Classification methods based on fitting logistic regression to positive and unlabeled data

Konrad Furmańczyk[0000−0002−7683−4787],
Kacper Paczutkowski[0000−0001−7408−6060],
Marcin Dudziński[0000−0003−4242−8411], and
Diana Dziewa-Dawidczyk[0000−0001−9486−1685]

Institute of Information Technology, Warsaw University of Life Sciences, Warsaw, Poland {konrad_furmanczyk,kacper_paczutkowski, marcin_dudzinski,diana_dziewa_dawidczyk}@sggw.edu.pl

Abstract. In our work, we consider the classification methods based on the model of logistic regression for positive and unlabeled data. We examine the following four methods of the posterior probability estimation, where the risk of logistic loss function is optimized, namely: the naive approach, the weighted likelihood approach, as well as the quite recently proposed methods - the joint approach, and the LassoJoint method. The objective of our study is to evaluate the accuracy, the recall, the precision and the F1-score of the considered classification methods. The corresponding assessments have been carried out on 13 machine learning model schemes by conducting some numerical experiments on selected real datasets.

Keywords: positive unlabeled learning · logistic regression · empirical risk minimization · thresholded Lasso.

1 Introduction

Learning from positive and unlabeled data, i.e. the so-called PU learning, is an approach where the only information the researcher has consists of positive examples and unlabeled data. In the PU setting, the training data contains positive and unlabeled examples, which means that the true labels $Y \in \{0, 1\}$ are not observed directly in the data and we only observe the surrogate variable $S \in \{0, 1\}$, which indicates whether an example is labeled (and consequently positive, $S = 1$ then) or not ($S = 0$ in this case). The history of PU learning dates back to the early 2000s (see, e.g., [10]) and this idea has gained much attention throughout recent years. The main reason for such a rapid development of the PU learning scheme is that this setting is very useful in numerous important applications. In particular, the PU learning method can be applied in the case when under-reporting is present in survey data (see [1]). It is quite common while analyzing some records from medical surveys, when we wish to predict the presence of a specific disease. Namely, it often happens that, although some respondents openly admit to suffering from a disease (the surrogate variable $S = 1$

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_3

and consequently, the true label $Y = 1$ in this case), there also exists a group of respondents who do not report such a disease (we put $S = 0$ then). This second group includes both the respondents who in fact have an examined disease, but do not admit to it (we have $Y = 1$ and $S = 0$ in this case) and the respondents who actually do not suffer from it (we have $Y = 0$ and $S = 0$ then). Such the under-reporting phenomenon is frequently justified by the fact that individuals suffering from some diseases (e.g. - from HIV or alcoholism) are often negatively perceived and treated by the rest of society. Some other interesting examples where the under-reporting is present may be found in the papers by Bekker and Davis [1] and Teisseyre et al. [15].

Now, suppose that X is a feature vector and, as previously, $Y \in \{0, 1\}$ stands for a true class label and $S \in \{0, 1\}$ denotes the surrogate variable that indicates, whether an example is labeled ($S = 1$ in this case) or not ($S = 0$ then). We consider a single sample scenario, where it is assumed that, there is a certain unknown distribution P, of (Y, X, S), such that $(Y_i, X_i, S_i), i = 1, \ldots, n$, form an iid sample obtained from this distribution, and that only empirical data $(X_i, S_i), i = 1, \ldots, n$, are observed. Thus, we do not have a traditional sample (X_i, Y_i), which is considered in standard classification problems, and we only observe a sample (X_i, S_i), where S_i are the observations of variable $S \in \{0, 1\}$ (since S is a surrogate of the true label Y, then each S_i depends on (X_i, Y_i)). In the considered concept only positive examples (i.e., examples for which $Y = 1$) may be labeled, which means that $P(S = 1|X, Y = 0) = 0$. It should be emphasized that in the PU design, the true class labels Y are only partially observed, which means that if $S = 1$, then we know that $Y = 1$, but if $S = 0$, then Y may be either 1 or 0.

The following constraint, called the Selected Completely At Random (SCAR) condition, is assumed

$$P(S = 1|Y = 1, X) = P(S = 1|Y = 1).$$

The SCAR assumption implies that X and S are independent given Y, since $P(S = 1|Y = 0, X) = P(S = 1|Y = 0) = 0$. Let $c = P(S = 1|Y = 1)$. The parameter c is called the label frequency and plays a key role in the PU learning scheme.
The main objective of our study is to apply the PU learning concept in order to estimate the posterior probability $f(x) = P(Y = 1|X = x)$, where, as previously, $Y \in \{0, 1\}$ denotes a true class label and X stands for the feature vector. Based on logistic model, three basic methods of this estimation have been proposed so far. They consist in minimizing the empirical risk of logistic loss function and are known as the naive method, the weighted method and the joint method (the latter has been quite recently introduced in Teisseyre et. al. [15]).
Now, let us briefly describe the above mentioned methods.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_3

First, we aim to present the naive method. In this case, having the empirical data (X_i, S_i), we minimize the empirical risk of the form

$$\widehat{R_1}(b) = -\frac{1}{n}\sum_{i=1}^{n}\left[S_i log(\sigma(X_i^T b)) + (1-S_i) log(1-\sigma(X_i^T b))\right],$$

where $\sigma(s) = 1/(1+exp(s))$. Then, the corresponding estimate of the posterior probability $f(x)$ is determined as

$$\widehat{f}_{naive}(x) = c^{-1}\sigma(x^T \widehat{b}_{naive}),$$

where c stands for the label frequency (i.e., $c = P(S=1|Y=1)$) and $\widehat{b}_{naive} = argmin_b \widehat{R_1}(b)$.

Using the weighted likelihood method (the weighted method in short, see [1]), we minimize the weighted empirical risk given by

$$\widehat{R_2}(b) = -\frac{1}{n}\sum_{i:S_i=1}\left[c^{-1} log(\sigma(X_i^T b)) + (1-c^{-1}) log(1-\sigma(X_i^T b))\right] + \sum_{i:S_i=0} log(1-\sigma(X_i^T b)).$$

Then, the corresponding estimator of the posterior probability $f(x)$ is expressed as

$$\widehat{f}_{weighted}(x) = c^{-1}\sigma(x^T \widehat{b}_{weighted}),$$

where $\widehat{b}_{weighted} = argmin_b \widehat{R_2}(b)$.

The joint method from Teisseyre et al. [15] consists in minimizing - with respect to both the parameter vector b and the label frequency c - the following empirical risk

$$\widehat{R_3}(b,c) = -\frac{1}{n}\sum_{i=1}^{n}\left[S_i log(\sigma(cX_i^T b)) + (1-S_i) log(1-\sigma(cX_i^T b))\right].$$

Then, the corresponding estimator of the posterior probability $f(x)$ is stated as follows

$$\widehat{f}_{joint}(x) = c^{-1}\sigma(x^T \widehat{b}_{joint}),$$

where $\left\{\widehat{b}_{joint}, \widehat{c}_{joint}\right\} = argmin_{b,c}\widehat{R_3}(b,c)$.

In order to optimize $\widehat{R_3}(b,c)$, the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm has been applied in Teisseyre et al. [15]. This algorithm enables to determine the formula for partial derivatives of $\widehat{R_3}(b,c)$. It is worth mentioning in this place that the Minorisation-Maximisation (MM) algorithm has been considered for the purpose of optimization by Łazęcka et al. [11].

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_3

In the most recent time, Furmańczyk et al. [5] proposed the LassoJoint procedure. It derives its name from the fact that it combines the thresholded Lasso procedure with the joint method from Teisseyre et al. [15]. It is a three-step procedure. Namely, in its two first steps, we perform - for some prespecified level - the thresholded Lasso procedure, in order to obtain the support for coefficients of a feature vector X, while in the third step, we apply - on the previously determined support - the joint method proposed by Teisseyre et al. [15]. More precisely, the LassoJoint method may be described as follows:

(1) For available PU dataset (S_i, X_i), $i = 1, \ldots, n$, we perform the ordinary Lasso procedure (see Tibshirani [18]) for some tuning parameter $\lambda > 0$, i.e. we compute the following Lasso estimator of β^*

$$\widehat{\beta}^{(L)} = \arg \min_{\beta \in R^{p+1}} \widehat{R}(\beta) + \lambda \sum_{j=1}^{p} |\beta_j| \,,$$

where

$$\widehat{R}(\beta) = -\frac{1}{n} \sum_{i=1}^{n} \left[S_i \log \left(\sigma(X_i^T \beta) \right) + (1 - S_i) \log \left(1 - \sigma(X_i^T \beta) \right) \right]$$

and subsequently, we obtain the corresponding support $Supp^{(L)} = \{1 \leq j \leq p : \widehat{\beta}_j^{(L)} \neq 0\}$;

(2) We perform the thresholded Lasso for some prespecified level δ and obtain the support $Supp^{(TL)} = \{1 \leq j \leq p : \left| \widehat{\beta}_j^{(L)} \right| \geq \delta\}$;

(3) We apply the joint method from Teisseyre et al. [15] for the predictors from $Supp^{(TL)}$.

It should be stressed that under some mild regularity conditions, the LassoJoint procedure obeys the screening property (all significant predictors of the model are chosen in the first two steps, see Theorem 1(b) in [5]).
Apart from the works where different learning methods - based on application of the logistic regression model for PU data - have been proposed, there are also some other interesting articles where various machine learning tools are used in the PU learning problems. In this context, it is worthwhile to mention: the papers of Hou [8] and Guo [6] - where the generative adversial networks (GAN) for the PU problem have been employed, the work of Mordelet and Vert [13] - where the bagging Support Vector Machine (SVM) approach for the PU data has been applied. Most relevant methods regarding the learning from PU data may be found in Lee and Liu [10] and Sansone et al. [14].

There are two essential objectives of our research. Its first goal is to verify and compare the accuracy, the recall, the precision and the F1-score of classifications obtained by the so far introduced primary methods of the posterior probability estimation, providing that Y is governed by the logistic regression

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_3

model and PU data are available. For the corresponding comparisons, we aim to use AdaSampling methods (see [20]). In turn, our second goal is to give a recommendation for the method that seems the most stable and efficient. The details regarding our study have been given in further parts of our work. The remainder of our paper is structured as follows. In Section 1, we present the ideas and concepts used in our investigations, especially the methods that enable attaining the set objectives. Furthermore - in Section 2 we introduce the applied models, in Section 3 we present our numerical experiments together with the obtained results, while Section 4 summarizes our study. In order to carry out our simulations, we used the RStudio server module from the ICM UW Topola server[1]. We implemented the following libraries: AdaSampling [21], e1071 [12], glmnet [4], and some additional libraries available from two selected GitHub repositories: PUlogistic [16], PU_class_prior [17].

2 Objectives and methods

The first goal of our study is to check and compare the accuracy, the recall, the precision and the F1-score of classifications obtained with use of the recently proposed methods - the joint method from Teisseyre et al. [15] and the LassoJoint approach from Furmańczyk et al. [5], as well as with use of the earlier established estimation methods consisting in fitting the logistic model, i.e. by additional application of the naive method, the weighted method and the oracle method for the case when the vector of coefficients is known.
The accuracy, recall, precision and F1-score metrics are defined as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN},$$

$$Recall = \frac{TP}{TP + FN},$$

$$Precision = \frac{TP}{TP + FP},$$

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall},$$

where TP, FN, TN and FP stand for: the number of true positives, false negatives, true negatives and false positives, respectively.

The assessments of the mentioned metrics have been gained by conducting some numerical studies on nine real datasets from the UCI Machine Learning Repository [2] and the 'caret' package [9]. The second purpose of our research is to recommend the most reliable and efficient estimation method for the posterior probability $f(x) = P(Y = 1|X = x)$ assessment, where – as in the previous

[1] This research was carried out with the support of the Interdisciplinary Centre for Mathematical and Computational Modelling (ICM) at the University of Warsaw, under computational allocation No. g88-1185.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_3

procedures – it is assumed that Y is governed by the logistic model and the PU data are available. Our study was constructed on 13 machine learning (ML) model schemes. We applied the LassoJoint method from [5] by considering the joint method for two scenarios - with the BFGS or the MM algorithm. The LassoJoint approach is a three-step procedure. In its first step, the initial selection of predictors is carried out by employing the Lasso method, for which the tuning parameter λ is either obtained by using the 10-fold cross-validation technique or is fixed. In turn, in the second step, the thresholded Lasso is performed, whereas in the last step, the joint method is employed for the variables selected in the second step. The naive logistic regression approach, the joint method, the LassoJoint approach and the weighted method for c estimated from the joint method (for the BFGS or the MM algorithm) have been employed and the corresponding results have been compared with the results obtained by implementing the oracle method when the true label variable Y is known. Moreover, in order to compare the classification methods based on fitting the logistic regression model, the two machine learning methods - namely, the Support Vector Machine (SVM) approach and the k-nearest neighbors algorithm (KNN) have been used - both in the AdaSampling scheme (see Yang et al. [19] and Yang et al. [20]). An application of the AdaSampling design results in constructing an adaptive sampling-based noise reduction method, which enables dealing with noisy data. We have also performed the min-max transformation of our features, which - compared to the original data - greatly improved the accuracy of all of the obtained results.

3 Numerical experiments

3.1 Datasets

We consider nine datasets from the UCI Machine Learning Repository [2] and the 'caret' package [9]. In Table 1, we present basic characteristics of each dataset (from left to right: the number of features, the number of observations, the number of binary and continuous variables, the number of negative and positive cases, the percentage of positive cases). The values of these characteristics are obtained through fundamental preprocessing, including the one-hot encoding and removing the missing values. In our simulations, we set 1-class as a larger class for each dataset. The selection of datasets was conducted by taking into account various types of potential difficulties that may appear while applying the ML methods. Thus, we tested both a strict low-dimensional datasets ('*Banknote*') and datasets with many predictors ('*Dhfr*'). In addition to that, we also considered the sets with only binominal ('*Vote*') or continuous predictors ('*Wdbc*', '*Spambase*') and mixed instances.

The naive logistic regression approach, the joint method, the LassoJoint approach and the weighted method for c estimated with use of the joint method have been employed. The corresponding results have been compared with the results obtained by implementing the oracle method. We deal with the problem

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_3

Table 1. Basic characteristics of the datasets. (from left to right: dataset name, no. of features, no. of observations, no. of binary variables, no. of continuous variables, no. of negative instances, no. of positive instances, percentage of positive instances)

dataset	p	n	nbin_variables	ncon_variables	n_0	n_1	**% of 1-class**
Banknote	4	1372	0	4	610	762	55.5%
Breastc	9	683	0	9	239	444	65.0%
Credit_a	37	653	31	6	296	357	54.7%
Credit_g	24	1000	12	12	300	700	70.0%
Dhfr	228	325	11	217	122	203	62.5%
Diabetes	8	768	0	8	268	500	65.1%
Spambase	57	4601	0	57	1813	2788	60.6%
Vote	32	435	32	0	168	267	61 %
Wdbc	30	569	0	30	212	357	62.7%

of PU data classification. From the above, completely labeled datasets, we randomly select $c\%$ of the labeled observations S, for $c = 0.1; 0.3; 0.5; 0.7; 0.9$, and then, we randomly split these datasets into the training sample (80%) and the test sample (20%). By applying the LassoJoint method in its first step, we use the Lasso method with tuning parameters λ, chosen either on the basis of the 10-fold cross-validation scheme - in the first scenario (where lambda.min gives the minimum mean cross-validated error, while lambda.1se stands for the largest value of λ such that an error is within 1 standard error of the cross-validated errors for lambda.min.) or by putting the fixed λ of the form $\lambda = ((\log p)/n)^{1/3}$ - in the second scenario, as in [5]. In the second step, we apply the thresholded Lasso design for $\delta = 0.5\lambda$, with λ selected in the first step. Next, we determine the classification metric by simulating from 100 Monte Carlo replications of our experiment. Subsequently, in order to compare the logistic regression-based classification methods, the tools of machine learning, such as an AdaSampling (see Yang et al. [19] and Yang et al. [20]) together with the Support Vector Machine (SVM) concept and the k-nearest neighbors algorithm (KNN) have been employed.

3.2 Results

We conducted our simulation study on 13 ML model schemes based on the four methods described in Introduction. In our work, we applied four measures based on the confusion matrix: the accuracy, the recall, the precision, the F1-score. All of our metrics are the averages of the obtained values of metrics on a test subset after 100 repetitions. We decided to set a cut-off point at the level of 0.5. This level is typical in cases when the logistic or the logistic-based models are fitted. In the examples from the *AdaSampling* package documentation [21] the level of 0.5 is commonly used. The average values of the accuracy and the recall are given in Fig.1 and Fig.2. Additionally, we provide a dedicated visualization for comparison between the joint-wise models with and without the Lasso component (see Fig.3 and Fig.4). Tables presenting the precise values

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_3

of some metrics and the charts depicting the values of the remaining measures are available in our Supplementary Materials [2]. These Supplementary Materials also include all of our codes in R.

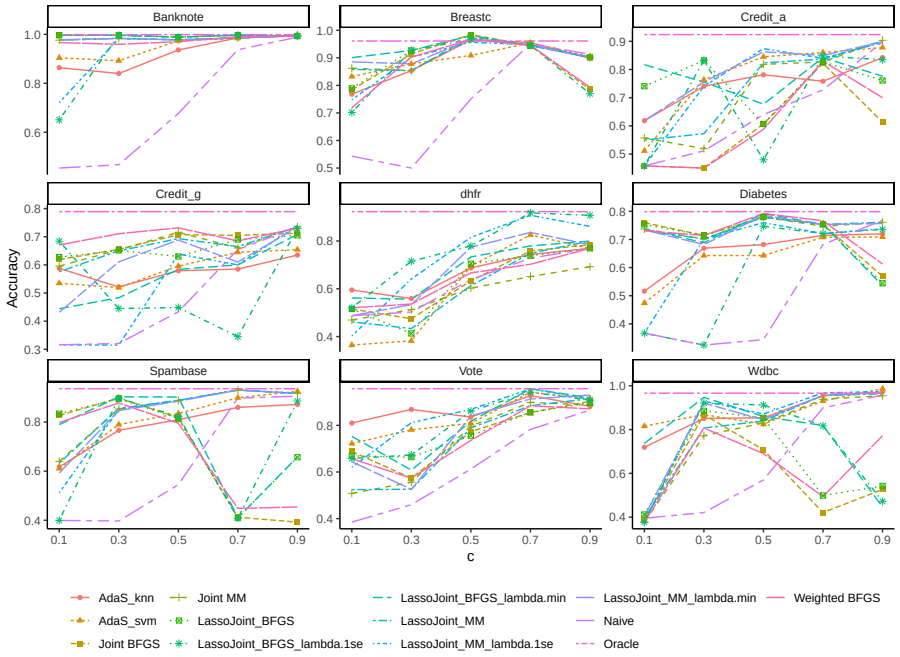

Fig. 1. The accuracy for the test datasets

It is clear that in the considered scenarios, performance of the oracle method may be perceived as a natural top ('the best') benchmark. On the other hand, in many scenarios the bottom ('the worst') benchmark is connected with performance of the naive method, but it may not always be treated as a strict rule.

Apart from obtaining appropriate metric values, we have also developed, for each value of c, the corresponding ranking methods. The ranking has been obtained on the basis of calculating the average values of ranks in a single scenario (the greater rank value is, the worse a given method is in our ranking). The ranking results are collected in Tables 2-5. The best methods (except the oracle approach) are underlined in the columns. Some additional comments and remarks regarding the obtained results are contained in the next section.

[2] http://github.com/kfurmanczyk/ICCS22

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_3

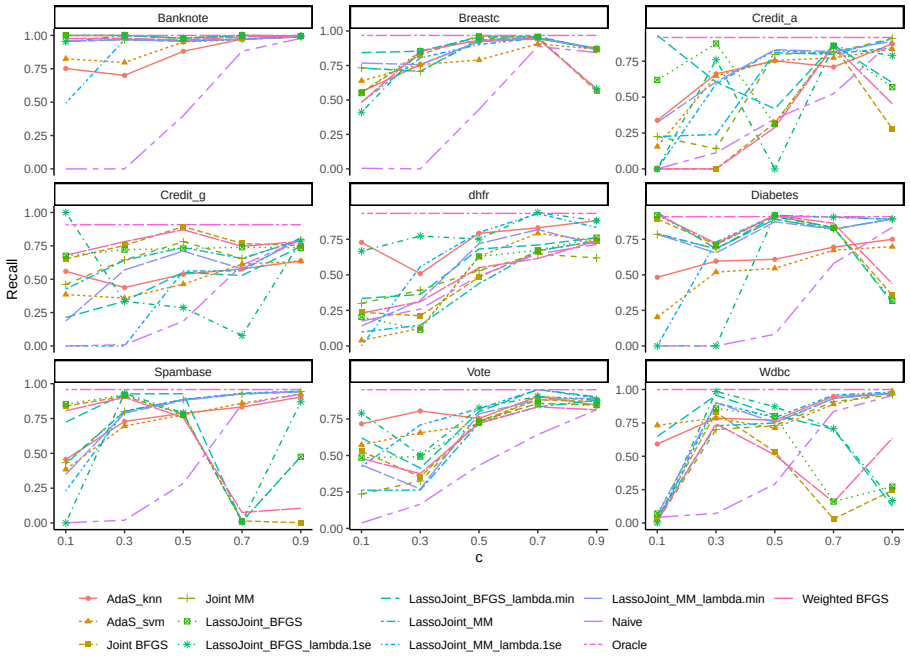

Fig. 2. The recall for the test datasets

Table 2. Avg.rank method based on the accuracy

method	$c = 0.1$	0.3	0.5	0.7	0.9
Oracle	1.00	1.00	2.00	1.11	1.56
LassoJoint_BFGS_lambda.min	4.11	5.33	5.00	7.22	9.11
LassoJoint_MM_lambda.min	7.22	8.22	5.56	6.33	5.22
LassoJoint_MM_lambda.1se	11.22	6.67	5.44	5.56	4.44
LassoJoint_MM	7.67	8.78	6.89	6.33	4.56
LassoJoint_BFGS	4.44	4.89	7.11	8.56	10.00
Joint MM	7.11	8.22	6.56	7.56	5.67
Joint BFGS	5.44	6.00	7.00	7.78	10.22
AdaS_svm	7.78	8.67	8.67	6.56	5.44
LassoJoint_BFGS_lambda.1se	9.00	5.22	6.78	7.56	8.56
AdaS_knn	6.11	8.22	8.78	8.44	8.56
Weighted BFGS	8.11	7.56	9.00	8.11	10.33
Naive	11.78	12.22	12.22	9.89	7.33

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_3

Table 3. Avg.rank method based on the recall

method	$c = 0.1$	0.3	0.5	0.7	0.9
Oracle	1.89	1.44	1.67	1.78	1.00
LassoJoint_BFGS_lambda.min	3.67	5.56	5.22	6.44	8.44
LassoJoint_BFGS	4.78	4.89	6.22	6.67	8.89
LassoJoint_BFGS_lambda.1se	8.11	4.78	5.89	6.56	7.56
LassoJoint_MM_lambda.1se	11.56	6.44	5.33	5.44	5.11
Joint BFGS	5.78	6.33	6.33	5.89	9.89
Joint MM	7.33	8.22	6.56	7.11	6.00
AdaS_knn	5.78	7.44	8.11	8.11	6.89
LassoJoint_MM_lambda.min	7.44	8.44	6.78	7.44	6.33
LassoJoint_MM	7.89	9.33	7.67	7.67	5.67
Weighted BFGS	7.11	6.78	8.67	8.22	10.33
AdaS_svm	7.89	9.11	10.22	8.67	6.11
Naive	11.78	12.22	12.33	11.00	8.78

Table 4. Avg.rank method based on the precision

method	$c = 0.1$	0.3	0.5	0.7	0.9
AdaS_svm	5.44	3.89	2.89	3.22	5.00
Oracle	3.00	5.67	5.67	6.00	5.56
LassoJoint_MM_lambda.min	3.00	5.11	6.11	7.22	8.00
LassoJoint_MM	5.33	4.67	7.22	6.56	7.56
LassoJoint_BFGS_lambda.min	6.78	7.67	7.78	6.56	6.11
Naive	10.22	7.78	5.00	4.00	8.11
LassoJoint_BFGS	5.56	7.56	9.00	8.56	4.67
Joint BFGS	8.22	7.67	7.33	7.44	4.78
Joint MM	5.89	6.56	6.89	8.44	8.89
LassoJoint_MM_lambda.1se	10.11	7.33	6.33	8.00	8.22
AdaS_knn	7.22	8.00	6.78	8.78	10.56
LassoJoint_BFGS_lambda.1se	10.22	9.11	9.67	6.78	8.00
Weighted BFGS	10.00	10.00	10.33	9.44	5.56

Table 5. Avg.rank method based on the F1-score

method	$c = 0.1$	0.3	0.5	0.7	0.9
Oracle	1.00	1.00	2.00	1.11	1.56
LassoJoint_BFGS_lambda.min	4.56	5.44	5.33	7.56	9.11
LassoJoint_MM_lambda.1se	10.33	6.67	5.78	5.44	4.11
LassoJoint_BFGS_lambda.1se	7.89	4.00	6.00	7.00	8.11
LassoJoint_MM_lambda.min	7.89	8.56	6.22	6.44	5.56
LassoJoint_BFGS	4.22	5.11	7.11	8.56	10.00
Joint BFGS	5.33	6.33	6.00	7.89	10.56
Joint MM	7.22	8.44	7.00	7.78	5.89
LassoJoint_MM	8.33	9.44	7.44	6.78	5.22
AdaS_svm	8.44	8.11	9.22	6.67	5.56
AdaS_knn	6.44	8.33	8.33	8.00	7.78
Weighted BFGS	7.56	7.22	8.22	7.89	9.89
Naive	11.78	12.33	12.33	9.89	7.67

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_3

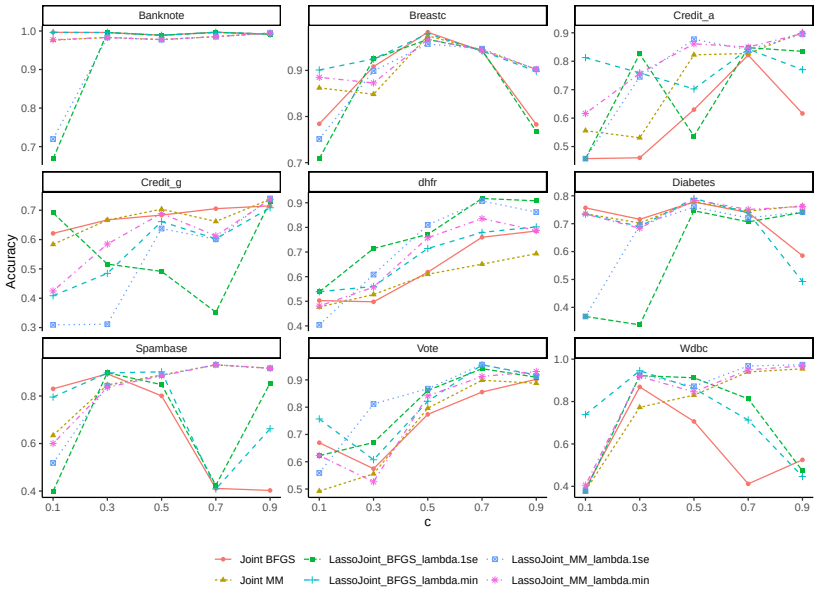

Fig. 3. The accuracy for the test datasets - the joint-wise models

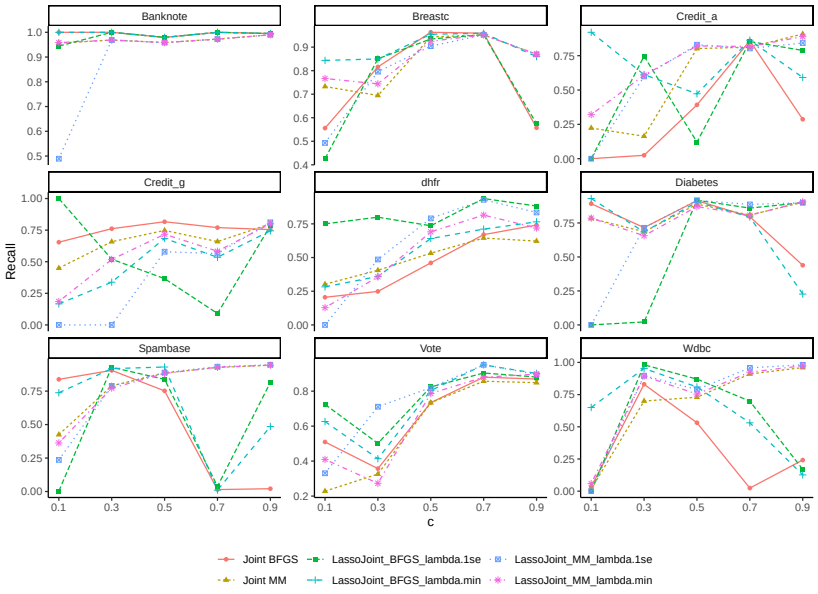

Fig. 4. The recall for the test datasets - the joint-wise models

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_3

4 Conclusions

The primary purpose of our study was to conduct a comprehensive evaluation of 13 ML model schemes, including the methods from literature and the methods obtained as a result of some modifications we implemented into some other procedures (such that conducting the MM optimization in the LassoJoint procedure). We decided to apply a few measures in our research, in order to get a guarantee of a proper complexity of our assessments. In a typical approach to PU problems, the main attention is focus on calculating the AUC and Accuracy metrics, whereas in our work we provide additional analysis regarding different assessment measures. This extension enables to evaluate fractions of the true '1-class' and fractions of the predicted '1-class', among the real positive instances, which may be very useful in many applications regarding popular PU problems. For instance, in the credit risk management, we want to detect all frauds, even if we label too many observations (equivalently - we agree for a larger type I error). In this case we need to control the recall measure with a greater emphasis. On the other hand, in various marketing campaigns related models (e.g., such as uplifting models), we wish to focus our attention on customers who actually want to buy certain products. In this case we prefer to control the precision measure. The results of our numerical experiments show that if c increases, then the percentage of correct classifications increases as well in most cases. Usually, the LassoJoint procedure helps to improve the classification metrics and prevails over other methods (see Tables 2-5 and Tables 1-35 in the Supplementary Materials). The LassoJoint method has been constructed for the high-dimensional cases (i.e., when $p > n$), but it has to be stressed that it may be also so in the low-dimensional cases (i.e., when $p < n$), as we observe that the joint method performance improves while applying the basic metrics on most of the tested datasets, except for the *Credit_g*, *Diabetes*, and *Spambase*. Only in few cases, the method based on the BFGS optimization performs worse for large values of c, but the corresponding accuracy is still acceptable for small values of c. We may also observe that the classifications obtained by applying the LassoJoint method with the MM algorithm result in smaller classification errors (and thus in better classification accuracy) for larger labeling levels c. Moreover, the methods with tuning values λ, obtained by using the cross-validation scheme, display better accuracy than the methods with fixed values. Based on the obtained accuracy, recall and F1-score, we recommend using the LassoJoint method with: (a) the BFGS variant - for small values of c, (b) the MM variant for the values of c above 0.5 (for comparison - see Fig. 3). Furthermore, it is worth mentioning that considering the selected cases with small values of c, we do not observe classification instances from the '1-class'. Most of this cases are connected with the naive method for $c = 0.1, 0.3$, which can be seen in Fig. 2 and therefore, using more complex methods is highly recommended in these cases. However, it is not easy to point out a general winning method by taking into account all of considered measures. For example, an application of AdaSampling with the SVM kernel provides the classification results of the highest precision for almost every dataset scenarios. This high level of precision assures greater certainty that

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_3

the predicted positives are real positives. On the other hand, the values of the accuracy, the recall and the F1-score are not satisfactory in most cases. In addition to that, the obtained simulations show that the labels noising can boost the precision metrics, since some methods provide better values of precision measures than the oracle approach (see Table 4). It is important to remember that all of the methods based on fitting the logistic regression model assume the celebrated SCAR condition. It is a common approach to impose this assumption in majority of methods dealing with PU learning and only in very few approaches the researchers try to omit this constraint (see [1]). In further investigations, it would be interesting to introduce some new methods which will not require the SCAR assumption. It would also be interesting to check robustness of existing methods under some disturbances of the SCAR condition.

References

1. Bekker, J., Davis, J.: Learning from positive and unlabeled data: a survey. Available from http: arxiv.org/abs/1811.04820v3 (2020)
2. Dua, D. and Graff, C.: UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science (2019)
3. Friedman, J., Hastie, T., Simon, N., Tibshirani, R.: Glmnet: Lasso and elastic-net regularized generalized linear models. R package version 2.0 (2015)
4. Friedman, J., Hastie, T., Tibshirani, R.: Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of Statistical Software, **33**(1), 1-22, (2010). https://www.jstatsoft.org/v33/i01/.
5. Furmańczyk, K., Dudziński, M., Dziewa-Dawidczyk, D.: Some proposal of the high dimensional PU learning classification procedure. ICCS 2021, Lecture Notes In Computer Science **12744**, 18-25 (2021)
6. Guo, T., Xu, C., Huang, J., Wang, Y., Shi, B., Xu, C., Tao, D.: On positive-unlabeled classification in GAN. CVPR (2020)
7. Hastie, T., Fithian, W.: Inference from presence-only data; the ongoing controversy. Ecography **36**: 864–867 (2013)
8. Hou, M., Chaib-draa, B., Li, C., Zhao, Q.: Generative adversarial positive-unlabeled learning. Proceedings of the twenty-seventh International Joint Conference on Artificial Intelligence (IJCAI-18) (2018)
9. Kuhn, M. caret: Classification and Regression Training. R package version 6.0-90, (2021). https://CRAN.R-project.org/package=caret
10. Lee, W. S. and Liu, B. Learning with Positive and Unlabeled Examples Using Weighted Logistic Regression. In ICML, Washington, D.C., AAAI Press, 448–455, Aug. (2003)
11. Łazęcka, M., Mielniczuk J., Teisseyre, P.: Estimating the class prior for positive and unlabelled data via logistic regression. Advances in Data Analysis and Classification **15**, 1039–1068 (2021)
12. Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., Leisch, F. e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. R package version 1.7-9 (2021). https://CRAN.R-project.org/package=e1071

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_3

13. Mordelet, F., Vert, J.P.: A bagging SVM to learn from positive and unlabeled examples. Pattern Recognition Letters (2013)
14. Sansone, E., De Natale, F. G. B., Zhou, Z.H.: Efficient Training for Positive Unlabeled Learning. TPAMI, Jul. (2018)
15. Teisseyre, P., Mielniczuk J., Łazęcka, M.: Different strategies of fitting logistic regression for positive and unlabelled data. Computational Sciences-ICCS 2020: 3–17 (2020)
16. Teisseyre, P.: Repository from https://github.com/teisseyrep/Pulogistic. Accessed 25 Jan 2022
17. Teisseyre, P.: Repository from, https://github.com/teisseyrep/PU_class_prior. Accessed 25 Jan 2022
18. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. Roy. Statist. Soc. Ser. B **58**, 267–288 (1996)
19. Yang, P., Liu, W., Yang. J.: 6. Positive unlabeled learning via wrapper-based adaptive sampling. 6. International Joint Conferences on Artificial Intelligence (IJCAI), 3272-3279 (2017)
20. Yang, P., Ormerod, J., Liu, W., Ma, C., Zomaya, A., Yang, J.: 7. AdaSampling for positive-unlabeled and label noise learning with bioinformatics applications.7. IEEE Transactions on Cybernetics, doi:10.1109/TCYB.2018.2816984 (2018)
21. Yang, P. AdaSampling: Adaptive Sampling for Positive Unlabeled and Label Noise Learning. R package version 1.3. (2019). https://CRAN.R-project.org/package=AdaSampling

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_3