Journal of the Operational Research Society (2017)

www.palgrave.com/journals

Benchmarking sampling techniques for imbalance learning in churn prediction

Bing Zhu[1,2]*, Bart Baesens[2,3], Aimée Backiel[2] and Seppe K. L. M. vanden Broucke[2]

[1]*Business School, Sichuan University, No. 24 South Section 1, Yihuan Road, Chengdu 610065, China;*
[2]*Department of Decision Sciences and Information Management, KU Leuven, 3000 Louvain, Belgium; and*
[3]*School of Management, University of Southampton, Highfield, Southampton SO17 1BJ, UK*

Class imbalance presents significant challenges to customer churn prediction. Many data-level sampling solutions have been developed to deal with this issue. In this paper, we comprehensively compare the performance of several state-of-the-art sampling techniques in the context of churn prediction. A recently developed maximum profit criterion is used as one of the main performance measures to offer more insights from the perspective of cost–benefit. The experimental results show that the impact of sampling methods depends on the used evaluation metric and that the impact pattern is interrelated with the classifiers. An in-depth exploration of the reaction patterns is conducted, and suitable sampling strategies are recommended for each situation. Furthermore, we also discuss the setting of the sampling rate in the empirical comparison. Our findings will offer a useful guideline for the use of sampling methods in the context of churn prediction.

Journal of the Operational Research Society (2017). doi:10.1057/s41274-016-0176-1

Keywords: churn prediction; class imbalance; sampling technique; maximum profit measure

1. Introduction

Customer churn prediction is an important issue in customer relationship management. Nowadays, due to fierce market competition and improved access to information, customers can easily switch between competitors. Therefore, more and more companies realize that it is crucial to invest in customer churn prediction to prevent potential customer defection. Marketing research has suggested that attracting new customers is five to six times more expensive than serving current customers (Bhattacharya, 1998; Colgate and Danaher, 2000). Meanwhile, long-term customers tend to be less sensitive to competitive marketing activities and produce higher profits (Ganesh *et al*, 2000; Zeithaml *et al*, 1996). Customer churn prediction and retention program have hence been recognized as marketing priorities, and scholars have witnessed various applications of data mining techniques in this field (Hadden *et al*, 2007; Verbeke *et al*, 2011).

In many industries, such as telecommunication, customer churn is a rare event, i.e., the number of churners is significantly outnumbered by the non-churners. From the perspective of machine learning, the task of customer churn prediction can be presented as a binary classification task with an imbalanced class distribution (Baesens, 2014), where the churners belong to the minority class and non-churners belong to the majority class. The class imbalance problem brings great challenges to standard classification learning algorithms. Most of them tend to misclassify the minority instances more often than the majority instances on imbalanced data sets (Sun *et al*, 2009). In extreme cases, they may classify all instances to the majority class, resulting in high overall precision but unacceptably low accuracy with respect to the interesting minority class. For example, when a model is trained on a data set with 1% of instances from the minority class, a 99% accuracy rate can be achieved simply by classifying all instances as belonging to the majority class. Indeed, the problem of learning on imbalanced data sets is considered to be one of the ten challenging problems in data mining research as listed in Yang and Wu (2006).

In order to solve the problem of learning from imbalanced data sets, many solutions have been proposed in the past few years (Sun *et al*, 2009; He and Garcia, 2009). These solutions can be roughly categorized into two groups: data-level solutions and algorithm-level solutions. The algorithm-level solutions aim to develop new algorithms or modify existing ones to deal with imbalanced data sets, while data-level solutions apply resampling as a preprocessing step to reduce the negative effect caused by class imbalance. The algorithm-level solutions are specific to a certain classification algorithm and are hence often sensitive to the context in which they are applied. Moreover, to develop an algorithm-level solution, one needs to possess extensive knowledge about

*Correspondence: Bing Zhu, Business School, Sichuan University, No. 24 South Section 1, Yihuan Road, Chengdu 610065, China.
E-mail: zhubing1866@hotmail.com

both the learning algorithm and application domain. Data-level approaches, on the other hand, act as a preprocessing phase. Their usage is assumed to be independent of the classifier and can be applied to any learning algorithm. Hence, data-level solutions are more favored in practice, and many such data-level sampling techniques have been proposed. However, although every sampling method claims to improve the performance, conclusions about what is the best sampling solution to address the class imbalance issue never reach consensus. In particular, there is no benchmark comparison of the state-of-the-art sampling methods in the context of churn prediction.

In this paper, we compare the effect of several state-of-the-art sampling methods in the context of churn prediction. The contribution of our work is twofold. First, we include a recently developed profit-based measure called maximum profit measure (MP) as one of the performance measures in the experiment. As pointed out by Raeder *et al* (2012), the choice of evaluation metrics plays an important role in imbalance learning. Traditionally, the area under the receiver operating curve (AUC) and top-decile lift are widely used in churn prediction (Lemmens and Croux, 2006). However, both measures do not take the real cost–benefit brought by the churn model into account. Meanwhile, the MP measure is developed from a cost–benefit perspective by Verbeke *et al* (2012). It combines the output of data mining techniques with the profit of a retention campaign by considering the rewards brought by the retention campaign. The great advantage of the MP measure is that it not only helps firms measure the profit associated with retention campaigns with given prediction models, but also determines the optimal fraction of the customer group to be targeted. It has been shown by Verbeke *et al* (2012) that using the MP measure to select churn prediction models will provide a significant increase in profits compared to AUC and top-decile lift. In our study, we will evaluate the performance of sampling techniques by using the MP measure together with the traditional AUC and top-decile lift measure. Experimental results show that the used evaluation metric has great impact on the performance of sampling techniques and that the impact pattern is also interrelated with the classifier used. We then offer an in-depth exploration of sampling techniques' reaction with regard to different evaluation measures and classifiers and try to find a suitable sampling method in each situation. Second, we discuss the setting of the sampling rate in the experiments, which is an important issue for sampling techniques. It is found that the sampling rate toward a completely balanced class distribution is not necessarily the optimal, while a sampling rate offering a less balanced class distribution would be a better choice.

The remainder of this paper is organized as follows. In the next section, we give a brief literature review of churn prediction and sampling methods for imbalance learning. Then, we will describe the performance measures used in churn prediction. Afterward, two sections present a detailed experimental methodology and discuss the experimental results, respectively. Finally, in the last section, we provide conclusions and directions for further research.

2. Literature review

2.1. Churn prediction

Churn prediction aims at identifying potential churning customers based on past information and prior behavior. Accurate churn prediction models aid firms to develop effective customer retention programs. Neslin *et al* (2006) suggested that prediction techniques have a huge impact on the return of subsequent retention actions and end users should thus continuously strive to improve such techniques. As a result, the optimization of churn prediction models has been explored extensively in the last few years.

Current studies focus on two aspects to optimize churn prediction models. The first aspect focuses on the learning algorithm itself. Numerous algorithms from the research fields of machine learning and statistics have been adopted for churn prediction. For example, Jamal and Bucklin (2006) exploited the hazard model to understand the churn behavior of television subscribers. Coussement and Van den Poel (2008) applied support vector machines to construct churn models in a newspaper subscription context. Keramati *et al* (2014) employed decision trees, artificial neural networks and *k*-nearest neighbors to improve churn prediction. All the works listed above implement the prediction model as a single model. In recent years, researchers have also explored the use of so-called ensemble or hybrid learning techniques toward customer churn prediction. For example, boosting and bagging models have been used Lemmens and Croux (2006). De Bock and Van den Poel (2011) applied rotation-based ensemble classifiers to customer churn prediction. Farquad *et al* (2014) tested a hybrid data mining technique which combines support vector machines with Naive Bayes trees on analyzing bank credit card customer churn in analytical CRM. Although various algorithms have been applied, there appears no single dominating method or approach which is useful in every churn-related context. For an extensive literature review on customer churn prediction techniques, we refer the interested reader to Hadden *et al* (2007) and Verbeke *et al* (2011).

Another research stream puts the emphasis on the data used to build the churn model. Coussement and Van den Poel (2008), for instance, found evidence showing that adding sentiment- and emotion-based features such as those expressed in customer emails increases the predictive power of churn models. Lima *et al* (2009) attempted to show how domain knowledge can be incorporated in a churn prediction model by evaluating coefficient signs in a logistic regression model or by analyzing a decision table extracted from a decision tree or rule-based classifier. Chen *et al* (2012) integrated static

customer data and longitudinal behavioral data to improve the performance of churn prediction models. The research in our paper also falls into this second type.

2.2. Class imbalance

Most standard classification algorithms attempt to reduce global measures such as error rate. They thus frequently exhibit a bias toward the majority class and suffer from class imbalance. Weiss (2004) drew up six types of problems brought on by imbalanced data sets. More details about how class imbalance can influence different types of classification algorithms can be found in Sun *et al* (2009).

The common solutions for class imbalance can be roughly divided into two categories: data-level solutions and algorithm-level solutions. The data-level solutions try to re-balance the class distribution by resampling the original, unbalanced data set. Algorithm-level approaches attempt to adapt existing learning algorithms to strengthen their learning ability with regard to the minority class. Algorithm-level solutions require deep knowledge and understanding of both the corresponding classifier itself and the application domain in which it will be applied. Therefore, data-level solutions are more favored in practice because they are assumed to be independent of the learning algorithm.

Data-level solutions consist of many different forms of resampling techniques. Random sampling techniques offer the simplest approaches, which includes random oversampling (ROS) and random undersampling (RUS). ROS tries to create a superset of the original data set by randomly replicating the minority instances from the existing data set. RUS aims to balance the class distribution through random elimination of majority class examples. Both techniques have their pros and cons: Both are easy to use and understand, but RUS can discard potentially useful data instances, whereas ROS makes exact copies of existing instances so it can increase the likelihood of overfitting. In order to deal with these drawbacks, some intelligent sampling methods have been proposed, of which synthetic minority oversampling technique (SMOTE) (Chawla *et al*, 2002) is probably one of the most popular of such intelligent sampling methods. It first randomly selects one or more nearest neighbors of a minority class instance and produces new instances based on the linear interpolations between the original examples and randomly selected nearest neighbors. Although SMOTE is well acknowledged in the academic community, it still suffers from some drawbacks. For instance, SMOTE generates the same number of synthetic minority instances for each original minority example and does not take the neighboring examples belonging to the majority class into consideration, which can result in an increase of overlap between the classes. To overcome the shortcomings of SMOTE, some variants have been developed. Adaptive synthetic sampling (ADASYN) uses the density distribution to decide one the number of synthetic samples that need to be generated for each minority example (He *et al*, 2008). It first finds nearest neighbors for each minority instance and counts the number of majority examples in these neighbor instances. The number of instances generated for each instance is then calculated by this ratio. The Borderline-SMOTE (Han *et al*, 2005) technique is another modified version of SMOTE. By calculating the number of majority instances in the nearest neighbors for each instance, this algorithm divides the minority class instances into three groups: safe, dangerous and noisy. Borderline-SMOTE only generates artificial instances for those dangerous minority examples that are assumed to be borderline data. Recently, a new intelligent sampling method, called majority weighted minority oversampling technique (MWMOTE), was also presented (Barua *et al*, 2014). MWMOTE first identifies the hard-to-learn informative minority class samples and assigns them weights according to the distance from the nearest majority class samples. It then generates the synthetic samples from the weighted informative minority samples using a clustering approach.

Some intelligent methods for undersampling have also been proposed. Kubat and Matwin (1997) proposed the one-sided selection (OSS) technique, which selectively removes the majority instances that either are redundant or borderline majority examples. The cluster-based undersampling algorithm (CLUS) presented by Yen and Lee (2009) organizes the training data into groups with homogeneous characteristics and then downsizes the number of majority class samples in each cluster.

There are also several hybrid sampling techniques. Some techniques combine oversampling with data cleaning techniques to reduce the overlapping introduced by oversampling methods. For instance, SMOTE–Tomek finds pairs of minimally distanced nearest neighbors of opposite classes. When two samples form a Tomek link, either one of them is noisy or both samples are borderline data. SMOTE–Tomek deletes Tomek links after SMOTE sampling so that all minimally distanced nearest neighbor pairs are in the same class. SMOTE–ENN is similar to SMOTE–Tomek, but it uses Wilson's edited nearest neighbor (ENN) rule to remove instances after SMOTE sampling (Batista *et al*, 2004). That is, any example that is misclassified by its three nearest neighbors is removed from the data set. Apart from the usage of data cleaning techniques, some researchers presented hybrid sampling methods that apply computational intelligence technologies such as genetic algorithms and particle swarm optimization to identify more useful instances (García and Herrera, 2009; Yang *et al*, 2009).

In order to find suitable sampling methods from all the candidates, there are several experimental comparison works. Chawla (2003) studied three sampling methods, i.e., SMOTE, ROS and RUS. He concluded that SMOTE improves the AUC value of C4.5 decision tree models over ROS and RUS. Van Hulse *et al* (2007) presented a comprehensive experimentation

with eight sampling methods. It was found that random sampling methods perform better than intelligent sampling methods like SMOTE and Borderline-SMOTE. García *et al* (2012) investigated the influence of both imbalance ratio and classifier on several resampling strategies to deal with imbalanced data sets. Experiments showed that oversampling consistently outperforms undersampling when data sets are strongly imbalanced, whereas there are no significant differences on data sets with a low class imbalance. Marqués *et al* (2013) investigated the suitability and performance of several resampling techniques over five real-world credit data sets. Experimental results demonstrated that the use of resampling methods consistently improves the performance and oversampling techniques perform better than any undersampling approach in general. López *et al* (2013) analyzed the performance of five representative sampling approaches on imbalanced data sets. SMOTE and SMOTE–ENN are recognized here as the top methodologies, with AUC being used as the evaluation metric. Seiffert *et al* (2014) present a systematic set of experiments designed to identify which learners and data sampling techniques are most robust when confronted with noisy and imbalanced software quality data. They found that RUS is the best sampling techniques in terms of AUC. To summarize the above comparison studies, we observe that the conclusions about the performance of sampling techniques do not reach a clear agreement and that most studies consider AUC as the evaluation measure. Since model performance on imbalanced data sets is influenced by many factors, we argue that it is not feasible to determine which sampling method is best suited in any general context. Therefore, we narrow our contribution and scope to the churn prediction domain. Besides the AUC measure, we will present an in-depth exploration of sampling methods' performance with the domain-specific MP measure.

The configuration of the sampling rate is an important issue for sampling techniques, for which there is no universally accepted setting. Weiss and Provost (2003) demonstrated that the optimal class distribution is near the natural class distribution (unbalanced) when accuracy is the evaluation metric, while the best class distribution tends to be near the balanced class distribution when AUC is used. Khoshgoftaar *et al* (2007) indicated that less balanced class ratios 1:3 or 1:2 (minority vs. majority) may lead to better classification performance when the number of minority instances is limited. More studies are needed to investigate this important topic.

In churn prediction, the techniques to deal with class imbalance do not go beyond the scope of data-level and algorithm-level methods. Several algorithm-level solutions have been proposed in this domain. For example, Xie *et al* (2009) applied the improved balanced random forests approach to churn prediction. Xiao *et al* (2012) proposed a method that handles class imbalance by combining ensemble learning with cost-sensitive learning. However, sampling is still the prevalent choice in most current works such as in Chen *et al* (2012) or in Ali and Ariturk (2014), where random undersampling and SMOTE sampling are used. Burez and Van den Poel (2009) have performed some pioneering studies to compare sampling methods with cost-sensitive learning in churn prediction, but only two sampling methods were considered.

3. Evaluation metrics in churn prediction

The selection of an evaluation metric, that is, the metric which will be applied to evaluate a model's performance after training, plays an important role in the overall construction of a learning system. In this section, we will provide a brief description of the three metrics used in our experiments.

3.1. ROC/AUC and top-decile lift

The receiver operating characteristic (ROC) curve is one of the popular approaches that can be used to measure the performance of classifiers on imbalanced data sets. ROC analysis has its origin in signal detection theory as a method to choose a threshold or an operating point for the receiver to detect the presence or absence of a signal. Bradley (1997) introduced ROC analysis to the machine learning field in 1997. The ROC graph plots true-positive rates versus false-positive rates. Classifiers can be selected based on their trade-offs between true positives and false positives. Rather than visually comparing curves, the area under the ROC curve metric (AUC) aggregates the performance of classification methods into a single number, which makes it easier to compare the overall performance of multiple classification models. A random classifier has an AUC of 0.5, and an perfect classifier possesses an AUC equal to 1.

Top-decile lift is another widely used performance measure in churn prediction. To calculate top-decile lift, customer instances are first sorted based on the churn propensity score obtained by the prediction model in a descending order. Then, the top-decile lift is computed as the ratio of the percentage of correctly classified churners (the minority class) in the top 10% ranked cases and the percentage of actual churners in the entire data set. For instance, a top-decile lift value of 2 indicates that the classifier identifies twice as many churners in the top 10% ranked customer group as a random classifier would do.

Although AUC and top-decile lift are two widely used performance measures in churn prediction, both measures have their own drawbacks. Top-decile lift only considers a fixed fraction of customers as the target group, a fraction that may not be the optimal one that should be included in the retention campaign. AUC averages the misclassification loss over different cost ratio distributions, but it uses different misclassification cost distributions for different classifiers, which means the AUC evaluates a classifier using a metric which

depends on the classifier itself. Hence, the AUC metric is an incoherent measure of classifier performance (Hand, 2009). Besides, the most important point is that both measures do not take the real cost–benefit brought by the churn model into account. Consequently, they lead to suboptimal model selection from the perspective of profit as shown in Verbeke *et al* (2012).

3.2. Maximum profit criterion

Figure 1 presents the dynamical process of customer churn and retention. In the retention process, a firm will first apply the churn prediction model to the current customer base. Then, a fraction α of the customers with the highest churn propensities are targeted and offered some incentive. There are true and false would-be churners within the target customers. In the true would-be churner group, a fraction γ of customers will accept the offer and stay active, whereas the remaining fraction $1-\gamma$ will still defect. In the false would-be churner group, all the customers will accept the incentives and they do not churn because actually they have no intension to leave. Meanwhile, in the $1-\alpha$ fraction (the non-target group), the would-be churners will leave, whereas non-would-be churners will stay as well. In this process, we can see that the profit of launching a retention program is influenced by the customers' churn propensity in the target fraction, their probability of accepting an incentive offer, the cost of the incentive offer, customer lifetime value and so on. Neslin *et al* (2006) established the following expression to get the total profit for a retention campaign:

$$P = N\alpha[\beta\gamma(\mathrm{CLV}-c-\delta)+\beta(1-\gamma)(-c) + (1-\beta)(-c-\delta)]-A \quad (1)$$

where P is the profit of the retention campaign, N is the number of customers in the current customer base, α is the fraction of the target customers in the retention campaign, β is the portion of churners within the targeted customers, γ is the probability of would-be churners accepting the offer and staying with the company, CLV is the average customer lifetime value of the retained customers, c is the cost of the incentive when a customer accepts the offer, δ is the cost of contacting a customer to offer the incentive, and A is the fixed administrative cost of running the retention program. The first term in the bracket of Eq. (1) reflects the revenue contributed by the retained potential churners. The second term presents the loss coming from the identified potential customers who do not accept the offer and finally defect. The last term in the bracket is the loss of sending incentives to the customers who are actually non-churners.

The target fraction α in Eq. (1) is a fixed value, and it does not maximize the profit in retention campaigns given a churn prediction model. Recently, Verbeke *et al* (2012) presented a maximum profit (MP) criterion to address this issue. The definition of the MP criterion is as follows:

$$\mathrm{MP} = \max_{\alpha}(N\alpha[(\gamma\mathrm{CLV}+\delta(1-\gamma))\beta_0\lambda-\delta-c]) \quad (2)$$

where β_0 is the churn rate in the whole customer base and λ is the lift corresponding to the target fraction α.

By selecting the optimal target size $N\alpha$, the MP measure will give the maximum profit that the retention campaign can achieve given the churn prediction model. As mentioned in Verbeke *et al* (2012), the MP criterion will result in higher profits than the commonly used top-decile lift and AUC measures. We will use Eq. (2) to compute the potential maximum profit brought by each model in our experiments.

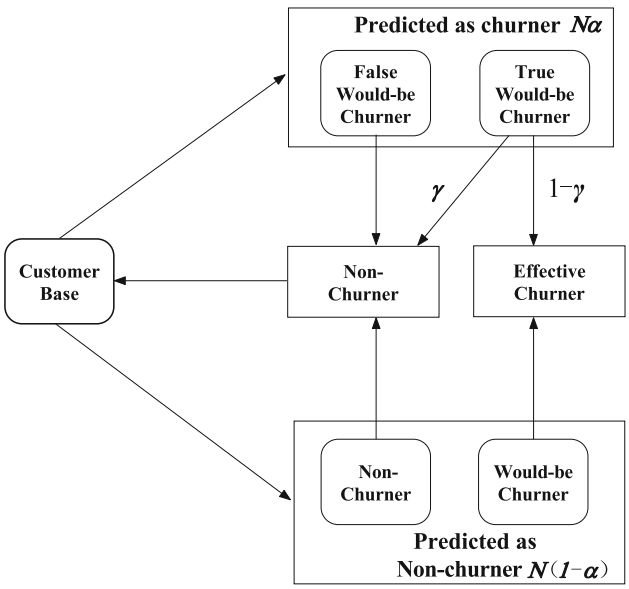

Figure 1 The dynamic process of customer churn and retention.

4. Experimental framework

In this section, we present the framework used to carry out the experiments. We provide details of the real-world customer data sets used in the experiments in the first subsection. Then, the next subsection briefly describes the sampling algorithms and classification methods included in the study as well as other experimental settings. Finally, we present the statistical tests that we have applied to the experimental results in the last subsection.

4.1. Data sets

The eleven real-world data sets used in the experiments come from the telecommunication industry. Table 1 summarizes the main characteristics of the data sets, where each column presents the name, abbreviation (Abbr.), data source, region, number of observation (#Obs.) and attributes (#Att.) as well as the churn rate. Among all the eleven data sets, seven are commercial data sets: Chile, K1, K2, K3, K4, K5, Tele1, which come from telecom operators of South America, East Asia and Europe. Duke1 and Duke2 were provided by the CRM Center of Duke University. UCI and KDDcup data sets are public available data sets. The UCI data set stems from the well-known UCI machine learning repository, and the KDDcup data set originates from the 2009 KDD cup competition. As Table 1 shows, the churn rates range from 1.8 to 14.5%, and most of them are highly imbalanced.

Two steps are performed for data preparation before proceeding with the model construction. The first step involves dealing with missing values. In case of categorical variables, the missing values are assigned as special values. For the continuous variables, if more than 50% of the values are missing, then the variables are removed. Otherwise, the median imputation is used to avoid the influence of extreme values. In order to avoid the "curse of dimensionality" problem, a feature selection procedure was conducted as the second step for data preparation. The feature selection approach based on the Fisher score is used on five data sets: Chile, Duke1, Duke2, KDD and Tele1. We use the following formula to calculate the Fisher score for each variable:

$$\text{Fisher Score} = \frac{|\bar{x}_c - \bar{x}_{nc}|}{\sqrt{s_c^2 + s_{nc}^2}} \tag{3}$$

where $\bar{x}_c$ and $\bar{x}_{nc}$ are the mean value, and s_c^2 and s_{Nc}^2 are the variance of a variable for churners and non-churners, respectively. Features with highest Fisher scores on the four above-mentioned data sets are retained, and the number of variables in these data sets is reduced to 30.

4.2. Experimental setting

Nine sampling methods representing the state-of-the-art are considered and have been listed in Table 2 together with their parameter settings. The basic ideas of all the included methods have been described in the first subsection. Apart from the sampling methods, we also include a non-sampling strategy in the experiments to analyze whether the use of sampling is beneficial, which is denoted as "None." Most parameter values of sampling methods were selected according to the recommendation of the corresponding authors of each algorithm, as referenced in Table 2. For CLUS, we have experimented with several values for the number of clusters $K \in \{3, 5, 7\}$, with similar results, so that the one requiring the minimum computational cost $K = 3$ was selected. To find suitable configurations of sampling rates, we consider three different settings for the class ratios: 1:3, 2:3 and 1:1 (minority vs. majority). They are hereafter referred to as "less balanced (LB)," "roughly balanced (RB)" and "perfectly balanced (PB)" strategies in this paper.

A 5×2 cross-validation strategy is applied in our study as follows: Each original data set is randomly split into two equally sized parts. First, one part is used as training data to build the model, while the other part acts as test data to calculate the performance. Then the two parts switch their roles. This process is repeated five times to get the average value of each performance measure. Four benchmark classifiers are used in the experiments to build the model: logistic regression, C4.5 decision tree, support vector machine (SVM) and random forests (RF), which are widely used in churn prediction. For the SVM method, we choose the radial basis

Table 1 Summary of the churn data sets used in the experiment

Data set	*Abbr.*	*Source*	*Region*	*#Obs.*	*#Att.*	*Churn rate (%)*
Chile	Chile	Operator	South American	5300	41	5.66
Duke_current	Duke1	Duke	North American	51306	173	1.80
Duke_future	Duke2	Duke	North American	100462	173	1.80
KDDcup	KDDcup	KDD CUP 2009	Europe	50000	231	7.34
Korean1	K1	Operator	East Asia	2019	10	3.96
Korean2	K2	Operator	East Asia	2941	14	4.42
Korean3	K3	Operator	East Asia	5990	36	4.34
Korean4	K4	Operator	East Asia	2183	9	4.58
Korean5	K5	Operator	East Asia	26224	11	4.19
Tele1	Tele1	Operator	Europe	4350	87	8.05
UCI	UCI	UCI ML repository	–	3333	19	14.5

Table 2 Sampling methods included in the experiment with their parameter setting

Sampling methods	*Parameters*
ADASYN	$k = 5$ (He *et al*, 2008)
Borderline-SMOTE	$m = k = 5$ (Han *et al*, 2005)
CLUS	Number of clusters $K = 3$
MWMOTE	$k1 = 5, k2 = 3, k3 = \lvert S_{\min}\rvert/2, C_p = 3, C_f(th) = 5, CMAX = 2$ (Barua *et al*, 2014)
RUS	No parameter
ROS	No parameter
SMOTE	$k = 5$ (Chawla *et al*, 2002)
SMOTE–ENN	$k_{SMOTE} = 5, k_{ENN} = 3$ (Batista *et al*, 2004)
SMOTE–Tomek	$k_{SMOTE} = 5$ (Batista *et al*, 2004)

function (RBF) as kernel function. We select AUC, top-decile lift and MP to measure the model performance. To calculate the maximum profit, values of the parameters γ, CLV, δ and c in Eq. (2) are set to be 0.30, 200, 10 and 1, respectively. The setting of these values is based on previous scientific literature (Neslin *et al*, 2006; Jahromi *et al*, 2014) and discussion with data scientist of the telecommunication industry. In total, eleven data sets, four classifiers, ten sampling methods (including the non-sampling strategy) and three evaluation metrics are considered in the experiment. All the experiments were conducted using the open-source R language.

4.3. Statistical analysis

Following the recommendation of Demšar (2006), the non-parametric Friedman test and the Holm's post hoc procedure were used in our empirical study to evaluate the statistical differences. In each scenario, we first use the Friedman test to detect statistical differences among a group of results with the null hypothesis that different factor levels have similar ranks. As Demšar notes, the Friedman test is safer than a parametric tests since it does not assume normal distributions or homogeneity of variance. The Friedman test is defined as follows:

$$\chi_F^2 = \frac{12 * N}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right] \tag{4}$$

where N is the number of data sets, k is the number of factor levels, R_j is the average rank of factor level $j = 1, \ldots, k$ over N data sets. Under the null hypothesis the Friedman statistic is distributed according to χ_F^2 with $k - 1$ degrees of freedom. If this null hypothesis is rejected, it implies the existence of differences among different factor levels. In that case, a further post hoc test should be applied to point out where differences lay. In our study, we apply Holm's post hoc test (Holm, 1970) to do so, as it appropriately controls the family-wise error rate (FWER) and hence it is a more powerful test than Bonferroni–Dunn's test in a rigorous comparison (Demšar, 2006). The Holm test is a step-down procedure, and it obtains the normal distribution statistic as follows:

$$Z = \frac{(R_i - R_j)}{\sqrt{k(k+1)/(6 * N)}} \tag{5}$$

where R_i and R_j are the average ranks of factor level i and j over all the N data sets. The Z statistic is used to find the corresponding p-value from the normal distribution table. Starting from the most significant p-value, Holm's procedure compares each p_i with $\alpha/(k-i)(i = 1, 2, \ldots, k-1)$ where significance level α is the given significance level. If p_i is smaller than $\alpha/(k-i)$, the corresponding hypothesis is rejected and it continues with the next comparison $(i = i + 1)$. The process will continue until a certain hypothesis cannot be rejected and all the remaining null hypotheses are accepted.

5. Experimental results

In this section, we present and analyze the experimental results. Our aim is to answer the following two questions.

(1) Which is the appropriate configuration of sampling rates?
(2) How do different sampling techniques perform in churn prediction? How do evaluation metrics influence their effectiveness?

To answer the above questions, we divide our analysis into two stages. In the first stage, we will analyze which configuration of sampling rate is the best. In the second stage, we will develop a comparison of different sampling methods and investigate the influence of evaluation metrics.

5.1. Comparison of sampling rates

We start our investigation by comparing the different sampling rates. Since we cannot control the exact class ratio when using SMOTE–ENN and SMOTE–Tomek, only seven sampling techniques are considered when comparing the sampling rate. We calculate the mean performance of each sampling rate across different classifiers and sampling methods. Table 3 shows the results, where the last row gives the average ranks of

Table 3 Average performance with different sampling rates

Data set	*AUC*			*MP*			*Top-decile lift*		
	LB (1:3)	*RB (2:3)*	*PB (1:1)*	*LB (1:3)*	*RB (2:3)*	*PB (1:1)*	*LB (1:3)*	*RB (2:3)*	*PB (1:1)*
Chile	**0.6881**	0.6864	0.6825	**0.5003**	0.4614	0.4259	**3.4481**	3.3956	3.3320
Duke1	0.5566	**0.5597**	0.5587	0.0000	0.0000	0.0000	**1.3805**	1.3740	1.3345
Duke2	0.5643	**0.5701**	0.5680	**4.64** $\times 10^{-5}$	0.0000	0.0000	**1.4952**	1.4949	1.4545
KDD	0.6643	**0.6706**	0.6670	0.1577	**0.1649**	0.1155	**2.4183**	2.3255	2.1862
K1	**0.7255**	0.7231	0.7226	**0.0217**	0.0109	0.0082	**3.1906**	3.1769	3.1794
K2	0.8067	**0.8139**	0.8112	**0.1258**	0.1103	0.1143	4.1343	**4.1375**	4.1049
K3	0.8895	**0.8913**	0.8877	**0.8713**	0.8297	0.8244	**7.3926**	7.2504	7.0697
K4	**0.8468**	0.8454	0.8406	**1.0235**	0.9972	0.9141	5.7003	**5.7057**	5.5692
K5	**0.7783**	0.7771	0.7731	**0.3326**	0.2876	0.2490	4.2927	**4.3221**	4.2343
Tele1	**0.7657**	0.7595	0.7579	**0.4705**	0.3119	0.2161	**3.6743**	3.4218	3.2308
UCI	**0.8606**	0.8563	0.8533	**4.7636**	4.6583	4.3635	**4.3524**	4.0713	3.9229
AR	1.8182	**1.5454**	2.6364	**1.1818**	2.0455	2.7727	**1.2727**	1.8181	2.9091

the three sampling rates in terms of each measure on every data set. The best values are shown in bold.

As Table 3 shows, when AUC is taken into consideration, the roughly balanced (RB) sampling strategy (class ratio 2:3) has the best performance. It has the highest average rank and takes the first place on five out of eleven data sets. The less balanced (RB) sampling strategy (ratio 1:3) comes second and performs the best on six data sets. Sampling rates corresponding to the perfect balanced class distribution (PB) perform the worst. It never appears in the top for all the 11 data sets. As to the MP measure, the performance of the LB strategy dominates across ten data sets and has the smallest ranking values. The average rank of the RB strategy takes the middle position and ranks the first only in one data set (KDD). The PB strategy again has the worst ranking. Top-decile lift results in similar rankings as with the MP measure, but the dominance of LB strategy is less pronounced: It only ranks first on eight data sets, and the difference of average ranks between the RB and LB strategies is smaller.

To determine the statistical difference, we used the Friedman test and obtained the p values of 0.0289, 0.0004 and 0.0005 for AUC, MP and top-decile lift, respectively. This indicates that there is a significant difference between the different sampling rates for all the three measures at a 5% significance level. As a consequence, we used the Holm's post hoc procedure to further explore the statistical difference. Tables 4, 5 and 6 illustrate the results of the Holm's test, where each column shows the Z statistic, p value, threshold for comparison and the decision. As it can be seen from these tables, when MP is used, the LB strategy significantly outperforms the other two strategies. When AUC and top-decile lift are considered, the LB sampling also significantly outperforms the PB strategy, while it is statistically equivalent to the RB strategy. According to these results, if only one strategy can be selected, the less balanced strategy with a class ratio of 1:3 would be our recommendation due to its relatively good performance.

5.2. Comparison of sampling techniques

We now proceed with the analysis by comparing different sampling methods. Since we have observed the good overall performance of the less balanced strategy (LB), comparison of different sampling methods is based on that configuration. For SMOTE–ENN and SMOTE–Tomek, we present results under the same parameter setting that helps SMOTE reach a less balanced class ratio. Tables 7, 8, 9 and 10 summarize the experimental results, grouped by classifiers. The last columns give the average rank (AR) of sampling methods for each combination of evaluation metric and classifier. The best values in each situation are depicted in bold. By using Holm's test, the average ranks that are significantly different from the top performance at the 10% significance level are also underlined. Figures 2, 3 and 4 illustrate the rankings of sampling methods. In each graph, the horizontal axis presents the average rank of different sampling methods over the eleven data sets for a given classifier. Each dot represents one sampling method. The sampling methods are ranked in descending order according to their average ranks, where the bottom method is the control method with the best average rank. Each horizontal line starts from average rank of the control method on the left end, and the length gives the critical difference for a particular comparison. So if a dot goes beyond the right end of the horizontal line, it means the corresponding method has significant different average rank with the control method at the 10% significance level. The results show that sampling methods indeed exhibit different behaviors with different evaluation measures, which is interrelated with classifier. In order to make analysis more clear, we will divide the results into three groups. The results obtained with logistic regression and SVM make up the first and third group. The results brought by C4.5 decision trees and random forests form the second group. We will analyze the results of the three groups separately and investigate the impact of evaluation metrics in each group.

Table 4 Holm test based on AUC measure for different sampling rates

i	*Algorithm*	*Z-value*	*p-value*	*Holm*	*Hypothesis*
Control methods: RB strategy (2:3)					
2	PB strategy (1:1)	2.5584	0.0105	0.025	Rejected
1	LB strategy (1:3)	0.6396	0.5524	0.05	Not rejected

Table 5 Holm test based on MP measure for different sampling rates

i	*Algorithm*	*Z-value*	*p-value*	*Holm*	*Hypothesis*
Control methods: LB strategy (1:3)					
2	PB strategy (1:1)	3.7310	0.0002	0.025	Rejected
1	RB strategy (2:3)	2.0254	0.0428	0.05	Rejected

Table 6 Holm test based on top-decile lift for different sampling rates

i	*Algorithm*	*Z-value*	*p-value*	*Holm*	*Hypothesis*
Control methods: LB strategy (1:3)					
2	PB strategy (1:1)	3.8376	0.0001	0.025	Rejected
1	RB strategy (2:3)	1.2792	0.2008	0.05	Not rejected

5.2.1. Logistic regression As we can see from Table 7 and Figure 2, the non-sampling strategy achieves the best results with both AUC and top-decile lift. It takes the first place on five out of eleven data sets in terms of AUC. Although the non-sampling strategy never ranks first on any data set as to top-decile lift, it still has the best average rank. When MP is considered, the non-sampling strategy takes the fourth place and it is statistically equivalent to SMOTE–ENN which has the smallest average rank. To sum up, most current sampling methods do not improve the performance of logistic regression on imbalanced data sets. SMOTE–ENN may have a small improvement when the MP measure is considered.

5.2.2. C4.5 decision tree and random forests With respect to C4.5 and random forests, the effectiveness of sampling methods depends on the measures as shown in Figures 3 and 4. In terms of AUC, most sampling methods have a positive impact on the results. RUS has the best results for C4.5, while ROS is the best option for random forests. The non-sampling strategy is also significantly worse than the top ranked methods at the 10% significance level for both classifiers. For both top-decile lift and the MP measure, on the contrary, sampling methods have a mixed influence. The non-sampling strategy ends the first for C4.5 and ranks the fourth for random forests in terms of top-decile lift measure. With respect to the MP measure, the non-sampling strategy ranks the third for C4.5 and the seventh for random forests, which is a little higher compared with top-decile lift measure. The results show that some sampling techniques have a negative impact, while others have positive impact. For both top-decile lift and MP measure, there is no statistical difference between non-sampling and top ranked sampling methods according to Holm's test.

5.2.3. SVM In contrast to logistic regression, C4.5 and random forests, most sampling techniques work well with SVMs across different measures. As Figure 5 shows, the non-sampling strategy stays at the worst place of the ranking list for all the three measures. Non-sampling is also statistically worse than the best ranking methods for both AUC and top-decile lift measures. The results indicate that sampling methods have a positive impact on the performance. We further investigate the results of each individual measure. It is interesting to note that ROS is the best performing sampling technique when using AUC. For top-decile lift, SMOTE–Tomek and ROS are the top two methods. Finally, RUS method shows their superiority when using the MP measure, which takes the best place. However, note that the difference between RUS and the non-sampling strategy is not significant in this situation.

5.3. Results summary and discussion

We draw the following conclusions from the experimental results.

Firstly—the sampling rate toward the less balanced class ratio 1:3 seems to be a good and general option for sampling

Table 7 Experimental results for logistic regression

Measure	*Method*	*Chile*	*Duke1*	*Duke2*	*KDD*	*K1*	*K2*	*K3*	*K4*	*K5*	*Tele1*	*UCI*	*AR*
AUC	None	**0.7326**	**0.6004**	0.6024	**0.7152**	0.7488	0.8473	0.9305	**0.7642**	**0.7978**	0.8014	0.8080	**3.3636**
	ADASYN	0.7273	0.5920	**0.6035**	0.7131	0.7446	0.8534	0.9284	0.7476	0.7963	0.8035	0.8066	5.1818
	Borderline-SMOTE	0.7143	0.5686	0.5682	0.7104	0.6976	0.8311	0.9253	0.7311	0.7946	**0.8082**	0.8092	8.0000
	CLUS	0.7069	0.5869	0.5523	0.6962	0.7303	0.8403	0.9161	0.7405	0.7777	0.7905	0.7963	9.3636
	MWMOTE	0.7128	0.5937	0.5701	0.7133	0.7498	0.8571	0.9287	0.7555	0.7955	0.8091	0.8087	5.0000
	ROS	0.7249	0.6003	0.5987	0.7151	0.7418	**0.8627**	**0.9318**	0.7341	0.7958	0.7999	0.8088	4.6364
	RUS	0.7148	0.5940	0.5980	0.7144	0.7169	0.8506	0.9149	0.7511	0.7927	0.7939	0.8111	6.6818
	SMOTE	0.7254	0.6001	0.6001	0.7136	0.7509	0.8541	0.9261	0.7620	0.7964	0.8046	0.8133	3.4545
	SMOTE–ENN	0.7225	0.5981	0.6021	0.7128	**0.7611**	0.8561	0.9315	0.7492	0.7954	0.8016	**0.8137**	4.2727
	SMOTE–Tomek	0.7244	0.5934	0.6029	0.7144	0.7437	0.8458	0.9227	0.7475	0.7965	0.8038	0.8120	5.0455
MP	None	0.6571	0.0000	0.0000	0.3126	**0.0291**	0.1982	1.3353	**0.0543**	0.0000	0.0309	3.6870	4.9091
	ADASYN	0.6506	0.0000	0.0000	0.3059	0.0034	**0.2616**	1.3430	0.0000	0.3273	0.0125	3.6941	5.2273
	Borderline-SMOTE	0.4065	0.0000	0.0000	0.0000	0.0003	0.0081	0.0439	0.0000	0.0087	**0.1737**	0.9341	7.6818
	CLUS	0.5912	0.0000	0.0000	0.2840	0.0056	0.1712	1.0679	0.0000	0.1601	0.0021	3.4724	7.5000
	MWMOTE	0.6443	0.0000	0.0000	0.3143	0.0130	0.2569	1.3691	0.0191	0.2990	0.0525	3.6801	4.2727
	ROS	0.6364	0.0000	0.0000	0.3178	0.0000	0.1537	**1.3991**	0.0000	0.2916	0.0490	3.7132	5.2727
	RUS	0.6568	0.0000	0.0000	**0.3200**	0.0000	0.1412	1.2211	0.0000	0.2569	0.0202	3.7072	6.0909
	SMOTE	0.6474	0.0000	0.0000	0.3176	0.0000	0.2129	1.3904	0.0072	0.3440	0.0267	3.7526	4.5000
	SMOTE–ENN	0.6532	0.0000	0.0000	0.3011	0.0250	0.2313	1.3655	0.0148	0.3211	0.0446	**3.8457**	**4.0000**
	SMOTE–Tomek	**0.6782**	0.0000	0.0000	0.3041	0.0000	0.1588	1.2974	0.0000	**0.3535**	0.0332	3.6918	5.5455
Top-decile lift	None	3.6101	1.7474	1.7185	2.6611	3.1917	4.2364	7.8310	2.9047	4.2910	3.6488	2.9258	**3.5000**
	ADASYN	3.5283	1.6751	**1.7316**	2.6480	2.9753	4.2951	7.8090	2.8857	4.2352	3.6697	2.8726	5.1364
	Borderline-SMOTE	3.5660	1.4817	1.5132	2.6621	2.7598	4.1040	7.8163	2.5415	**4.3589**	**3.7484**	**2.9113**	5.6818
	CLUS	3.2830	1.5794	1.5339	2.6020	2.8319	4.1187	7.1698	2.3314	3.8119	3.1560	2.8729	9.2727
	MWMOTE	3.4780	1.6687	1.5415	2.6621	3.2156	**4.3393**	7.8016	2.7901	4.1586	3.6488	2.8624	6.0909
	ROS	**3.6164**	**1.7538**	1.6936	2.6591	3.1915	4.2805	**7.8898**	2.6755	4.1603	3.6540	2.9045	4.4545
	RUS	3.4843	1.5922	1.6914	2.6702	2.4476	4.0746	7.5225	2.5034	4.1203	3.4705	2.9186	7.7273
	SMOTE	3.4906	1.7176	1.7164	2.6565	3.0955	4.3099	7.7869	**3.0194**	4.2561	3.6592	**2.9363**	4.3181
	SMOTE–ENN	3.5283	1.7283	1.7131	2.6454	**3.3116**	4.2806	7.8456	2.8858	4.2649	3.6592	2.8796	4.2727
	SMOTE–Tomek	3.5535	1.6900	1.7283	**2.6828**	3.1199	4.1775	7.7135	2.5226	4.2283	3.6855	2.9361	4.5454

Table 8 Experimental results for C4.5

Measure	*Method*	*Chile*	*Duke1*	*Duke2*	*KDD*	*K1*	*K2*	*K3*	*K4*	*K5*	*Tele1*	*UCI*	*AR*
AUC	None	0.6114	**0.5552**	**0.5921**	0.5676	0.5004	0.5574	0.7421	0.9599	0.6312	0.6368	**0.8591**	6.4545
	ADASYN	0.6518	0.5160	0.5388	0.5989	0.6554	0.6377	0.7466	0.9558	0.7492	0.7055	0.8333	5.2273
	Borderline-SMOTE	0.6335	0.5464	0.5651	0.6200	0.7215	0.6893	0.7737	0.9477	**0.7628**	0.7221	0.8356	**3.7273**
	CLUS	0.6517	0.5192	0.5394	0.6230	**0.7750**	0.7528	**0.8665**	0.9495	0.7484	0.7001	0.8552	3.4545
	MWMOTE	0.6357	0.5088	0.5203	0.5726	0.5675	0.6226	0.7414	0.9470	0.7089	0.7173	0.8447	7.3636
	ROS	0.6346	0.5027	0.5084	0.5285	0.5838	0.5946	0.7074	0.9342	0.6480	0.6550	0.8133	9.0909
	RUS	0.6296	0.5374	0.5619	**0.6296**	0.7620	**0.7780**	0.8555	**0.9621**	0.7473	0.7334	0.8570	**2.9091**
	SMOTE	**0.6564**	0.5250	0.5388	0.6046	0.6264	0.6477	0.7417	0.9434	0.7560	0.7068	0.8529	4.8636
	SMOTE–ENN	0.6479	0.5226	0.5364	0.5860	0.6272	0.6672	0.8440	0.9244	0.7405	0.6867	0.8328	6.3636
	SMOTE–Tomek	0.6319	0.5181	0.5365	0.5968	0.6167	0.6296	0.7588	0.9491	0.7626	**0.7346**	0.8393	5.5454
MP	None	**0.6951**	0.0000	0.0000	0.0366	0.0000	0.0130	0.8589	**2.2262**	**0.5195**	**0.7562**	5.3562	5.0455
	ADASYN	0.3908	0.0000	0.0000	0.0792	0.0169	0.1139	0.8725	2.1539	0.4301	0.2125	5.5034	4.8182
	Borderline-SMOTE	0.4776	0.0000	0.0000	0.0000	0.0128	0.0523	0.5200	1.8319	0.3276	0.4466	5.3850	6.6364
	CLUS	0.0146	0.0000	0.0000	0.0000	**0.0322**	0.0690	1.1065	2.1419	0.3670	0.0000	5.6588	5.8637
	MWMOTE	0.2820	0.0000	0.0000	0.0000	0.0000	0.0884	0.8624	2.1920	0.3663	0.4051	5.5864	5.9545
	ROS	0.4702	0.0000	0.0000	0.0000	0.0000	0.0736	0.7305	2.1864	0.3343	0.5768	5.3761	6.2273
	RUS	0.0000	0.0000	0.0000	0.0000	0.0000	0.1155	1.0696	2.1679	0.4273	0.0000	**5.6920**	5.7273
	SMOTE	0.4125	0.0000	0.0000	0.0679	0.0000	0.0188	0.8764	2.1820	0.4210	0.4293	5.5915	5.3182
	SMOTE–ENN	0.2388	0.0000	0.0000	**0.0913**	0.0049	0.1275	**1.2587**	2.0995	0.4159	0.0568	5.4896	5.0909
	SMOTE–Tomek	0.4169	0.0000	0.0000	0.0761	0.0000	**0.1561**	1.0091	2.1958	0.4423	0.2540	5.4704	**4.3182**
Top-decile lift	None	**5.7673**	**1.3499**	**1.5705**	1.0689	4.7970	**5.9268**	5.4215	9.0581	**6.0506**	**4.8807**	**5.1267**	**2.8182**
	ADASYN	3.2579	1.1522	1.4752	2.0474	3.3596	3.8832	6.0018	8.6186	3.3237	4.1551	4.9009	6.3181
	Borderline-SMOTE	3.1950	1.2117	1.3981	2.0938	3.9590	3.6478	6.3397	8.9052	4.3868	4.0315	4.7736	4.2727
	CLUS-3	2.0252	1.0990	1.0581	1.7986	3.6223	3.6329	7.2286	8.8671	4.7056	2.3853	4.1943	7.1818
	MWMOTE	3.0943	1.1968	1.2286	1.9798	2.8555	4.0742	5.7520	8.8478	4.2405	3.6383	4.6185	6.9091
	ROS	4.4340	0.6547	0.7604	1.7330	**5.9496**	5.8972	6.5454	8.7715	4.1081	3.2765	4.6996	6.2727
	RUS	2.3899	1.2840	1.3753	**2.5668**	2.5668	3.7212	6.7070	**9.1345**	4.7021	3.1874	4.7557	5.0454
	SMOTE	3.2704	1.2797	1.3883	2.0641	3.4797	3.6479	6.0092	8.8479	4.2858	3.6907	4.5616	5.2273
	SMOTE–ENN	3.1321	1.2478	1.3492	2.1675	3.7436	3.9273	**7.2653**	8.8097	4.2945	3.2503	4.3710	5.3636
	SMOTE–Tomek	3.2579	1.2840	1.3275	2.0504	3.3594	4.0157	6.2736	8.8479	4.1847	3.7117	4.4843	5.5000

Table 9 Experimental results for SVM

Measure	*Method*	*Chile*	*Duke1*	*Duke2*	*KDD*	*K1*	*K2*	*K3*	*K4*	*K5*	*Tele1*	*UCI*	*AR*
AUC	None	0.6495	0.5289	0.5351	0.5733	0.6431	0.7786	0.8672	0.529	0.7111	0.6014	0.8851	9.5455
	ADASYN	0.6827	0.5506	0.5506	0.6236	0.6763	0.8142	**0.9393**	**0.7480**	0.7814	0.7361	0.8874	5.0909
	Borderline-SMOTE	0.6955	0.5383	0.5428	0.6430	0.7011	0.7922	0.9161	0.7104	0.7699	0.7346	0.8836	6.9091
	CLUS	0.6953	0.5454	0.5402	**0.6818**	0.6716	0.8395	0.9241	0.6998	0.7416	**0.7843**	0.8676	6.4545
	MWMOTE	0.6897	0.5536	0.5555	0.6366	0.6767	0.8169	0.9348	0.7458	0.7796	0.7277	**0.8903**	4.5454
	ROS	0.6984	0.5471	**0.5564**	0.6379	**0.7074**	0.8317	0.9363	0.7314	**0.7856**	0.7380	0.8881	**3.4545**
	RUS	**0.7072**	0.5494	0.5443	0.6632	0.6634	**0.8438**	0.9259	0.7037	0.7607	0.7583	0.8775	5.4545
	SMOTE	0.6977	0.5523	0.5501	0.6268	0.7125	0.8072	0.9340	0.7209	0.7718	0.7374	0.8850	5.2727
	SMOTE–ENN	0.7000	**0.5730**	0.5545	0.6474	0.6983	0.8213	0.9319	0.7400	0.7722	0.7484	0.8820	3.7273
	SMOTE–Tomek	0.6928	0.5570	0.5492	0.6380	0.6980	0.8196	0.9296	0.7370	0.7759	0.7407	0.8885	4.5455
MP	None	0.6548	0.0000	0.0000	0.0000	0.0000	0.1345	0.2977	0.0305	0.0000	0.0000	4.3472	7.0000
	ADASYN	0.3220	0.0000	0.0000	0.0000	0.0000	0.1954	1.1087	0.0000	0.1696	0.6285	5.1060	5.0000
	Borderline-SMOTE	0.2901	0.0000	0.0000	0.0009	0.0024	0.0235	0.5452	0.0083	0.0417	0.3526	4.2116	7.0000
	CLUS	0.5134	0.0000	**0.0013**	**0.2594**	0.0000	0.1793	**1.1976**	0.0033	0.1073	0.0677	4.7650	4.9091
	MWMOTE	0.3356	0.0000	0.0000	0.0000	0.0000	0.1853	0.5202	0.0087	0.2240	0.7398	**5.1728**	4.9091
	ROS	0.2780	0.0000	0.0000	0.0000	0.0000	0.1735	0.4904	0.0000	0.2158	**0.7785**	5.0995	6.2273
	RUS	**0.5957**	0.0000	0.0000	0.0989	0.0024	0.2336	1.1442	0.0147	0.0679	0.3672	4.8419	**4.3636**
	SMOTE	0.3525	0.0000	0.0000	0.0000	**0.0025**	0.1772	1.1341	0.0000	**0.2346**	0.6048	4.8887	4.9545
	SMOTE–ENN	0.3179	0.0000	0.0000	0.1400	0.0000	0.1722	1.1932	0.0000	0.2136	0.2388	4.9543	5.8636
	SMOTE–Tomek	0.3480	0.0000	0.0000	0.0209	**0.0025**	**0.2350**	0.8401	0.0000	0.1558	0.5531	4.9599	4.7727
Top-decile lift	None	3.0377	1.1522	1.2721	1.3308	2.0638	3.5156	6.0606	2.2932	3.3032	1.2267	4.4875	8.7273
	ADASYN	3.4151	1.3669	1.4589	2.1433	1.6318	4.1628	7.8530	**2.6563**	3.9460	3.6697	**4.5862**	4.7727
	Borderline-SMOTE	3.3459	1.2925	1.4524	2.4592	2.1596	3.8392	7.6767	2.2166	3.7283	3.7431	4.3425	6.5000
	CLUS	3.4591	1.3265	1.2036	**2.5218**	2.3997	4.0010	7.8383	2.2169	3.0053	3.2870	4.1871	6.3181
	MWMOTE	3.4151	1.4009	**1.5045**	2.2992	2.1357	4.0010	7.9192	2.5990	3.7980	3.6121	4.4947	4.6818
	ROS	3.4465	1.3095	1.4361	2.3719	2.2558	4.0010	**7.9265**	2.4651	**4.1203**	3.7379	4.4873	4.4545
	RUS	**3.6038**	1.3754	1.3286	2.4259	2.0397	**4.3100**	7.8457	2.1213	3.3049	**3.7588**	4.3073	5.4545
	SMOTE	3.3396	1.2861	1.4383	2.3941	**2.4958**	4.1922	7.8751	2.2169	3.8502	3.6802	4.4553	5.4091
	SMOTE–ENN	3.2453	**1.5093**	1.4926	2.5198	2.1599	4.1774	7.8090	2.5608	3.8624	3.6907	4.1870	4.5454
	SMOTE–Tomek	3.3208	1.4243	1.4730	2.4592	2.1598	4.2952	7.8016	2.5226	3.8851	3.6855	4.5227	**4.1364**

Table 10 Experimental results for random forests

Measure	*Method*	*Chile*	*Duke1*	*Duke2*	*KDD*	*K1*	*K2*	*K3*	*K4*	*K5*	*Tele1*	*UCI*	*AR*
AUC	None	0.7047	0.5647	0.5745	0.6934	0.7896	0.8730	0.9248	**0.9745**	0.8110	0.7944	0.8704	7.5455
	ADASYN	0.6922	0.5605	0.5765	0.7006	0.7989	0.8823	0.9307	0.9671	0.8181	0.8063	0.9114	6.6818
	Borderline-SMOTE	0.6870	0.5623	0.5837	0.7057	0.7982	0.8692	0.9247	0.9724	0.8150	0.8149	0.9069	6.7273
	CLUS	0.7022	0.5726	0.5810	0.7059	0.8220	0.8782	0.9241	0.9675	0.8186	0.7988	0.9030	5.7273
	MWMOTE	0.7083	0.5586	0.5602	0.7011	0.8115	0.8883	0.9290	0.9732	0.8174	0.8065	0.9147	5.0909
	ROS	**0.7138**	0.5701	0.6062	0.7069	0.8097	0.8856	0.9311	0.9720	0.8207	**0.8218**	0.9111	**2.8182**
	RUS	0.6791	**0.5782**	**0.6081**	**0.7126**	**0.8398**	0.8825	0.9235	0.9658	0.8216	0.8169	0.9101	4.5455
	SMOTE	0.6973	0.5559	0.5800	0.7017	0.8018	0.8851	0.9274	0.9718	**0.8221**	0.8085	**0.9150**	5.0455
	SMOTE–ENN	0.6912	0.5630	0.5865	0.7038	0.7828	0.8846	**0.9330**	0.9660	0.8181	0.8162	0.9094	5.7727
	SMOTE–Tomek	0.6931	0.5651	0.5757	0.7009	0.8168	**0.8865**	0.9274	0.9679	0.8189	0.8110	0.9112	5.0455
MP	None	0.7304	0.0000	0.0000	0.2879	0.1015	0.0000	0.1749	1.9185	0.5677	**1.3306**	4.9540	6.0909
	ADASYN	0.7009	0.0000	0.0000	0.2642	0.0872	0.0616	0.3412	1.9906	0.5739	0.7906	4.7525	7.1818
	Borderline-SMOTE	0.6580	0.0000	0.0000	0.3037	0.0869	0.0697	0.9498	2.0790	0.5884	1.3087	6.2336	4.1818
	CLUS	0.7265	0.0000	0.0000	0.2937	0.1157	**0.1130**	**1.0250**	1.7352	0.5792	0.0399	5.1468	5.0000
	MWMOTE	0.7448	0.0000	0.0000	0.2991	0.0259	0.0649	0.8846	1.9108	0.5759	1.3183	5.0335	5.0000
	ROS	**0.7768**	0.0000	0.0000	0.3002	0.0286	0.1334	0.0417	**2.1796**	0.4098	1.2994	**6.3261**	4.4545
	RUS	0.7377	0.0000	0.0000	**0.3217**	**0.1708**	0.1121	0.8241	1.7839	**0.5886**	1.2327	5.1271	**4.0909**
	SMOTE	0.6953	0.0000	0.0000	0.2676	0.0000	0.0625	0.3584	2.0513	0.5747	1.2382	4.9291	6.9091
	SMOTE–ENN	0.6974	0.0000	0.0000	0.2855	0.0209	0.1036	0.8183	1.9105	0.5309	1.1527	5.8707	6.6364
	SMOTE–Tomek	0.7175	0.0000	0.0000	0.2418	0.1462	0.1093	0.6863	2.0552	0.5696	1.2191	5.7918	5.4545
Top-decile lift	None	3.8239	1.3456	1.6675	2.6182	3.8877	4.0011	7.7796	**9.0009**	4.9635	4.3250	4.6853	5.5454
	ADASYN	3.7610	1.3605	1.6153	2.5465	3.9597	4.0745	7.8090	8.9626	4.9513	3.7903	5.5403	5.7727
	Borderline-SMOTE	3.6792	1.4732	1.6675	2.6359	3.8637	3.8098	7.8090	8.9054	4.9339	4.3303	5.4872	5.7727
	CLUS	3.7170	1.5327	1.7142	2.6359	4.2956	4.0598	6.8540	8.8097	5.0140	3.5020	5.1656	5.5909
	MWMOTE	3.9245	1.3796	1.4948	2.6126	3.8397	4.3393	7.8824	8.9626	4.9234	4.2988	5.5120	4.9545
	ROS	**3.9686**	1.3690	1.8847	**2.7580**	3.7917	4.2364	7.8237	8.9244	**5.0436**	**4.4823**	5.5332	**3.0909**
	RUS	3.6792	**1.5476**	**1.9260**	2.7494	**4.6316**	4.2070	7.7135	8.8672	4.9461	4.3041	5.4272	4.5455
	SMOTE	3.6792	1.2861	1.6349	2.5581	3.5517	**4.3834**	7.7869	8.9436	4.9443	4.1678	**5.5579**	6.3636
	SMOTE–ENN	3.6478	1.3605	1.6968	2.5919	3.5036	4.1922	**7.9559**	8.7715	4.9635	4.2569	5.1409	6.5455
	SMOTE–Tomek	3.6101	1.3669	1.5708	2.5430	4.0795	4.0009	7.8310	8.8480	4.9565	4.2045	5.4944	6.8181

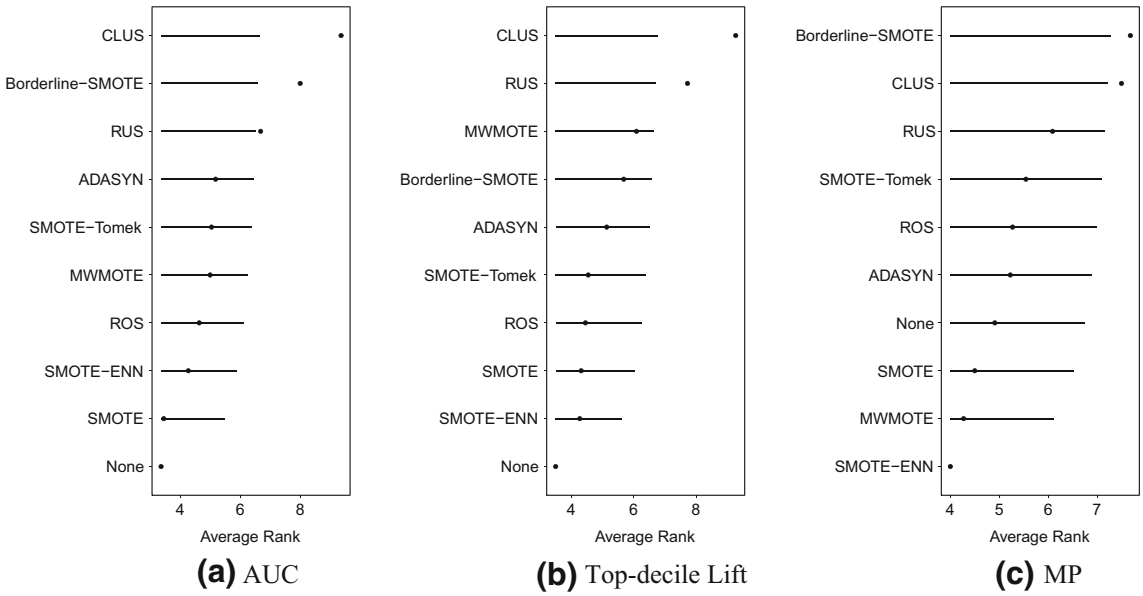

Figure 2 Rankings of sampling techniques with logistic regression.

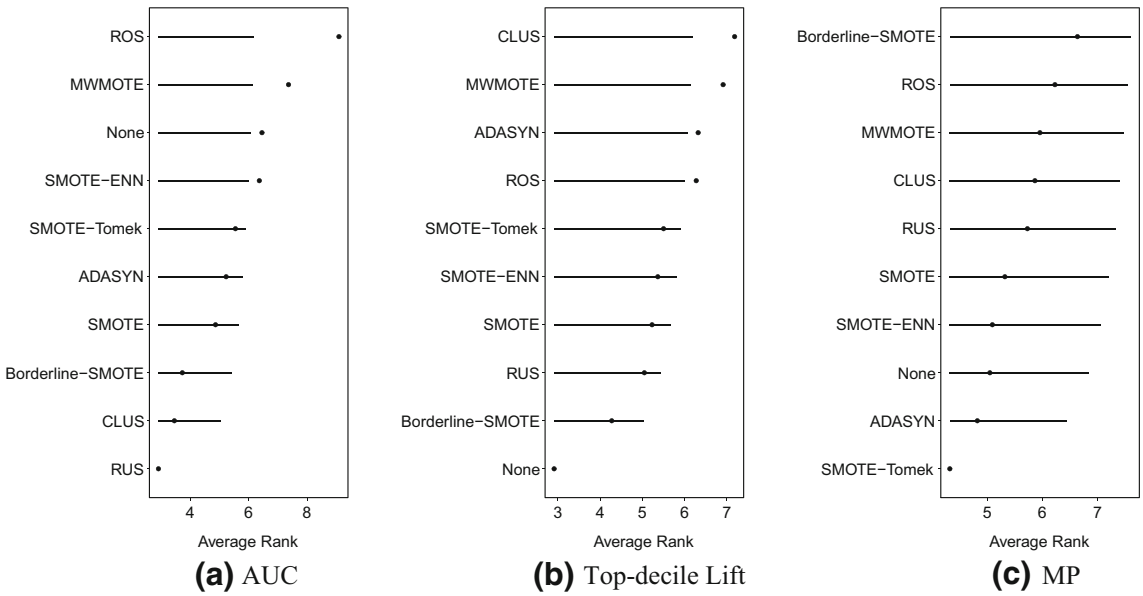

Figure 3 Rankings of sampling techniques with C4.5 decision tree.

methods in churn prediction. This less balanced strategy has the best average performance in terms of MP and top-decile lift measures. Although the performance in terms of AUC is slightly worse than the roughly balanced strategy, we could not observe a statistical difference between them, which confirms the claims of Weiss and Provost (2003) that AUC is insensitive to class distributions in a certain range. It hence appears unnecessary for sampling techniques to make balanced distributions, which is usually more computationally expensive. By transforming the data sets into some less imbalanced class distributions, the problem of class imbalance will already be solved.

Secondly—our experimental results show that sampling methods have different reactions to different evaluation metrics. One important novelty of our paper is that we introduce the MP measure into our experimental comparison. This metric gives us significant different rankings of sampling methods as Figures 2, 3 and 4 show, which provides us with some insight into the usage of sampling techniques from the perspective of cost–benefit optimization. Meanwhile, the influence of the evaluation metric is interrelated with the type of classifier. More specifically, sampling strategies have no influence on logistic regression for all the measures. At the same time, C4.5 decision trees and random forests present totally different reaction patterns with the

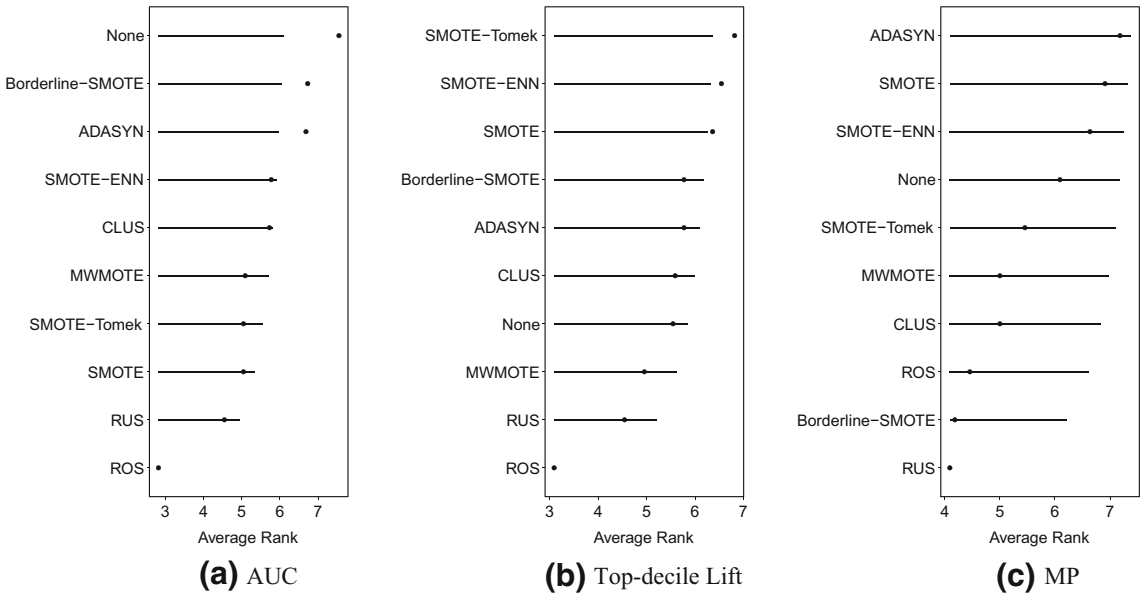

Figure 4 Rankings of sampling techniques with random forests.

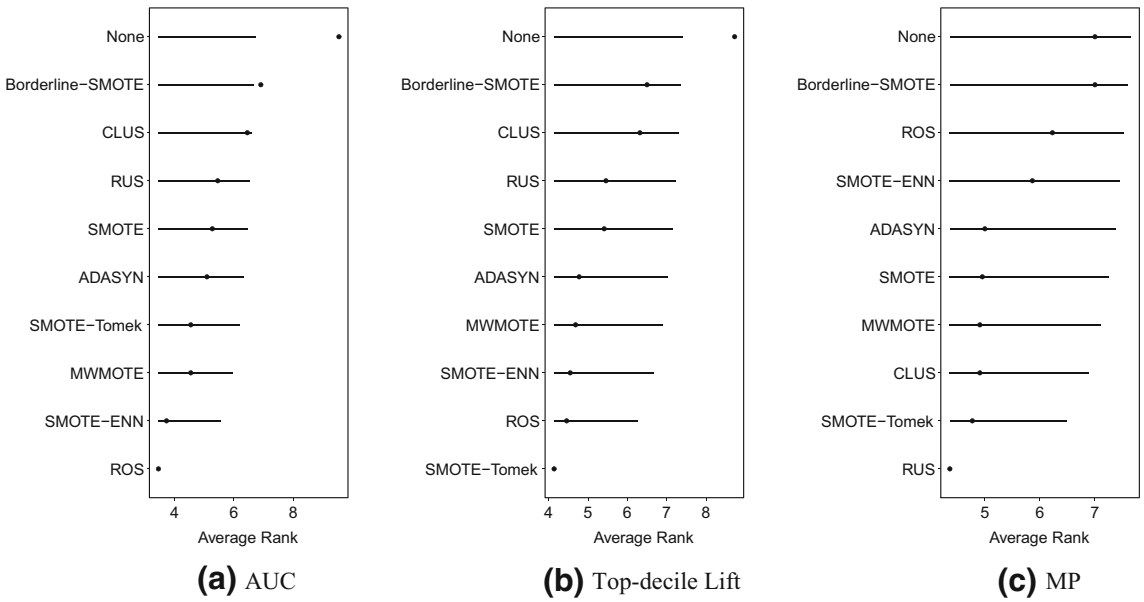

Figure 5 Rankings of sampling techniques with SVM.

three evaluation measures. Most sampling techniques improve the performance measured by AUC. RUS is the best for C4.5 decision trees, and ROS is the best option for random forests in this situation. However, sampling methods seem to bring no benefit in terms of the top-decile lift and the MP measure. SVM can gain benefits from resampling, and different sampling methods work best with different measures. ROS is suggested for AUC, while it is also a good option when using top-decile lift, as well as SMOTE–Tomek. Meanwhile, RUS is suitable to use when combining the MP measure. The experimental results rule against the common belief that sampling is independent of the consequent learning steps. In other words, people should choose appropriate suitable sampling techniques in a specific domain. Our study offers some guidelines in the churn prediction domain.

Thirdly—it is important to notice that current sampling techniques do not seem to significantly improve the MP measure of random forests and SVM. Therefore, the development of cost–benefit sensitive sampling methods appears as an interesting avenue to explore.

6. Conclusions

Customer churn management is an indispensable part of all business, in which churn prediction is the key step. Usually, there is a relatively small proportion of churners in the customer base. How to deal with the class imbalance and identify the minority churner group from a large number of non-churners is the core of successful customer churn prediction. In our study, we present a comprehensive comparison of sampling methods for dealing with this issue. More specially, a recently developed MP measure is used to evaluate the results from the perspective of cost–benefit. The experimental results show that the effectiveness of sampling methods is affected by the evaluation metric as well as the classifier. For some classifiers, applying sampling methods offers little help, while for other classifiers, the influence depends on the performance measure for other classifiers. We have recommended suitable sampling strategies for each combination of classifier evaluation metric. We also find that a sampling rate around a less balanced class ratio is a good general option in practice. Our research provides some guidance to choose suitable methods to deal with class balance in practice. In the future, we plan to compare the sampling methods with other algorithm-level solutions using the profit-based measure.

Acknowledgements—This work is supported by National Natural Science Foundation of China (Grant No. 71401115) and the MOE (Ministry of Education in China) Youth Project of Humanities and Social Sciences (Grant No. 13YJC630249). Bing Zhu is supported by postdoctoral fellowships from the China Scholarship Council (CSC).

References

Ali OG and Ariturk U (2014). Dynamic churn prediction framework with more effective use of rare event data: The case of private banking. *Expert Systems with Applications* 41(17):7889–7903.

Baesens B (2014). *Analytics in a big data world*. Wiley, New York.

Barua S, Islam M and Yao X (2014). MWMOTE-majority weighted minority oversampling technique for imbalanced data set learning. *IEEE Transactions on Knowledge and Data Engineering* 26(2):405–425.

Batista GE, Prati RC and Monard MC (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM Sigkdd Explorations Newsletter* 6(1):20–29.

Bhattacharya CB (1998). When customers are members: Customer retention in paid membership contexts. *Journal of the Academy of Marketing Science* 26(1):31–44.

Bradley AP (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition* 30(7):1145–1159.

Burez J and Van den Poel D (2009). Handling class imbalance in customer churn prediction. *Expert Systems with Applications* 36(3):4626–4636.

Chawla NV, Bowyer KW, Hall KO and Kegelmeyer WP (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research* 16(3):321–357.

Chawla NV (2003). C4.5 and imbalanced data sets: Investigating the effect of sampling method, probabilistic estimate, and decision tree structure. In: *Proceedings of the ICML*.

Chen ZY, Fan ZP, and Sun M (2012). A hierarchical multiple kernel support vector machine for customer churn prediction using longitudinal behavioral data. *European Journal of Operational Research* 223(2):461–472.

Colgate M and Danaher P (2000). Implementing a customer relationship strategy: The asymmetric impact of poor versus excellent execution. *Journal of the Academy of Marketing Science* 28(3):375–387.

Coussement K and Van den Poel D (2008). Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques. *Expert Systems with Applications* 34(1):313–327.

Coussement K and Van den Poel D (2008). Integrating the voice of customers through call center emails into a decision support system for churn prediction. *Information & Management* 45(3):164–174.

De Bock, KW and Van den Poel D (2011). An empirical evaluation of rotation-based ensemble classifiers for customer churn prediction. *Expert Systems with Applications* 38(10):12293–12301.

Demšar J (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7:1–30.

Farquad MAH, Ravi V and Raju SN (2014). Churn prediction using comprehensible support vector machine: An analytical CRM application. *Applied Soft Computing* 19:31–40.

Ganesh J, Arnold M, and Reynolds K (2000). Understanding the customer base of service providers: An examination of the differences between switchers and stayers. *Journal of Marketing* 64(3):65–87.

García S and Herrera F (2009). Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy. *Evolutionary Computation* 17(3):275–306.

García S, Sánchez JS and Mollineda RA (2012). On the effectiveness of preprocessing methods when dealing with different levels of class imbalance. *Knowledge-Based Systems* 25(1):13–21.

Hadden J, Tiwari A, Roy R and Ruta D (2007). Computer assisted customer churn management: State-of-the-art and future trends. *Computers & Operations Research* 34(10):2902–2917.

Hand DJ (2009). Measuring classifier performance: A coherent alternative to the area under the ROC curve. *Machine Learning* 77(1):103–123.

Han H, Wang WY and Mao BH (2005). Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In: *Proceedings international conference on intelligent computing*, pp 878–887.

He H, Bai Y, Garcia EA and Li S (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In: *Proceedings of IEEE international joint conference on neural networks*, pp 1322–1328.

He H and Garcia EA (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering* 21(9):1263–1284.

Holm S (1970). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics* 6:65–70.

Jahromi AT, Stakhovych S, and Ewing M (2014). Managing B2B customer churn, retention and profitability. *Industrial Marketing Management* 43(7):1258–1268.

Jamal Z and Bucklin RE (2006). Improving the diagnosis and prediction of customer churn: A heterogeneous hazard modeling approach. *Journal of Interactive Marketing* 20(3–4):16–29.

Keramati A, Jafari-Marandi R, Aliannejadi M, Ahmadian L, Mozaffari M and Abbasi U (2014). Improved churn prediction in telecommunication industry using data mining techniques. *Applied Soft Computing* 24:994–1012.

Khoshgoftaar T, Seiffert C, Van Hulse J, Napolitano A and Folleco A (2007). Learning with limited minority class data. In: *Proceeding of international conference on machine learning and applications*, pp 348–353.

Kubat M and Matwin S (1997). Addressing the curse of imbalanced training sets: One-sided selection. In: *Proceedings of international conference on machine learning*, pp 179–186.

Lima E, Mues C, Baesens B (2009). Domain knowledge integration in data mining using decision tables: Case studies in churn prediction. *Journal of the Operational Research Society* 8(8):1096–1106.

Lemmens A and Croux C (2006). Bagging and boosting classification trees to predict churn. *Journal of Marketing Research (JMR)* 43(2):276–286.

López V, Fernández A, García S, Palade V and Herrera F (2013). An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences* 250:113–141.

Marqués AI, García V, Sánchez JS (2013). On the suitability of resampling techniques for the class imbalance problem in credit scoring. *Journal of the Operational Research Society* 64(7): 1060–1070.

Neslin S, Gupta S, Kamakura W, Lu J and Mason C (2006). Detection defection: Measuring and understanding the predictive accuracy of customer churn models. *Journal of Marketing Research* 43(2): 204–211.

Raeder T, Forman G and Chawla NV (2012). Learning from imbalanced data: Evaluation matters. In: *Data mining: Foundations and intelligent paradigms*. Springer, pp 315–331.

Seiffert C, Khoshgoftaar TM, Van Hulse J and Folleco A (2014). An empirical study of the classification performance of learners on imbalanced and noisy software quality data. *Information Sciences* 259:571–595.

Sun Y, Wong AK and Kamel MS (2009). Classification of imbalanced data: A review. *International Journal of Pattern Recognition and Artificial Intelligence* 23(4):687–719.

Van Hulse J, Khoshgoftaar TM and Napolitano A (2007). Experimental perspectives on learning from imbalanced data. In: *Proceedings of the 24th international conference on machine learning*, pp 935–942.

Verbeke W, Martens D, Mues C and Baesens B (2011). Building comprehensible customer churn prediction models with advanced rule induction techniques. *Expert Systems with Applications* 38(3):2354–2364.

Verbeke W, Dejaeger K, Martens D, Hur J and Baesens B (2012). New insights into churn prediction in the telecommunication sector: A profit driven data mining approach. *European Journal of Operational Research* 218(1):211–229.

Weiss GM (2004). Mining with rarity: A unifying framework. *ACM Sigkdd Explorations Newsletter* 6(1):7–19.

Weiss GM and Provost F (2003). Learning when training data are costly: The effect of class distribution on tree induction. *Journal of Artificial Intelligence Research* 19:315–354.

Xiao J, Xie L, He C, and Jiang X (2012). Dynamic classifier ensemble model for customer classification with imbalanced distribution. *Expert Systems with Applications* 39(3):3668–3675.

Xie Y, Li X, Ngai E and Ying M (2009). Customer churn prediction using improved balanced random forests. *Expert Systems with Applications* 36(3):5445–5449.

Yang Q and Wu X (2006). Challenging problems in data mining research. International Journal of Information Technology Decision Making 5(4):597–604.

Yang P, Xu L, Zhou BB, Zhang Z and Zomaya AY (2009). A particle swarm based hybrid system for imbalanced medical data sampling. *BMC Genomics* 10(Suppl 3):S34.

Yen SJ and Lee YS (2009). Cluster-based under-sampling approaches for imbalanced data distributions. *Expert Systems with Applications* 36(3):5718–5727.

Zeithaml V, Berry L and Parasuraman A (1996). The behavioural consequences of service quality. *Journal of Marketing* 60(2):31–46.

Received 14 March 2016;
accepted 19 December 2016