2015 IEEE Symposium Series on Computational Intelligence

Calibrating Probability with Undersampling for Unbalanced Classification

Andrea Dal Pozzolo*, Olivier Caelen[†], Reid A. Johnson[‡], Gianluca Bontempi*[§]
*Machine Learning Group, Computer Science Department, Université Libre de Bruxelles, Brussels, Belgium.
Email: adalpozz@ulb.ac.be
[†]Fraud Risk Management Analytics, Worldline S.A., Brussels, Belgium.
Email: olivier.caelen@worldline.com
[‡] iCeNSA, Computer Science and Engineering Department, University of Notre Dame, Notre Dame IN, USA.
Email: rjohns15@nd.edu
[§]Interuniversity Institute of Bioinformatics in Brussels $(\mathrm{IB})^2$, Brussels, Belgium.
Email: gbonte@ulb.ac.be

***Abstract*—Undersampling is a popular technique for unbalanced datasets to reduce the skew in class distributions. However, it is well-known that undersampling one class modifies the priors of the training set and consequently biases the posterior probabilities of a classifier [9]. In this paper, we study analytically and experimentally how undersampling affects the posterior probability of a machine learning model. We formalize the problem of undersampling and explore the relationship between conditional probability in the presence and absence of undersampling. Although the bias due to undersampling does not affect the ranking order returned by the posterior probability, it significantly impacts the classification accuracy and probability calibration. We use Bayes Minimum Risk theory to find the correct classification threshold and show how to adjust it after undersampling. Experiments on several real-world unbalanced datasets validate our results.**

I. Introduction

In several binary classification problems, the two classes are not equally represented in the dataset. For example, in fraud detection, fraudulent transactions are normally outnumbered by genuine ones [8]. When one class is underrepresented in a dataset, the data is said to be unbalanced. In such problems, typically, the minority class is the class of interest. Having few instances of one class means that the learning algorithm is often unable to generalize the behavior of the minority class well, hence the algorithm performs poorly in terms of predictive accuracy [16].

A common strategy for dealing with unbalanced classification tasks is to under-sample the majority class in the training set before learning a classifier [1]. The assumption behind this strategy is that in the majority class there are many redundant observations and randomly removing some of them does not change the estimation of the within-class distribution. If we make the assumption that training and testing sets come from the same distribution, then when the training is unbalanced, the testing set has a skewed distribution as well. By removing majority class instances, the training set is artificially rebalanced. As a consequence, we obtain different distributions for the training and testing sets, violating the basic assumption in machine learning that the training and testing sets are drawn from the same underlying distribution.

In this paper, we study the impact of the bias introduced by undersampling on classification tasks with unbalanced data. We start by discussing literature results showing how the posterior probability of an algorithm learnt in the presence of undersampling is related to the conditional probability of the original distribution. Using synthetic data we see that the larger the overlap between the two within-class distributions (i.e. the greater the non-separability of the classification task), the larger the bias in the posterior probability. The mismatch between the posterior probability obtained with the original dataset and after undersampling is assessed in terms of loss measure (Brier Score), predictive accuracy (G-mean) and ranking (AUC).

Based on the previous works of Saerens et al. [21] and Elkan [13], we propose an analytical method to correct the bias introduced by undersampling that can produce well-calibrated probabilities. The method is equivalent to adjusting the posterior probability in the presence of new priors. The use of unbiased probability estimates requires an adjustment to the probability threshold used to classify instances. When using class priors as misclassification costs, we show that this new threshold corresponds to the one used before undersampling. In order to have complete control over the data generation process, we have first recourse to synthetic datasets. This allows us to simulate problems of different difficulty and see the impact of undersampling on the probability estimates. To confirm the results obtained with the simulated data, we also run our experiments on several UCI datasets and a real-world fraud detection dataset made available to the public.

This paper has the following contributions. First, we review how undersampling can induce a bias in the posterior probabilities generated by machine learning methods. Second, we leverage this understanding to develop an analytical method that can counter and reduce this bias. Third, we show how to use unbiased probability estimates for decision making in unbalanced classificiation. We note that while the framework we derive in this work is theoretically equivalent to the problem of a change in class priors [21], our perspective is different. We interpret undersampling as a problem of sample selection bias, wherein the bias is not intrinsic to the data but rather introduced artificially [19].

978-1-4799-7560-0/15 $31.00 © 2015 IEEE
DOI 10.1109/SSCI.2015.33

The paper is organized as follows. Section II introduces some well-known methods for unbalanced datasets and section III formalizes the sampling selection bias due to undersampling. Undersampling is responsible for a shift in the posterior probability which leads to biased probability estimates, for which we propose a corrective method. Section IV shows how to set the classification threshold to take into account the change in the priors. Finally, section VI uses real-world datasets to validate the probability transformation presented in section III and the use of the classification threshold proposed in IV.

II. SAMPLIN´ FOR UNBALANCED CLASSIFICATION

Let us consider a binary classification task where the distribution of the target class is highly skewed. When the data is unbalanced, standard machine learning algorithms that maximise overall accuracy tend to classify all observations as majority class instances [16]. This translates into poor accuracy on the minority class (low recall), which is typically the class of interest. There are several methods that deal with this problem, which we can distinguish between methods that operate at the data and algorithmic levels [6].

At the data level, the unbalanced strategies are used as a pre-processing step to re-balance the two classes before any algorithm is applied. At the algorithmic level, algorithms are themselves adjusted to deal with the minority class detection [2]. Here we will restrict ourselves to consider a subset of data-level methods known as sampling techniques.

Undersampling [11] consists of down-sizing the majority class by removing observations at random until the dataset is balanced. In an unbalanced problem, it is often realistic to assume that many observations of the majority class are redundant and that by removing some of them at random the data distribution will not change significantly. However the risk of removing relevant observations from the dataset is still present, since the removal is performed in an unsupervised manner. In practice, this technique is often adopted since it is simple and speeds up the learning phase.

Oversampling [11] consists of up-sizing the minority class at random, decreasing the level of class imbalance. By replicating the minority class until the two classes have equal frequency, oversampling increases the risk of over-fitting by biasing the model towards the minority class. Other drawbacks of the approach are that it does not add any new *valuable* minority examples and that it increases the training time. This can be particularly ineffective when the original dataset is fairly large.

SMOTE [7] over-samples the minority class by generating synthetic minority examples in the neighborhood of observed ones. The idea is to form new minority examples by interpolating between examples of the same class. This has the effect of creating clusters around each minority observation.

In this paper we focus on understanding how undersampling affects the posterior probability of a classification algorithm.

III. THE IMPACT OF SAMPLIN´ ON POSTERIOR PROBABILITIES

In binary classification we typically learn a model on training data and use it to generate predictions (class or posterior probability) on a testing set with the assumption that both come from the same distribution. When this assumption does not hold, we encounter the so-called problem of sampling selection bias [19]. Sampling selection bias can occur due to a bad choice of the training set. For example, consider the problem where a bank wants to predict whether someone who is applying for a credit card will be able to repay the credit at the end of the month. The bank has data available on customers whose applications have been approved, but has no information on rejected customers. This means that the data available to the bank is a biased sample of the whole population. The bias in this case is intrinsic to the dataset collected by the bank.

A. Sample Selection Bias due to undersampling

Rebalancing unbalanced data is just the sample selection bias problem with a known selection bias introduced by design (rather than by constraint or accident) [19]. In this section, we investigate the sampling selection bias that occurs when undersampling a skewed training set.

To begin, let us consider a binary classification task where the goal is to learn a classifier $f : R^n \to \{0, 1\}$, where $\mathbf{X} \in R^n$ is the input and $\mathbf{Y} \in \{0, 1\}$ the output domain. Let us call class 0 negative and class 1 positive. Further, assume that the number of positive observations is small compared to the number of negatives, with rebalancing performed via undersampling.

Let us denote as $(\mathcal{X}, \mathcal{Y})$ the original unbalanced training sample and as (X, Y) a balanced sample of $(\mathcal{X}, \mathcal{Y})$. This means that $(X, Y) \subset (\mathcal{X}, \mathcal{Y})$ and it contains a subset of the negatives in $(\mathcal{X}, \mathcal{Y})$. Let us define s as a random binary selection variable for each of the N samples in $(\mathcal{X}, \mathcal{Y})$, which takes the value 1 if the point is in (X, Y) and 0 otherwise. It is possible to derive the relationship between the posterior probability of a model learnt on a balanced subset and the one learnt on the original unbalanced dataset.

We assume that the selection variable s is independent of the input x given the class y (*class-dependent selection*): $p(s|y, x) = p(s|y)$. This assumption implies $p(x|y, s) = p(x|y)$, i.e. by removing observation at random in the majority class we do not change within-class distributions. With undersampling there is a change in the prior probabilities ($p(y|s = 1) \neq p(y)$) and as a consequence the class-conditional probabilities are different as well, $p(y|x, s = 1) \neq p(y|x)$. The probability that a point (x, y) is included in the balanced training sample is given by $p(s = 1|y, x)$. Let the sign $+$ denote $y = 1$ and $-$ denote $y = 0$, e.g. $p(+, x) = p(y = 1, x)$ and $p(-, x) = p(y = 0, x)$. From Bayes' rule, using $p(s|y, x) = p(s|y)$, we can write:

$$p(+|x, s = 1) = \frac{p(s = 1|+)p(+|x)}{p(s = 1|+)p(+|x) + p(s = 1|-)p(-|x)} \tag{1}$$

As shown in our previous work [9], since $p(s = 1|+) = 1$ we can write (1) as:

$$p(+|x, s = 1) = \frac{p(+|x)}{p(+|x) + p(s = 1|-)p(-|x)} \tag{2}$$

Let us denote $\beta = p(s = 1|-)$ as the probability of selecting a negative instance with undersampling, $p = p(+|x)$ as the posterior probability of the positive class on the original dataset, and $p_s = p(+|x, s = 1)$ as the posterior probability after sampling. We can rewrite equation (2) as:

$$p_s = \frac{p}{p + \beta(1 - p)} \tag{3}$$

Using (3) we can obtain an expression of p as a function of p_s:

$$p = \frac{\beta p_s}{\beta p_s - p_s + 1} \tag{4}$$

Balancing an unbalanced problem corresponds to the case when $\beta = \frac{p(+)}{p(-)} \approx \frac{N^+}{N^-}$, where N^+ and N^- denote the number of positive and negative instances in the dataset. In the following we will assume that $\frac{N^+}{N^-}$ provides an accurate estimation of the ratio of the prior probabilities. For such level of β, a small variation at the high values of p_s induces a large change in p, while the opposite occurs for small values of p_s [9]. When $\beta = 1$, all the negative instances are used for training, while for $\beta < 1$, a subset of negative instances are included in the training set. As β decreases towards $\frac{N^+}{N^-}$, the resulting training set becomes more balanced. Note that $\frac{N^+}{N^-}$ is the minimum value for β, as for $\beta < \frac{N^+}{N^-}$ we would have more positives than negatives.

Let's suppose we have an unbalanced problem where the positives account for 10% of 10,000 observations (i.e., we have 1,000 positives and 9,000 negatives). Suppose we want to have a balanced dataset $\beta = \frac{N^+}{N^-} \approx 0.11$, where $\approx 88.9\%$ (8000/9000) of the negative instances are discharged. Table I shows how, by reducing β, the original unbalanced dataset becomes more balanced and smaller as negative instances are removed. After undersampling, the number of negatives is $N_s^- = \beta N^-$, while the number of positives stays the same $N_s^+ = N^+$. The percentage of negatives ($perc^-$) in the dataset decreases as $N_s^- \rightarrow N^+$.

TABLE I. UNDERSAMPLING A DATASET WITH 1,000 POSITIVES IN 10,000 OBSERVATIONS. N_s DEFINES THE SIZE OF THE DATASET AFTER UNDERSAMPLING AND N_s^- (N_s^+) THE NUMBER OF NEGATIVE (POSITIVE) INSTANCES FOR A GIVEN β. WHEN $\beta = 0.11$ THE NEGATIVE SAMPLES REPRESENT 50% OF THE OBSERVATIONS IN THE DATASET.

N_s	N_s^-	N_s^+	β	$perc^-$
2,000	1,000	1,000	0.11	50.00
2,800	1,800	1,000	0.20	64.29
3,700	2,700	1,000	0.30	72.97
4,600	3,600	1,000	0.40	78.26
5,500	4,500	1,000	0.50	81.82
6,400	5,400	1,000	0.60	84.38
7,300	6,300	1,000	0.70	86.30
8,200	7,200	1,000	0.80	87.80
9,100	8,100	1,000	0.90	89.01
10,000	9,000	1,000	1.00	90.00

B. Bias and class separability

In this section we are going to show how the impact of bias depends on the separability nature of the classification task. Let ω^+ and ω^- denote the class conditional probabilities $p(x|+)$ and $p(x|-)$, and π^+ (π_s^+) the class priors before (after) undersampling. It is possible to derive the relation between the bias and the difference $\delta = \omega^+ - \omega^-$ between the class conditional distributions. From Bayes' theorem we have:

$$p = \frac{\omega^+\pi^+}{\omega^+\pi^+ + \omega^-\pi^-} \tag{5}$$

Suppose $\delta = \omega^+ - \omega^-$, we can write (5) as:

$$p = \frac{\omega^+\pi^+}{\omega^+\pi^+ + (\omega^+ - \delta)\pi^-} = \frac{\omega^+\pi^+}{\omega^+(\pi^+ + \pi^-) - \delta\pi^-} = \frac{\omega^+\pi^+}{\omega^+ - \delta\pi^-} \tag{6}$$

since $\pi^+ + \pi^- = 1$. Similarly, since ω^+ does not change with undersampling:

$$p_s = \frac{\omega^+\pi_s^+}{\omega^+ - \delta\pi_s^-} \tag{7}$$

Now we can write $p_s - p$ as:

$$p_s - p = \frac{\omega^+\pi_s^+}{\omega^+ - \delta\pi_s^-} - \frac{\omega^+\pi^+}{\omega^+ - \delta\pi^-} \tag{8}$$

Since $p_s \geq p$ because of (3), $1 \geq p_s \geq 0$ and $1 \geq p \geq 0$ we have: $1 \geq p_s - p \geq 0$. In Figure 1 we plot $p_s - p$ as a function of δ when $\pi_s^+ = 0.5$ and $\pi^+ = 0.1$. For small values of the class conditional densities it appears that the bias takes the highest values for δ values close to zero. This means that the bias is higher for similar class conditional probabilities (i.e. low separable configurations).

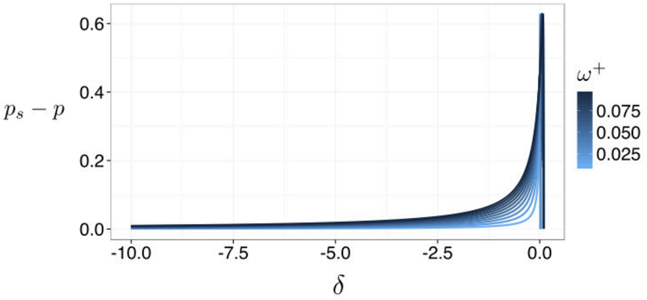

Fig. 1. $p_s - p$ as a function of δ, where $\delta = \omega^+ - \omega^-$ for values of $\omega^+ \in \{0.01, 0.1\}$ when $\pi_s^+ = 0.5$ and $\pi^+ = 0.1$. Note that δ is upper bounded to guarantee $1 \geq p_s \geq 0$ and $1 \geq p \geq 0$.

C. Adjusting posterior probabilities to new priors

Equation (3) shows how the conditional distribution of the balanced configuration relates to the conditional distribution in the original unbalanced setting. However, after a classification model is learnt on a balanced training set, it is normally used to predict a testing set, which is likely to have an unbalanced distribution similar to the original training set. This means that the posterior probability of a model learnt on the balanced training set should be adjusted for the change in priors between the training and testing sets. In this paper we propose to use equation (4) to correct the posterior probability estimates after undersampling. Let us call p' the bias-corrected probability obtained from p_s using (4):

$$p' = \frac{\beta p_s}{\beta p_s - p_s + 1} \tag{9}$$

Equation (9) can be seen as a special case of the framework proposed by Saerens et al. [21] and Elkan [13] for correcting the posterior probability in the case of testing and training sets sharing the same priors (see Appendix). When we know the priors in the testing set we can correct the probability with Elkan's and Saerens' equations. However, these probabilities are usually unknown and must be estimated. If we make the assumption that training and testing have the same priors we can used (9) for calibrating p_s. Note that the above transformation will not affect the ranking produced by p_s. Equation (9) defines a monotone transformation, hence the ranking of p_s will be the same as p'. While p is estimates using all the samples in the unbalanced dataset, p_s and p' are computed considering a subset of the original samples and therefore their estimations are subjected to higher variance [9]. The variance effect is typically addressed by the use of averaging strategies (e.g. UnderBagging [23]), but is not the focus of our paper.

D. Synthetic datasets

We now use two synthetic datasets to analysis the bias introduced by undersampling and understand how it affects the posterior probability. Given the simulated setting we are able to control the true posterior probability p and measure the sampling bias embedded in p_s. We see that the bias is larger when the two classes are overlapping and that stronger undersampling induces a larger bias.

Let us consider two binary classification tasks, wherein positive and negative observations are drawn randomly from two distinct normal distributions. For both datasets we set the number of positives to be 10% of 10,000 observations, with $\omega^- \sim N(0, \sigma)$ and $\omega^+ \sim N(\mu, \sigma)$, where $\mu > 0$. The distance between the two normal distributions, μ, is used to control the degree of separation between the classes. When μ is large, the two classes are well-separated, while for small μ they strongly overlap. In the first dataset, we simulate a classification problem with a very low degree of separation (using $\mu = 3$), in the second a task with well-separated classes using $\mu = 15$ (see Figure 2). The first simulates a difficult classification task, the latter an easy one. For both dataset we set $\sigma = 3$.

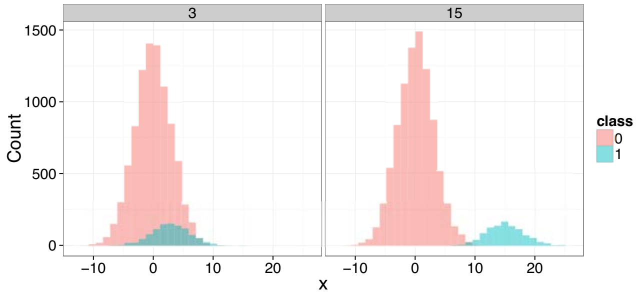

Fig. 2. Synthetic datasets with positive and negative observations sampled from two different normal distributions. Positives account for 10% of the 10,000 random values. On the left we have a difficult problem with overlapping classes ($\mu = 3$), on the right an easy problem where the classes are well-separated ($\mu = 15$).

Figure 3 shows how p_s changes with β (p corresponds to $\beta = 1$). When $\beta \rightarrow \frac{N^+}{N^-}$ the probability shifts to the left, allowing for higher probabilities on the right hand side of the chart (where positive observations are located). In other words, removing negative samples with undersampling increases the positive posterior probability, moving the classification boundary so that more samples are classified as positive. The stronger the undersampling, the larger the shift, i.e. the drift of p_s from p. The drift is larger in the dataset with non-separable classes confirming the results of Section III-B.

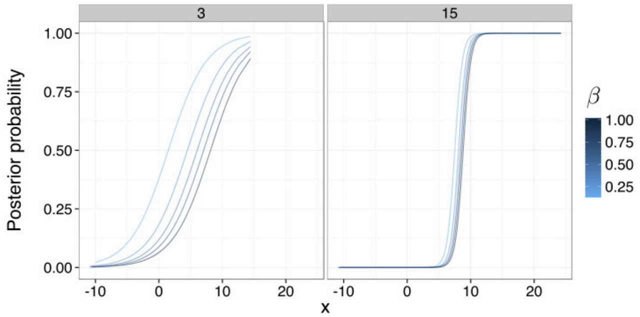

Fig. 3. Posterior probability as a function of β. On the left the task with $\mu = 3$ and on the right the one with $\mu = 15$. Note that p corresponds to $\beta = 1$ and p_s to $\beta < 1$.

Figure 4 displays p_s, p' and p for $\beta = \frac{N^+}{N^-}$ in the dataset with overlapping classes ($\mu = 3$) and we see that p' closely approximates p. As $p' \approx p$, we can say that the above transformation based on (9) is able to correct the probability drift that occurs with undersampling. The correction seems particularly effective on the left-hand side (where the majority class is located), while is less precise on the right-hand side where we expect to have larger variance on p' due to the small number of positive samples.

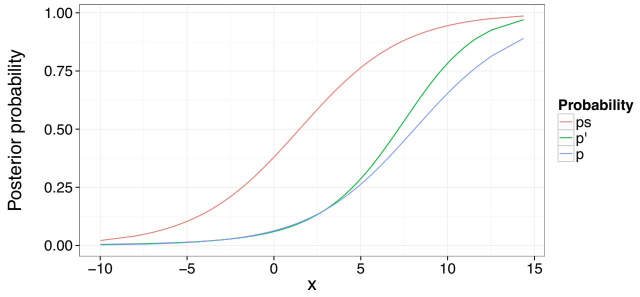

Fig. 4. Posterior probabilities p_s, p' and p for $\beta = \frac{N^+}{N^-}$ in the dataset with overlapping classes ($\mu = 3$).

IV. Classification Threshold with Unbiased Probabilities

In the previous section we showed how undersampling induces biased posterior probabilities and presented a method to correct for this bias. We now want to investigate how to use them for classification.

A. Threshold with Bayes Minimum Risk

Standard decision making process based on Bayes decision theory developed in most textbooks on pattern recognition or machine learning (see for example [s4], [3], [1s]) defines the optimal class of a sample as the one minimizing the risk (expected value of the loss function). In a binary classification problem, the risk of the positive and negative class is defined as follows:

$$r^+ = (1-p)l_{1,0} + pl_{1,1}$$

$$r^- = (1-p)l_{0,0} + pl_{0,1}$$

where $p = p(+|x)$ and $l_{i,j}$ is the loss (cost) incurred in deciding i when the true class is j.

TABLE II. LOSS k ATRIX

	Actual I ositive	Actual Negative
I redicted I ositive	$l_{1,1}$	$l_{1,0}$
I redicted Negative	$l_{0,1}$	$l_{0,0}$

Bayes decision rule for minimizing the risk can be stated as follows: assign the positive class to samples with $r^+ \leq r^-$, and the negative otherwise. This is equivalent to predict a sample as positive when $p > \tau$ and the threshold τ is:

$$\tau = \frac{l_{1,0} - l_{0,0}}{l_{1,0} - l_{0,0} + l_{0,1} - l_{1,1}}$$

Typically the cost of a correct prediction is zero, hence $l_{0,0} = 0$ and $l_{1,1} = 0$. In an unbalanced problem, the cost of missing a positive instance (false negative) is usually higher than the cost of missing a negative (false positive). When the costs of a false negative and false positive are unknown, a natural solution is to set the costs using the priors. Let $l_{1,0} = \pi^+$ and $l_{0,1} = \pi^-$, where $\pi^+ = p(+)$ and $\pi^- = p(-)$. Then, since $\pi^- > \pi^+$ we have $l_{0,1} > l_{1,0}$ as desired. We can then write:

$$\tau = \frac{l_{1,0}}{l_{1,0} + l_{0,1}} = \frac{\pi^+}{\pi^+ + \pi^-} = \pi^+ \tag{10}$$

since $\pi^+ + \pi^- = 1$. This is also the optimal threshold in a cost-sensitive application where the costs are defined using the priors [13].

B. Classification threshold adjustment

Even if undersampling produces biased probability estimates, it is often used to balance datasets with skewed class distributions because several classifiers have empirically shown better performance when trained on balanced dataset [s5], [14]. Let τ_s denote the threshold used to classify an observation after undersampling, form (10) we have $\tau_s = \pi_s^+$, where π_s^+ is the positive class prior after undersampling. In the case of undersampling with $\beta = \frac{N^+}{N^-}$ (balanced training set) we have $\tau_s = 0.5$.

When correcting p_s with (9), we must also correct the probability threshold to maintain the predictive accuracy defined by τ_s (this is needed otherwise we would use different misclassification costs for p'). Let τ' be the threshold for the unbiased probability p'. From Elkan [13]:

$$\frac{\tau'}{1-\tau'} \frac{1-\tau_s}{\tau_s} = \beta \tag{11}$$

$$\tau' = \frac{\beta \tau_s}{(\beta - 1)\tau_s + 1} \tag{1s}$$

Using $\tau_s = \pi_s^+$, (1s) becomes:

$$\tau' = \frac{\beta \pi_s^+}{(\beta - 1)\pi_s^+ + 1}$$

$$\tau' = \frac{\beta \frac{N^+}{N^+ + \beta N^-}}{(\beta - 1)\frac{N^+}{N^+ + \beta N^-} + 1} = \frac{N^+}{N^+ + N^-} = \pi^+$$

The optimal threshold to use with p' is equal to the one for p. As an alternative to classifying observations using p_s with τ_s, we can obtain equivalent results using p' with τ'. In summary, as a result of undersampling, a higher number of observations are predicted as positive, but the posterior probabilities are biased due to a change in the priors. Equation (1s) allows us find the threshold that guarantees equal accuracy after the posterior probability correction. Therefore, in order to classify observations with unbiased probabilities after undersampling, we have to first obtain p' from p_s with (9) and then use τ' as a classification threshold.

V. k EASURES OF CLASSIFICATION ACCURACY AND I ROBABILITY CALIBRATION

The choice of balancing the training set or leaving it unbalanced has a direct influence on the classification model that is learnt. A model learnt on a balanced training set has the two classes equally represented. In the case of an unbalanced training set, the model learns from a dataset skewed towards one class. Hence, the classification model learnt after undersampling is different from the one learnt on the original dataset. In this section we compare the probability estimates of two models, one learnt in the presence and the other in the absence of undersampling. The probabilities are evaluated in terms of ranking produced, classification accuracy and calibration.

To asses the impact of undersampling, we first use accuracy measures based on the confusion matrix (Table III).

TABLE III. CONFUSION k ATRIX

	Actual I ositive	Actual Negative
I redicted I ositive	TI	FI
I redicted Negative	FN	TN

In an unbalanced class problem, it is well-known that quantities like TI R ($\frac{TP}{TP+FN}$), TNR ($\frac{TN}{FP+TN}$) and average accuracy ($\frac{TP+TN}{TP+FN+FP+TN}$) are misleading assessment measures [10]. Let us define I recision = $\frac{TP}{TP+FP}$ and Recall = $\frac{TP}{TP+FN}$. Typically we want to have high confidence that observations predicted as positive are actually positive (high I recision) as well as a high detection rate of the positives (high Recall). However, I recision and Recall share an inverse relationship, whereby high I recision comes at the cost of low Recall and vice versa. An accuracy measure based on both I recision and Recall is the F-measure, also known as F1-score or F-score. F-measure ($2\frac{Precision \times Recall}{Precision + Recall}$) and G-mean ($\sqrt{TPR \times TNR}$) are often considered to be useful and effective performance measures for unbalanced datasets.

An alternative way to measure the quality of a probability estimate is to look at the ranking produced by the probability. A good probability estimate should rank first all the minority class observations and then those from the majority class.

In other words, if $\hat{p}$ is a good estimate of $p(+|x)$, then $\hat{p}$ should give high probability to the positive examples and small probability to the negatives. A well-accepted ranking measure for unbalanced dataset is AUC (Area Under the ROC curve) [5]. To avoid the problem of different misclassification costs, we use an estimation of AUC based on the Mann-Whitney statistic [10]. This estimate measures the probability that a random minority class example ranks higher than a random majority class example [15].

In order to measure the probability calibration, we used the Brier Score (BS) [4]. BS is a measure of average squared loss between the estimated probabilities and the actual class value. It allows to evaluate how well the probabilities are calibrated, the lower the BS the more accurate are the probabilistic predictions of a model. Let $\hat{p}(y_i|x_i)$ be the probability estimate of sample x_i to have class $y_i \in \{1, 0\}$, BS is defined as:

$$BS = \frac{1}{N} \sum_{i=1}^{N} \{y_i - \hat{p}(y_i|x_i)\}^2 \qquad (13)$$

VI. Experimental results

In the previous sections we used synthetic datasets to study the effect of undersampling. We now consider real-world unbalanced datasets from the UCI repository used in [9]. For each dataset we adopt a 10-fold cross validation (CV) to test our models and we repeated the CV 10 times. In particular, we used a stratified CV, where the class proportion in the datasets is kept the same over all the folds. As the original datasets are unbalanced, the resulting folds are unbalanced as well. For each fold of CV we learn two models: one using all the observations and the other with the ones remaining after undersampling. Then both models are tested on the same testing set. We used several supervised classification algorithms available in R [20]: Random Forest [18], SVM [17], and Logit Boost [22].

We denote as $\hat{p}_s$ and $\hat{p}$ the posterior probability estimates obtained with and without undersampling and as $\hat{p}'$ the bias-corrected probability obtained from $\hat{p}_s$ with equation (9). Let τ, τ_s and τ' be the probability thresholds used for $\hat{p}$, $\hat{p}_s$ and $\hat{p}'$ respectively, where $\tau = \pi^+$, $\tau_s = \pi^+_s$ and $\tau' = \pi^+$. The goal of these experiments is to compare which probability estimates return the highest ranking (AUC), calibration (BS) and classification accuracy (G-mean) when coupled with the thresholds defined before. In undersampling, the amount of sampling defined by β is usually set to be equal to $\frac{N^+}{N^-}$, leading to a balanced dataset where $\pi^+_s = \pi^-_s = 0.5$. However, there is no reason to believe that this is the optimal sampling rate. Often, the optimal rate can be found only a posteriori after trying different values of β [9]. For this reason we replicate the CV with different β such that $\{\frac{N_+}{N_-} \leq \beta \leq 1\}$ and for each CV the accuracy is computed as the average G-mean (or AUC) over all the folds.

In table V we report the results over all the datasets. For each dataset, we rank the probability estimates $\hat{p}_s$, $\hat{p}$ and $\hat{p}'$ from the worst to the best performing for different values of β. We then sum the ranks over all the values of β and over all datasets. More formally, let $R_{i,k,b} \in \{1, 2, 3\}$ be the rank of probability i on dataset k when $\beta = b$. The probability with the highest accuracy in k when $\beta = b$ has $R_{i,k,b} = 3$ and the one with the lowest has $R_{i,k,b} = 1$. Then the sum of ranks for the probability i is defined as $\sum_k \sum_b R_{i,k,b}$. The higher the sum, the higher the number of times that one probability has higher accuracy than the others.

TABLE IV. Datasets from the UCI repository used in [9].

Datasets	N	N^+	N^-	N^+/N
ecoli	336	35	301	0.10
glass	214	17	197	0.08
letter-a	20000	789	19211	0.04
letter-vowel	20000	3878	16122	0.19
ism	11180	260	10920	0.02
letter	20000	789	19211	0.04
oil	937	41	896	0.04
page	5473	560	4913	0.10
pendigits	10992	1142	9850	0.10
PhosS	11411	613	10798	0.05
satimage	6430	625	5805	0.10
segment	2310	330	1980	0.14
boundary	3505	123	3382	0.04
estate	5322	636	4686	0.12
cam	18916	942	17974	0.05
compustat	13657	520	13137	0.04
covtype	38500	2747	35753	0.07

For AUC, a higher rank sum means a higher AUC and hence a better ranking returned by the probability. Similarly, with G-mean, a higher rank sum corresponds to higher predictive accuracy. However, in the case of BS, a higher rank sum means poorer probability calibration (larger bias). Table V has in bold the probabilities with the best rank sum according to the different metrics. For each metric and classifier it reports the p-values of the paired t-test based on the ranks between $\hat{p}$ and $\hat{p}'$ and between $\hat{p}$ and $\hat{p}_s$.

In terms of AUC, we see that $\hat{p}_s$ and $\hat{p}'$ have better performances than $\hat{p}$ for LB and SVM. The rank sum is the same for $\hat{p}_s$ and $\hat{p}'$ since the two probabilities are linked by a monotone transformation (equation (9)). If we look at G-mean, $\hat{p}_s$ and $\hat{p}'$ return better accuracy than $\hat{p}$ two times out of three. In this case, the rank sums of $\hat{p}_s$ and $\hat{p}'$ are the same since we used τ_s and τ' as the classification threshold, where τ' is obtained from τ_s using (12). If we look at the p-values, we can strongly reject the null hypothesis that the accuracy of $\hat{p}_s$ and $\hat{p}$ are from the same distribution. For all classifiers, $\hat{p}$ is the probability estimate with the best calibration (lower rank sum with BS), followed by $\hat{p}'$ and $\hat{p}_s$. The rank sum of $\hat{p}'$ is always lower than the one of $\hat{p}_s$, indicating that $\hat{p}'$ has lower bias than $\hat{p}_s$. This result confirms our theory that equation (9) allows one to reduce the bias introduced by undersampling.

In summary from this experiment we can conclude that undersampling does not always improve the ranking or classification accuracy of an algorithm, but when it is the case we should use $\hat{p}'$ instead of $\hat{p}_s$ because the first has always better calibration.

We now consider a real-world dataset, composed of credit card transactions from September 2013 made available by our industrial partner. [1] It contains a subset of online transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, where the positive class (frauds) account for 0.172% of all transactions, and the minimum value of β is ≈ 0.00173. In Figure 5 we have the AUC for different values of β.

[1] The dataset is available at http://www.ulb.ac.be/di/map/adalpozz/data/creditcard.Rdata

TABLE V. SUM OF RANKS AND P-VALUES OF THE PAIRED T-TEST BETWEEN THE RANKS OF $\hat{p}$ AND $\hat{p}'$ AND BETWEEN $\hat{p}$ AND $\hat{p}_s$ FOR DIFFERENT METRICS. IN **BOLD** THE PROBABILITIES WITH THE BEST RANK SUM (HIGHER FOR AUC AND G-MEAN, LOWER FOR BS).

Metric	Algo	$\sum R_{\hat{p}}$	$\sum R_{\hat{p}_s}$	$\sum R_{\hat{p}'}$	$\rho(R_{\hat{p}}, R_{\hat{p}_s})$	$\rho(R_{\hat{p}}, R_{\hat{p}'})$
AUC	LB	22,516	**23,572**	**23,572**	0.322	0.322
AUC	RF	**24,422**	22,619	22,619	0.168	0.168
AUC	SVM	19,595	**19,902.5**	**19,902.5**	0.873	0.873
G-mean	LB	**23,281**	23,189.5	23,189.5	0.944	0.944
G-mean	RF	22,986	**23,337**	**23,337**	0.770	0.770
G-mean	SVM	19,550	**19,925**	**19,925**	0.794	0.794
BS	LB	**19809.5**	29448.5	20402	0.000	0.510
BS	RF	**18336**	28747	22577	0.000	0.062
BS	SVM	**17139**	23161	19100	0.001	0.156

The boxplots of $\hat{p}_s$ and $\hat{p}'$ are identical because of (9), they increase with $\beta \rightarrow \frac{N^+}{N^-}$ and have higher median than the one of $\hat{p}$. This example shows how in case of extreme class imbalance, undersampling can improve predictive accuracy of several classification algorithms.

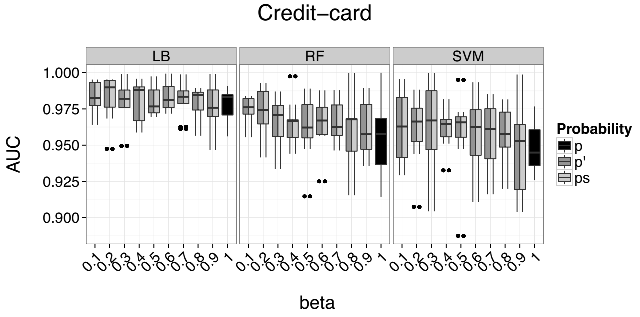

Fig. 5. Boxplot of AUC for different values of β in the *Credit-card* dataset.

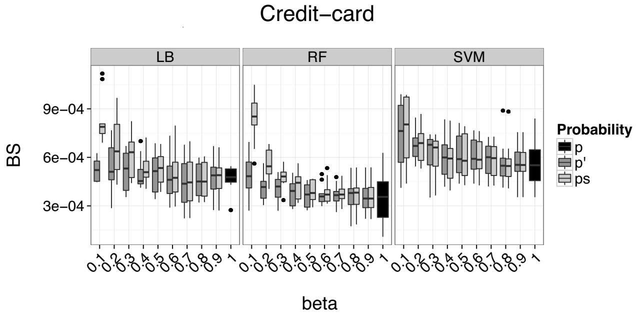

Fig. 6. Boxplot of BS for different values of β in the Credit-card dataset.

In Figure 6 we have the BS for different values of β. The boxplots of $\hat{p}'$ show in general smaller calibration error (lower BS) than those of $\hat{p}_s$ and the latter have higher BS especially for small values of β. This supports our previous results, which found that the loss in probability calibration for $\hat{p}_s$ is greater the stronger the undersampling.

VII. CONCLUSION

In this paper, we study the bias introduced in the posterior probabilities that occurs as an artifact of undersampling. We use several synthetic datasets to analyze this problem from a theoretical perspective, and then ground our findings with an empirical evaluation over several real-world datasets.

The first result of the paper is that the bias due to the instance selection procedure in undersampling is essentially equivalent to the bias the occurs with a change in the priors when class-within distributions remain stable. With undersampling, we create a different training set, where the classes are less unbalanced. However, if we make the assumption that the training and testing sets come from the same distribution, it follows that the probability estimates obtained after undersampling are biased. As a result of undersampling, the posterior probability $\hat{p}_s$ is shifted away from the true distribution, and the optimal separation boundary moves towards the majority class so that more cases are classified into the minority class.

By making the assumptions that prior probabilities do not change from training and testing, i.e. they both come form the same data generating process, we propose the transformation given in (9), which allows us to remove the drift in $\hat{p}_s$ due to undersampling. The bias on $\hat{p}_s$ registered by BS gets larger for small values of β, which means stronger undersampling produces probabilities with poorer calibration (larger loss). With synthetic, UCI and *Credit-card* datasets, the drift-corrected probability ($\hat{p}'$) has significantly better calibration than $\hat{p}_s$ (lower Brier Score).

Even if undersampling produces poorly calibrated probability estimates $\hat{p}_s$, several studies have shown that it often provides better predictive accuracy than $\hat{p}$ [25], [14]. To improve the calibration of $\hat{p}_s$ we propose to use $\hat{p}'$ since this transformation does not affect the ranking. In order to maintain the accuracy obtained with $\hat{p}_s$ and the probability threshold τ_s, we proposed to use $\hat{p}'$ together with τ' to account for the change in priors. By changing the undersampling rate β we give different costs to false positives and false negatives, combining $\hat{p}'$ with τ' allows one to maintain the same misclassification costs of a classification strategy with $\hat{p}_u$ and τ_u for any value of β.

Finally, we considered a highly unbalanced dataset (*Credit-card*), where the minority class accounts for only 0.172% of all observations. In this dataset, the large improvement in accuracy obtained with undersampling was coupled with poor calibrated probabilities (large BS). By correcting the posterior probability and changing the threshold we were able to improve calibration without losing predictive accuracy. Obtaining well-calibrated classifiers is particularly important in decision systems based on fraud detection. This is one of the rare papers making available the fraud detection dataset used for testing.

ACKNOWLEDGMENTS

A. Dal Pozzolo is supported by the Doctiris scholarship funded by Innoviris, Brussels, Belgium. G. Bontempi is supported by the *BridgeIRIS* and *BruFence* projects funded by Innoviris, Brussels, Belgium.

APPENDIX

Let $p_t = p2y_t = +|x_t)$ be the posterior probability for a testing instance $2x_t, y_t)$, where the testing set has priors: $\pi_t^- = \frac{N_t^-}{N_t}$ and $\pi_t^+ = \frac{N_t^+}{N_t}$. In the unbalanced training set we have $\pi^- = \frac{N^-}{N}$, $\pi^+ = \frac{N^+}{N}$ and $p = p2+|x)$. After undersampling the training set $\pi_s^- = \frac{\beta N^-}{N^+ + \beta N^-}$, $\pi_s^+ = \frac{N^+}{N^+ + \beta N^-}$ and $p_s = p2+|x, s = 1)$. If we assume that the class conditional distributions $p(x|+)$ and $p(x|-)$ remain the same between the training and testing sets, Saerens et al. [21] show that, given different priors between the training and testing sets, the posterior probability can be corrected with the following equation:

$$p_t = \frac{\frac{\pi_t^+}{\pi_s^+} p_s}{\frac{\pi_t^+}{\pi_s^+} p_s + \frac{\pi_t^-}{\pi_s^-}(1 - p_s)} \tag{14}$$

Let us assume that the training and testing sets share the same priors: $\pi_t^+ = \pi^+$ and $\pi_t^- = \pi^-$:

$$p_t = \frac{\frac{\pi^+}{\pi_s^+} p_s}{\frac{\pi^+}{\pi_s^+} p_s + \frac{\pi^-}{\pi_s^-}(1 - p_s)}$$

Then, since

$$\frac{\pi^+}{\pi_s^+} = \frac{\frac{N^+}{N^+ + N^-}}{\frac{N^+}{N^+ + \beta N^-}} = \frac{N^+ + \beta N^-}{N^+ + N^-} \tag{15}$$

$$\frac{\pi^-}{\pi_s^-} = \frac{\frac{N^-}{N^+ + N^-}}{\frac{\beta N^-}{N^+ + \beta N^-}} = \frac{N^+ + \beta N^-}{\beta(N^+ + N^-)} \tag{16}$$

we can write

$$p_t = \frac{\frac{N^+ + \beta N^-}{N^+ + N^-} p_s}{\frac{N^+ + \beta N^-}{N^+ + N^-} p_s + \frac{N^+ + \beta N^-}{\beta(N^+ + N^-)}(1 - p_s)} = \frac{\beta p_s}{\beta p_s - p_s + 1}$$

The transformation proposed by Saerens et al. [21] is equivalent to equation (4) and the one developed independently by Elkan [13] for cost-sensitive learning:

$$p_t = \pi_t^+ \frac{p_s - \pi_s^+ p_s}{\pi_s^+ - \pi_s^+ p_s + \pi_t^+ p_s - \pi_t^+ \pi_s^+} \tag{17}$$

$$p_t = \frac{(1 - \pi_s^+) p_s}{\frac{\pi_s^+}{\pi_t^+}(1 - p_s) + p_s - \pi_s^+}$$

using (15), $\pi_t^+ = \pi^+$ and $\pi_t^- = \pi^-$:

$$p_t = \frac{\frac{\beta N^-}{N^+ + \beta N^-} p_s}{\frac{N^+ + N^-}{N^+ + \beta N^-}(1 - p_s) + p_s - \frac{N^+}{N^+ + \beta N^-}} = \frac{\beta p_s}{\beta p_s - p_s + 1}$$

References

[1] Rehan Akbani, Stephen Kwek, and Nathalie Japkowicz. Applying support vector machines to imbalanced datasets. In *Machine Learning: ECML 2004*, pages 39–50. Springer, 2004.

[2] Urvesh Bhowan, Michael Johnston, Mengjie Zhang, and Xin Yao. Evolving diverse ensembles using genetic programming for classification with unbalanced data. *Evolutionary Computation, IEEE Transactions on*, 17(3):368–386, 2013.

[3] Christopher M Bishop et al. *Pattern recognition and machine learning*, volume 4. springer New York, 2006.

[4] Glenn W Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

[5] Nitesh V Chawla. Data mining for imbalanced datasets: An overview. In *Data mining and knowledge discovery handbook*, pages 853–867. Springer, 2005.

[6] Nitesh V Chawla, Nathalie Japkowicz, and Aleksander Kotcz. Editorial: special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter*, 6(1):1–6, 2004.

[7] NV Chawla, KW Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research (JAIR)*, 16:321–357, 2002.

[8] Andrea Dal Pozzolo, Giacomo Boracchi, Olivier Caelen, Cesare Alippi, and Gianluca Bontempi. Credit card fraud detection and concept-drift adaptation with delayed supervised information. In *Neural Networks (IJCNN), 2015 International Joint Conference on*. IEEE, 2015.

[9] Andrea Dal Pozzolo, Olivier Caelen, and Gianluca Bontempi. When is undersampling effective in unbalanced classification tasks? In *Machine Learning and Knowledge Discovery in Databases*. Springer, 2015.

[10] Andrea Dal Pozzolo, Olivier Caelen, Yann-Ael Le Borgne, Serge Waterschoot, and Gianluca Bontempi. Learned lessons in credit card fraud detection from a practitioner perspective. *Expert Systems with Applications*, 41(10):4915–4928, 2014.

[11] C. Drummond and R.C. Holte. C4.5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In *Workshop on Learning from Imbalanced Datasets II*. Citeseer, 2003.

[12] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2012.

[13] C. Elkan. The foundations of cost-sensitive learning. In *International Joint Conference on Artificial Intelligence*, volume 17, pages 973–978. Citeseer, 2001.

[14] Andrew Estabrooks, Taeho Jo, and Nathalie Japkowicz. A multiple resampling method for learning from imbalanced data sets. *Computational Intelligence*, 20(1):18–36, 2004.

[15] David J Hand and Robert J Till. A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine Learning*, 45(2):171–186, 2001.

[16] N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002.

[17] Alexandros Karatzoglou, Alex Smola, Kurt Hornik, and Achim Zeileis. kernlab-an s4 package for kernel methods in r. 2004.

[18] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002.

[19] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. The MIT Press, 2009.

[20] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. ISBN 3-900051-07-0.

[21] Marco Saerens, Patrice Latinne, and Christine Decaestecker. Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure. *Neural computation*, 14(1):21–41, 2002.

[22] Jarek Tuszynski. *caTools: Tools: moving window statistics, GIF, Base64, ROC AUC, etc.*, 2013. R package version 1.16.

[23] Shuo Wang, Ke Tang, and Xin Yao. Diversity exploration and negative correlation learning on imbalanced data sets. In *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*, pages 3259–3266. IEEE, 2009.

[24] Andrew R Webb. *Statistical pattern recognition*. John Wiley & Sons, 2003.

[25] Gary M Weiss and Foster Provost. The effect of class distribution on classifier learning: an empirical study. *Rutgers Univ*, 2001.