Calibrating Probability with Undersampling for Unbalanced Classification

Andrea Dal Pozzolo, Olivier Caelen,
Reid A. Johnson, and Gianluca Bontempi

8/12/2015

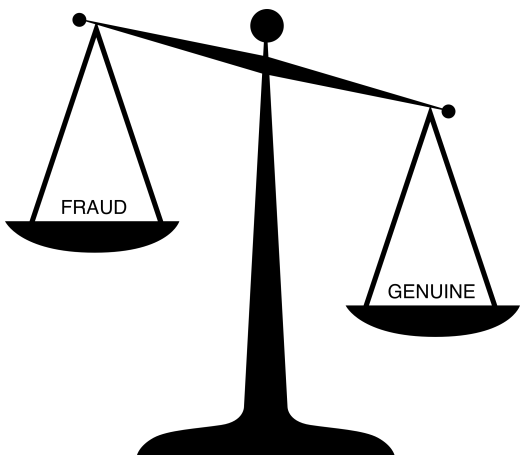
IEEE CIDM 2015
Cape Town, South Africa

worldline
e-payment services

ULB

Introduction

- In several binary classification problems, the two classes are not equally represented in the dataset.
- In Fraud detection for example, fraudulent transactions are rare compared to genuine ones (less than 1% [2]).
- Many classification algorithms performs poorly in with unbalanced class distribution [5].
- A standard solution to unbalanced classification is rebalancing the classes before training a classifier.



UNDERSAMPLING

- Undersampling is a well-know technique used to balanced a dataset.
- It consists in down-sizing the majority class by removing observations at random until the dataset is balanced.
- Several studies [11, 4] have reported that it improves classification performances.
- Most often, the consequences of undersampling on the posterior probability of a classifier are ignored.

Objective of this study

In this work we:

- formalize how undersampling works.
- show that undersampling is responsible for a shift in the posterior probability of a classifier.
- study how this shift is linked to class separability.
- investigate how this shift produces biased probability estimates.
- show how to obtain and use unbiased (calibrated) probability for classification.

The problem

- Let us consider a binary classification task $f : \mathbb{R}^n \to \{+, -\}$
- $\mathbf{X} \in \mathbb{R}^n$ is the input and $\mathbf{Y} \in \{+, -\}$ the output domain.
- $+$ is the minority and $-$ the majority class.
- Given a classifier $\mathcal{K}$ and a training set T_N, we are interested in estimating for a new sample (x, y) the posterior probability $p(y = +|x)$.

Effect of undersampling

- Suppose that a classifier $\mathcal{K}$ is trained on set T_N which is unbalanced.
- Let $\mathbf{s}$ be a random variable associated to each sample $(x, y) \in \mathrm{T}_N$, $\mathbf{s} = 1$ if the point is sampled and $\mathbf{s} = 0$ otherwise.
- Assume that $\mathbf{s}$ is independent of the input x given the class y (*class-dependent selection*):

$$p(\mathbf{s}|y, x) = p(\mathbf{s}|y) \Leftrightarrow p(x|y, \mathbf{s}) = p(x|y)$$

.

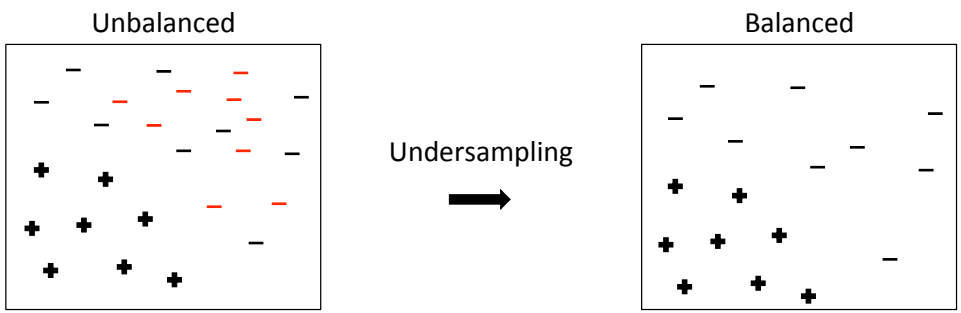


Figure : Undersampling: remove randomly majority class examples. In red samples that are removed from the unbalanced dataset ($\mathbf{s} = 0$).

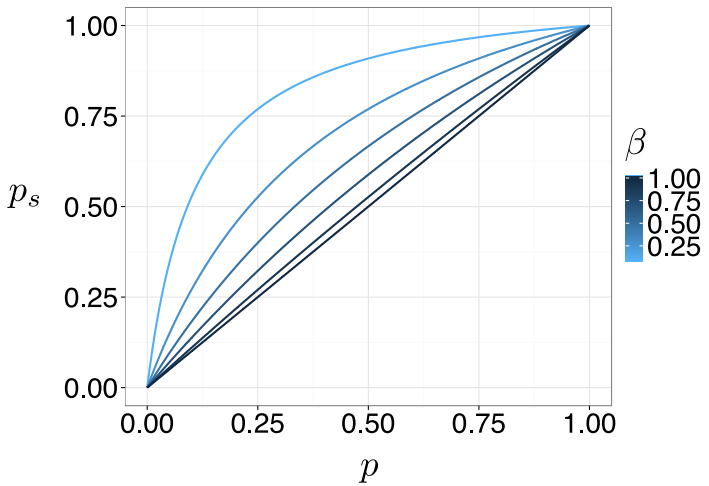
Posterior probabilities

Let $p_s = p(+|x, s = 1)$ and $p = p(+|x)$. We can write p_s as a function of p [1]:

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

where $\beta = p(s = 1|-)$. Using (1) we can obtain an expression of p as a function of p_s:

$$p = \frac{\beta p_s}{\beta p_s - p_s + 1} \tag{2}$$



Warping and class separability

Let ω^+ and ω^- denote $p(x|+)$ and $p(x|-)$, and π^+ (π_s^+) the class priors before (after) undersampling. Using Bayes' theorem:

$$p = \frac{\omega^+ \pi^+}{\omega^+ - \delta\pi^-} \tag{3}$$

where $\delta = \omega^+ - \omega^-$. Similarly, since ω^+ does not change with undersampling:

$$p_s = \frac{\omega^+ \pi_s^+}{\omega^+ - \delta\pi_s^-} \tag{4}$$

Now we can write $p_s - p$ as:

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

Warping and class separability

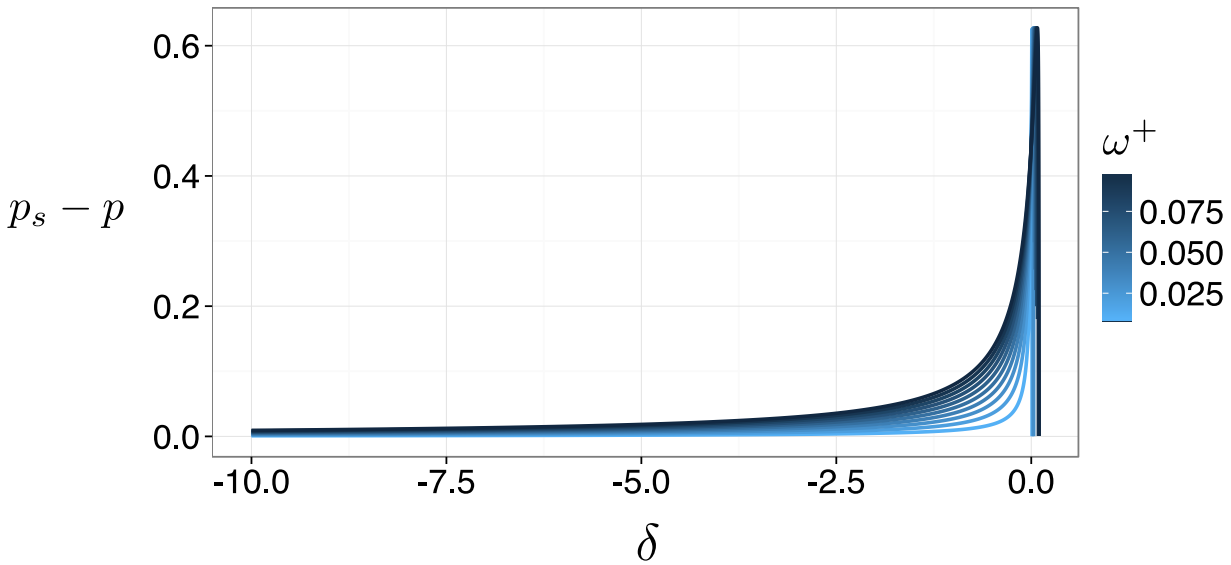
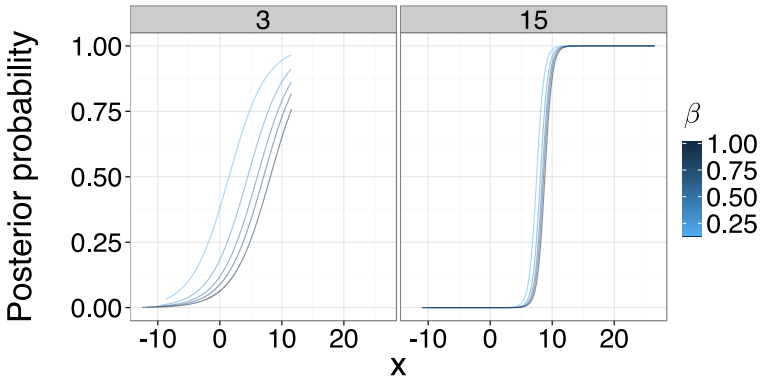
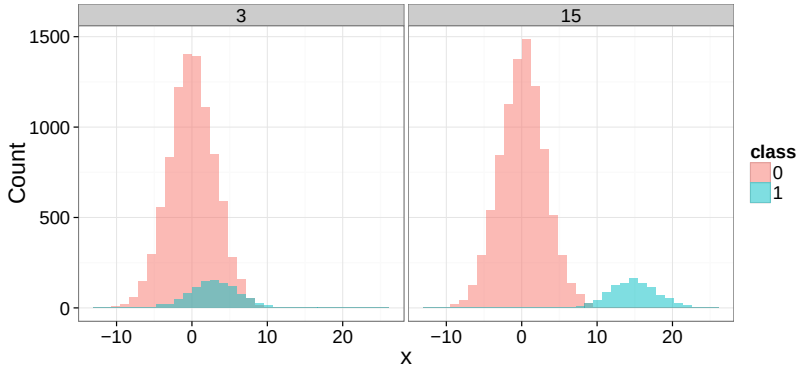


Figure : $p_s - p$ as a function of δ, where $\delta = \omega^+ - \omega^-$ for values of $\omega^+ \in \{0.01, 0.1\}$ when $\pi_s^+ = 0.5$ and $\pi^+ = 0.1$.

Warping and class separability II



(a) p_s as a function of β



(b) Class distribution

Figure : Class distribution and posterior probability as a function of β for two univariate binary classification tasks with norm class conditional densities $\mathcal{X}_- \sim \mathcal{N}(0, \sigma)$ and $\mathcal{X}_+ \sim \mathcal{N}(\mu, \sigma)$ (on the left $\mu = 3$ and on the right $\mu = 15$, in both examples $\sigma = 3$). Note that p corresponds to $\beta = 1$ and p_s to $\beta < 1$.

ADJUSTING POSTERIOR PROBABILITIES

We propose to use correct p_s with p', which is obtained using (2):

$$p' = \frac{\beta p_s}{\beta p_s - p_s + 1} \tag{6}$$

Eq. (6) is a special case of the framework proposed by Saerens et al. [8] and Elkan [3] (see Appendix in the paper).

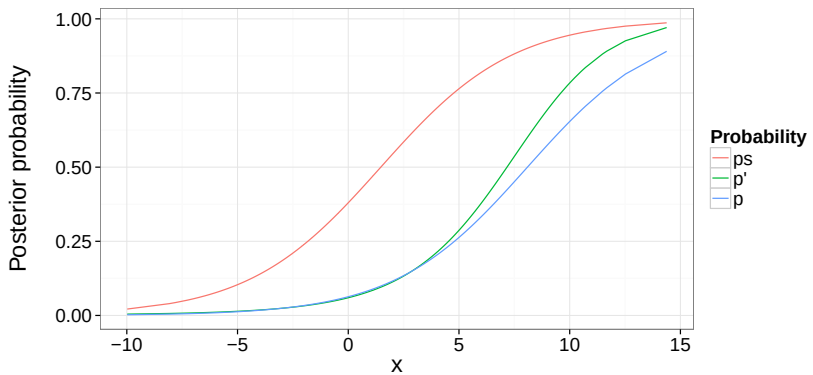


Figure : Posterior probabilities p_s, p' and p for $\beta = \frac{N^+}{N^-}$ in the dataset with overlapping classes ($\mu = 3$).

Classification threshold

Let r^+ and r^- be the risk of predicting an instance as positive and negative:

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

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

where $l_{i,j}$ is the cost in predicting i when the true class is j and $p = p(y = +|x)$. A sample is predicted as positive if $r^+ \leq r^-$ [10]:

$$\hat{y} = \begin{cases} + & \text{if } r^+ \leq r^- \\ - & \text{if } r^+ > r^- \end{cases} \tag{7}$$

Alternatively, predict as positive when $p > \tau$ with τ:

$$\tau = \frac{l_{1,0} - l_{0,0}}{l_{1,0} - l_{0,0} + l_{0,1} - l_{1,1}} \tag{8}$$

Correcting the classification threshold

When the costs of a FN ($l_{0,1}$) and FP ($l_{1,0}$) are unknown, we can use the priors. Let $l_{1,0} = \pi^+$ and $l_{0,1} = \pi^-$, from (8) we get:

$$\tau = \frac{l_{1,0}}{l_{1,0} + l_{0,1}} = \frac{\pi^+}{\pi^+ + \pi^-} = \pi^+ \tag{9}$$

since $\pi^+ + \pi^- = 1$. Then we should use π^+ as threshold with p:

$$p \longrightarrow \tau = \pi^+$$

Similarly

$$p_s \longrightarrow \tau_s = \pi_s^+$$

From Elkan [3]:

$$\frac{\tau'}{1-\tau'} \frac{1-\tau_s}{\tau_s} = \beta \tag{10}$$

Therefore, we obtain:

$$p' \longrightarrow \tau' = \pi^+$$

Experimental settings

- We denote as $\hat{p}_s$, $\hat{p}$ and $\hat{p}'$ the estimates of p_s, p and p'
- Goal: understand which probability return the highest ranking (AUC), calibration (BS) and classification accuracy (G-mean).
- We use a 10-fold cross validation (CV) to test our models and we repeated the CV 10 times.
- We test several classification algorithms: Random Forest [7], SVM [6], and Logit Boost [9].
- We consider real-world unbalanced datasets from the UCI repository used in [1].

Learning Framework

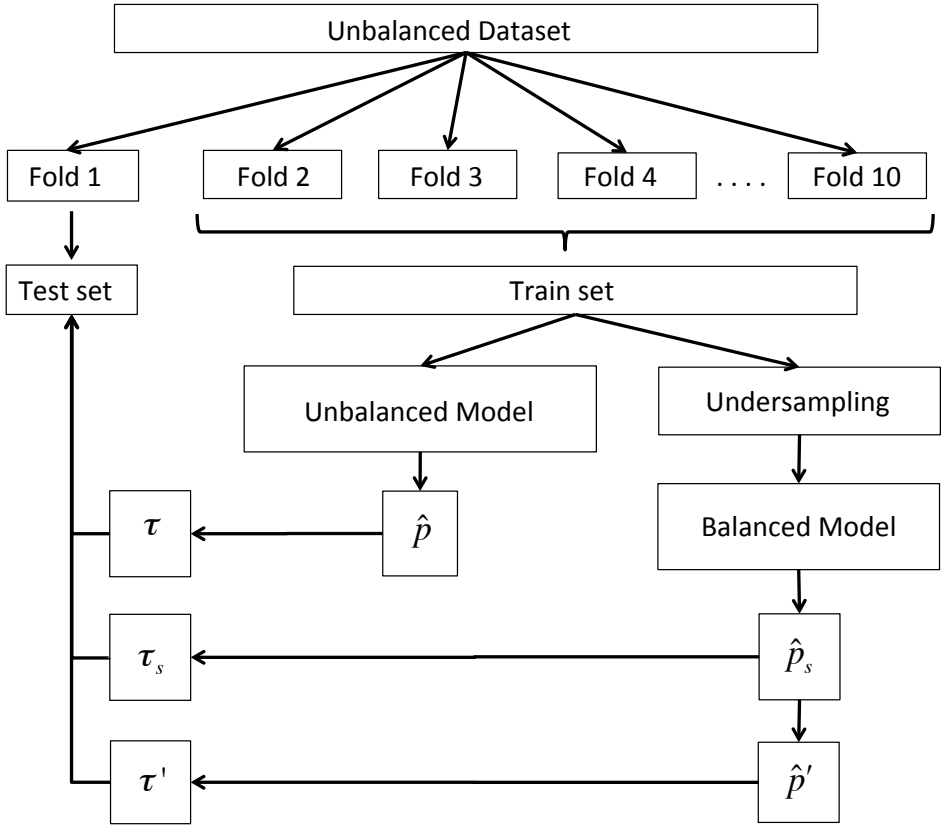


Figure : Learning framework for comparing models with and without undersampling using Cross Validation (CV). We use one fold of the CV as testing set and the others for training, and iterate the framework to use all the folds once for testing.

DATASETS

Table : Datasets from the UCI repository used in [1].

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

RESULTS

Table : 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).

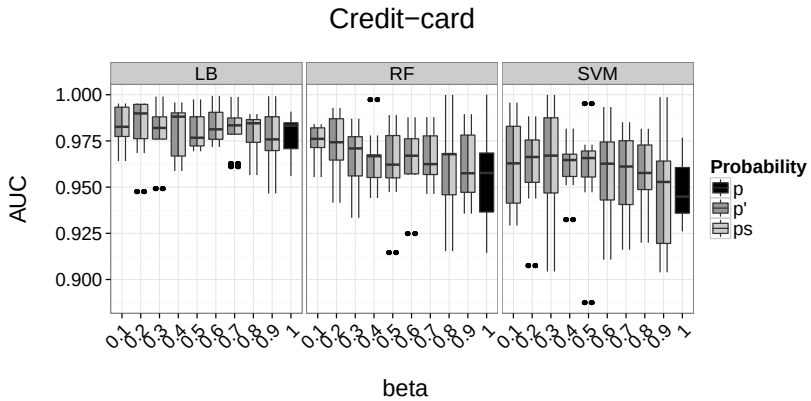
Metric	Algo	$\sum R_{\hat{p}}$	$\sum R_{\hat{p}_s}$	$\sum R_{\hat{p}'}$	$\rho(R_{\hat{p}}, R_{\hat{p}_s})$	$\rho(R_{\hat{p}}, R_{\hat{p}'})$
AUC	LB	22,516	**23,572**	**23,572**	0.322	0.322
AUC	RF	**24,422**	22,619	22,619	0.168	0.168
AUC	SVM	19,595	**19,902.5**	**19,902.5**	0.873	0.873
G-mean	LB	**23,281**	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

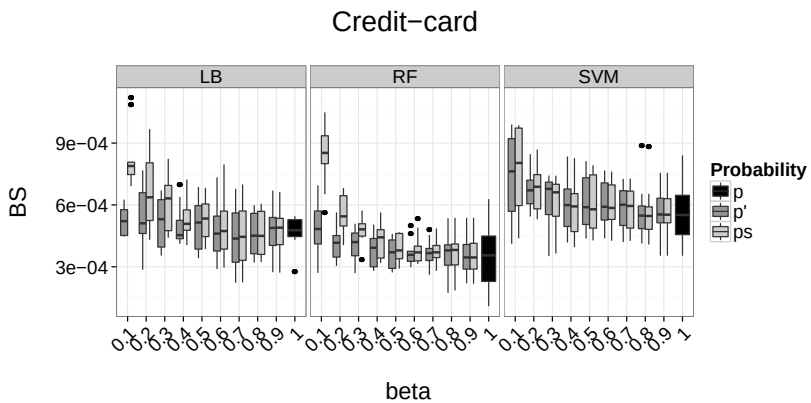
Results II

- The rank sum is the same for $\hat{p}_s$ and $\hat{p}'$ since (6) is monotone.
- Undersampling does not always improve the ranking (AUC) or classification accuracy (G-mean) of an algorithm.
- $\hat{p}$ is the probability estimate with the best calibration (lower rank sum with BS).
- $\hat{p}'$ has always better calibration than $\hat{p}_s$, then we should use $\hat{p}'$ instead of $\hat{p}_s$.

Credit cards dataset

Real-world credit card dataset with transactions from Sep 2013, frauds account for 0.172% of all transactions.





Conclusion

- As a result of undersampling, $\hat{p}_s$ is shifted away from $\hat{p}$.
- This shift is stronger for overlapping distributions and gets larger for small values of β.
- Using (6), we can remove the drift in $\hat{p}_s$ and obtain $\hat{p}'$ which has better calibration.
- $\hat{p}'$ provides the same ranking quality of $\hat{p}_s$.
- Results from UCI and credit card datasets show that using $\hat{p}'$ with τ' we are able to improve calibration without losing predictive accuracy.

Credit card dataset: `http://www.ulb.ac.be/di/map/adalpozz/data/creditcard.Rdata`
Website: `www.ulb.ac.be/di/map/adalpozz`
Email: adalpozz@ulb.ac.be

Thank you for the attention

Research is supported by the *Doctiris* scholarship funded by Innoviris, Brussels, Belgium.

Bibliography

[1] A. Dal Pozzolo, O. Caelen, and G. Bontempi.
When is undersampling effective in unbalanced classification tasks?
In *Machine Learning and Knowledge Discovery in Databases*. Springer, 2015.

[2] A. Dal Pozzolo, O. Caelen, Y.-A. Le Borgne, S. Waterschoot, and G. Bontempi.
Learned lessons in credit card fraud detection from a practitioner perspective.
Expert Systems with Applications, 41(10):4915–4928, 2014.

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

[4] A. Estabrooks, T. Jo, and N. Japkowicz.
A multiple resampling method for learning from imbalanced data sets.
Computational Intelligence, 20(1):18–36, 2004.

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

[6] A. Karatzoglou, A. Smola, K. Hornik, and A. Zeileis.
kernlab-an s4 package for kernel methods in r.
2004.

[7] A. Liaw and M. Wiener.
Classification and regression by randomforest.
R News, 2(3):18–22, 2002.

[8] M. Saerens, P. Latinne, and C. Decaestecker.
Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure.
Neural computation, 14(1):21–41, 2002.

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

[10] V. N. Vapnik and V. Vapnik.
Statistical learning theory, volume 1.
Wiley New York, 1998.

[11] G. M. Weiss and F. Provost.
The effect of class distribution on classifier learning: an empirical study.
Rutgers Univ, 2001.