected, using the proposed technique, a significant improvement is obtained in the majority of the categories. Using the conventional BR approach, F_1 achieved is quite low on the highly unbalanced concepts such as Category 9 (C9, $F_1=0.023$) in Yeast. This category or class consists of only 178 samples out of the total of 2417. Using the proposed inverse random sampling technique, there is an improvement of approx. 85% for this highly unbalanced category.

7. 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 inverting 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 have the capacity to realize a complex decision boundary. The distinctiveness of IRUS is assessed experimentally using 22 public UCI data sets. The results indicate significant performance gains when compared with other class imbalance methods. The proposed technique is also used to improve the accuracy of multi-label classification, a challenging research problem in many modern applications such as music, text and image categorization.

In this paper, C4.5 is used as a base classifier. It would be interesting to see how other well-known classifiers like Naive-Bayes, SVM, KNN, LDA behave when used as a base classifier in our proposed inverse under sampling method. Further, IRUS offers a useful methodology for solving practical pattern recognition problems involving disproportionate class sample sizes and puts forward a new approach for researchers to solve imbalance problem by maintaining a very high true positive rate through imbalance inversion and controlling the false positive rate through classifier bagging. Thus, likely to stimulate theoreticians in developing a sound theoretical model to underpin the methodology, with the potential benefits of making it even more effective in the future.

Acknowledgments

The authors would like to thank the four anonymous reviewers for their constructive comments.

References

[1] M. Kubat, S. Matwin, Addressing the course of imbalanced training sets: one-sided selection, in: Proceedings of International Conference of Machine Learning, 1997, pp. 179–186.
[2] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, Synthetic minority over-sampling technique, Journal of Artificial Intelligence Review 16) (2002) 321–357.
[3] M. Japkowicz, S. Stephen, The class imbalance problem: a systematic study, Intelligent Data Analysis (6) (2002) 429–449.
[4] G. Batista, R.C. Prati, M.C. Monard, A study of the behavior of several methods for balancing machine learning training data, SIGKDD Explorations 6 (2004) 20–29.
[5] C.P. de Souto Marcilio, V.G. Bittencourt, 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 (2006) 21–29.
[6] X. Guo, Y. Yin, C. Dong, G. Yang, G. Zhou, On the class imbalance problem, in: Proceedings of the 4th International Conference on Neural Computation, 2008.
[7] C. Diamantini, D. Potena, Bayes vector quantizer for class-imbalance problem, IEEE Transactions on Knowledge and Data Engineering 21 (2009) 638–651.
[8] X.Y. Liu, J. Wu, Z.H. Zhou, Exploratory under-sampling for class-imbalance learning, IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics (2009).
[9] M.A. Tahir, J. Kittler, K. Mikolajczyk, F. Yan, A multiple expert approach to the class imbalance problem using inverse random undersampling, in: International Workshop on Multiple Classifier System, 2009.
[10] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Transactions on Knowledge and Data Engineering 21 (9) (2009) 1263–1284.
[11] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, F. Herrera, A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews, to appear.
[12] I. Tomek, Two modifications of CNN, IEEE Transactions on Systems, Man and Cybernetics 6 (1976) 769–772.
[13] P.E. Hart, Condensed nearest neighbor rule, IEEE Transactions on Information Theory 14 (1968) 515–516.
[14] J. Laurikkala, Improving identification of difficult small classes by balancing class distribution, in: Artificial Intelligence in Medicine, Lecture Notes in Computer Science, Springer, 2001.
[15] P.K. Chan, 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, New York, NY, 1998, pp. 164–168.
[16] Q. Hai-Ni, L. Guo-Zheng, X. Wei-Sheng, An asymmetric classifier based on partial least squares, Pattern Recognition 43 (10) (2010) 3448–3457.
[17] M. Pazzani, C. Merz, P. Murphy, K.A.T. Hume, C. Brunk, Reducing misclassification costs, in: Proceedings of the 11th International Conference on Machine Learning, July, 1994, pp. 217–225.
[18] X. Jing-Hao, T.D. Michael, Do unbalanced data have a negative effect on LDA? Pattern Recognition 41 (5) (2008) 1558–1571.
[19] G.Z. Li, H.-H. Meng, W.-C. Lu, J.Y. Yang, M.Q. Yang, Asymmetric bagging and feature selection for activities prediction of drug molecules, BMC Bioinformatics 9 (Suppl. 6) (2008).
[20] Y. Sun, M.S. Kamel, A.K.C. Wong, Y. Wang, Cost-sensitive boosting for classification of imbalanced data, Pattern Recognition 40 (12) (2007) 3358–3378.
[21] D. Tao, X. Tang, X. Li, X. Wu, Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (7) (2006) 1088–1099.
[22] G.Z. Li, H.H. Meng, W.C. Lu, J.Y. Yang, M.Q. Yang, Asymmetric bagging and feature selection for activities prediction of drug molecules, BMC Bioinformatics 9 (6) (2008).
[23] S. Hido, H. Kashima, Y. Takahashi, Roughly balanced bagging for imbalanced data, Statistical Analysis and Data Mining 2 (5–6) (2009) 412–426.
[24] L. Breiman, Bagging predictors, Machine Learning 24 (1996) 123–140.
[25] S.B. Kotsiantis, P.E. Pintelas, Combining bagging and boosting, International Journal of Computational Intelligence 1 (2004) 324–333.
[26] C.D. Sutton, Classification and regression trees, and boosting, Handbook of Statistics 24 (2005) 303–329.
[27] M. Skurichina, R.P.W. Duin, The role of combining rules in bagging and boosting, in: F.J. Ferri, J.M. Inesta, A. Amin, P. Pudil (Eds.), Advances in Pattern Recognition Proceedings Joint International Workshops SSPR'2000 and SPR'2000, Springer-Verlag, 2000, pp. 631–640.
[28] J. Friedman, P. Hall, On bagging and nonlinear estimation, Journal of Statistical Planning and Inference 137 (2007) 669–683.
[29] L. Breiman, Heuristics of instability and stabilization in model selection, The Annals of Statistics 24 (6) (1996).
[30] J. Kittler, M. Hatef, R.P.W. Duin, J. Matas, On combining classifiers, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (3) (1998) 226–239.
[31] C. Blake, E. Keogh, C. J. Merz, UCI Repository of Machine Learning Databases, 2010.
[32] 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.
[33] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, 2005.

[34] R.J. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993.
[35] S.B. Kotsiantis, P.E. Pintelas, Mixture of expert agents for handling imbalanced data sets, Annals of Mathematics, Computing and Teleinformatics 1 (1) (2003) 46–55.
[36] G.I. Webb, Multiboosting: a technique for combining boosting and wagging, Machine Learning 40 (2) (2000) 159–196.
[37] G. Tsoumakas, I. Katakis, I. Vlahavas, Mining multilabel data, in: Data Mining and Knowledge Discovery Handbook, 2nd ed., Springer, 2009.
[38] M.R. Boutell, J. Luo, X. Shen, C.M. Brown, Learning multi-label scene classification, Pattern Recognition 37 (9) (2004) 1757–1771.
[39] G. Tsoumakas, I. Vlahavas, Random *k*-labelsets: an ensemble method for multilabel classification, in: Proceedings of the ECML, Warsaw, Poland, 2007.
[40] J. Furnkranz, E. Hullermeier, E.L. Menca, K. Brinker, Multilabel classification via calibrated label ranking, Machine Learning 23 (2) (2008) 133–153.
[41] M.L. Zhang, Z.H. Zhou, ML-KNN: a lazy learning approach to multi-label learning, Pattern Recognition 40 (7) (2007) 2038–2048.
[42] W. Cheng, E. Hullermeier, Combining instance-based learning and logistic regression for multilabel classification, Machine Learning 76 (2–3) (2009) 211–225.
[43] K. Chen, B.L. Lu, J. Kwok, Efficient classification of multi-label and imbalanced data using min–max modular classifiers, in: IJCNN, 2006, pp. 1770–1775.
[44] G. Tepvorachai, C. Papachristou, Multi-label imbalanced data enrichment process in neural net classifier training, in: IJCNN, 2008, pp. 1301–1307.
[45] J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier chains for multi-label classification, in: Proceedings of the ECML, 2009.
[46] A. Elisseeff, J. Weston, A kernel method for multi-labeled classification, in: Advances in NIPS, vol. 14, 2002.
[47] K. Trohidis, G. Tsoumakas, G. Kalliris, I. Vlahavas, Multilabel classification of music into emotions, in: 9th International Conference on Music Information Retrieval (ISMIR 2008), Philadelphia, PA, USA, 2008, pp. 325–330.
[48] G. Tsoumakas, I. Katakis, I. Vlahavas, Random *k*-labelsets for multi-label classification, IEEE Transactions on Knowledge and Data Engineering 23 (7) (2011) 1079–1089.

Dr. Muhammad Atif Tahir received his PhD degree from Queen University, Belfast, UK. He was a research fellow at the University of the West of England (2006–2008) and University of Surrey (2008–2010). He is currently working as a senior researcher at University of Northumbria, UK. His current research activities include pattern recognition, machine learning, digital signal processing, image processing, face recognition, and evolutionary heuristics.

Professor Josef Kittler has been a research assistant in the Engineering Department of Cambridge University (1973–1975), SERC Research Fellow at the University of Southampton (1975–1977), Royal Society European Research Fellow, Ecole Nationale Superieure des Telecommunications, Paris (1977–1978), IBM Research Fellow, Balliol College, Oxford (1978–1980), Principal Research Associate, SERC Rutherford Appleton Laboratory (1980–1984) and Principal Scientific Officer, SERC Rutherford Appleton Laboratory (1985). He also worked as the SERC coordinator for Pattern Analysis (1982), and was Rutherford Research Fellow in Oxford University, Department of Engineering Science (1985). He joined the Department of Electrical Engineering of Surrey University in 1986 as a reader in information technology, and became professor of machine intelligence in 1991 and gained the title distinguished professor in 2004.

Fei Yan is a research assistant in the Centre for Vision, Speech and Signal Processing (CVSSP) at the University of Surrey. His research interests include machine learning and computer vision, in particular, kernel methods, distance metric learning, object recognition, and object tracking. He has publications in major machine learning and computer vision conferences and journals, including ICDM, ECML, CVPR, PAMI, JMLR.