Different Strategies of Fitting Logistic Regression for Positive and Unlabelled Data

Paweł Teisseyre[1(✉)], Jan Mielniczuk[1,2], and Małgorzata Łazęcka[1,2]

[1] Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland
{teisseyrep,miel,malgorzata.lazecka}@ipipan.waw.pl
[2] Faculty of Mathematics and Information Sciences, Warsaw University of Technology, Warsaw, Poland

Abstract. In the paper we revisit the problem of fitting logistic regression to positive and unlabelled data. There are two key contributions. First, a new light is shed on the properties of frequently used naive method (in which unlabelled examples are treated as negative). In particular we show that naive method is related to incorrect specification of the logistic model and consequently the parameters in naive method are shrunk towards zero. An interesting relationship between shrinkage parameter and label frequency is established. Second, we introduce a novel method of fitting logistic model based on simultaneous estimation of vector of coefficients and label frequency. Importantly, the proposed method does not require prior estimation, which is a major obstacle in positive unlabelled learning. The method is superior in predicting posterior probability to both naive method and weighted likelihood method for several benchmark data sets. Moreover, it yields consistently better estimator of label frequency than other two known methods. We also introduce simple but powerful representation of positive and unlabelled data under Selected Completely at Random assumption which yields straightforwardly most properties of such model.

Keywords: Positive unlabelled learning · Logistic regression · Empirical risk minimization · Misspecification

1 Introduction

Learning from positive and unlabelled data (PU learning) has attracted much interest within the machine learning literature as this type of data naturally arises in many applications (see e.g. [1]). In the case of PU data, we have an access to positive examples and unlabeled examples. Unlabeled examples can be either positive or negative. In this setting the true class label $Y \in \{0, 1\}$ is not observed directly. We only observe surrogate variable $S \in \{0, 1\}$, which indicates whether an example is labeled (and thus positive; $S = 1$) or unlabeled ($S = 0$). PU problem naturally occurs in under-reporting [2] which frequently happens in survey data, and it

V. V. Krzhizhanovskaya et al. (Eds.): ICCS 2020, LNCS 12140, pp. 3–17, 2020.
https://doi.org/10.1007/978-3-030-50423-6_1

refers to situation when some respondents fail to the answer a question truthfully. For example, imagine that we are interested in predicting an occurrence of some disease ($Y = 1$ denotes presence of disease and $Y = 0$ its absence) using some feature vector X. In some cases we only have an access to self-reported data [3], i.e. respondents answer to the question concerning the occurrence of the disease. Some of them admit to the disease truthfully ($S = 1 \implies Y = 1$) and the other group reports no disease ($S = 0$). The second group consists of respondents who suffer from disease but do not report it ($Y = 1, S = 0$) and those who really do not have a disease ($Y = 0, S = 0$). Under-reporting occurs due to a perceived social stigma concerning e.g. alcoholism, HIV disease or socially dangerous behaviours such as talking on the phone frequently while driving. PU data occur frequently in text classification problems [4–6]. When classifying user's web page preferences, some pages can be bookmarked as positive ($S = 1$) whereas all other pages are treated as unlabelled ($S = 0$). Among unlabelled pages, one can find pages that users visit ($Y = 1, S = 0$) as well as those which are avoided by users ($Y = 0, S = 0$). The third important example is a problem of disease gene identification which aims to find which genes from the human genome are causative for diseases [7,8]. In this case all the known disease genes are positive examples ($S = 1$), while all other candidates, generated by traditional linkage analysis, are unlabelled ($S = 0$). Several approaches exist to learn with PU data. A simplest approach is to treat S as a class label (this approach is called naive method or non-traditional classification) [9]. To organize terminology, learning with true class label Y will be called oracle method. Although this approach cannot be used in practice, it may serve as a benchmark method with which all considered methods are compared.

In this paper we focus on logistic regression. Despite its popularity, there is a lack of thorough analysis of different learning methods based on logistic regression for PU data. We present the following novel contributions. First, we analyse theoretically the naive method and its relationship with oracle method. We show that naive method is related to incorrect specification of the logistic model and we establish the connection between risk minimizers corresponding to naive and oracle methods, for certain relatively large class of distributions. Moreover, we show that parameters in naive method are shrunk towards zero and the amount of shrinkage depends on label frequency $c = P(S = 1|Y = 1)$. Secondly, we propose an intuitive method of parameter estimation in which we simultaneously estimate parameter vector and label frequency c (called joint method hereafter). The method does not require prior estimation which is a difficult task in PU learning [10,11]. Finally, we compare empirically the proposed method with two existing methods (naive method and the method based on optimizing weighted empirical risk, called briefly weighted method) with respect to estimation errors.

Finally, the popular taxonomy used in PU learning [1] differentiates between three categories of methods. The first group are postprocessing methods which first use naive method and then modify output probabilities using label frequency [9]. The second group are preprocessing methods that weigh the examples using label frequency [12–14]. We refer to [1] (Sect. 5.3.2) for a description of general empirical risk minimization framework in which the weights of observations

depending on label frequency c, for any loss function are determined. The last group are methods incorporating label frequency into learning algorithms. A representative algorithm from this group is POSC4.5 [15], which is PU tree learning method. The three methods considered in this paper (naive, weighted and joint method) represent the above three categories, respectively.

This paper is organized as follows. In Sect. 2, we state the problem and discuss its variants and assumptions. In Sect. 3, we analyse three learning methods based on logistic regression in detail. Section 4 discusses the relationship between naive and oracle methods. We report the results of experiments in Sect. 5 and conclude the paper in Sect. 6. Technical details are stated in Sect. 7. Some additional experiments are described in Supplement[1].

2 Assumptions and Useful Representation for PU Data

In this work we consider single training data (STD) scenario, which can be described as follows. Let X be feature vector, $Y \in \{0, 1\}$ be a true class label and $S \in \{0, 1\}$ an indicator of whether an example is labelled ($S = 1$) or not ($S = 0$). We assume that there is some unknown distribution $P(Y, X, S)$ such that $(y_i, x_i, s_i), i = 1, \ldots, n$ is iid sample drawn from it and data $(x_i, s_i), i = 1, \ldots, n$, is observed. Thus, instead of a sample (x_i, y_i) which corresponds to classical classification task, we observe only sample (x_i, s_i), where s_i depends on (x_i, y_i). Only positive examples ($Y = 1$) can be labelled, i.e. $P(S = 1|X, Y = 0) = 0$. The true class label is observed only partially, i.e. when $S = 1$ we know that $Y = 1$, but when $S = 0$, then Y can be either 1 or 0. A commonly used assumption is SCAR (Selected Completely At Random) assumption which states that labelled examples are selected randomly from a set of positives examples, independently from X, i.e.

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

Note that this is equivalent to X and S being independent given Y (denoted $X \perp S|Y$) as $P(S = 1|Y = 0, X) = P(S = 1|Y = 0) = 0$. Parameter $c := P(S = 1|Y = 1)$ is called label frequency and plays an important role in PU learning. In the paper we introduce a useful representation of variable (X, S) under SCAR assumption. Namely, we show that S can be represented as

$$S = Y \cdot \varepsilon, \text{ where } \varepsilon \perp (X, Y) \text{ and } \varepsilon \sim Bern(1, p), \tag{1}$$

for a certain $0 < p < 1$ and $Bern(1, p)$ stands for Bernoulli distribution. Indeed, we have $S = Y\varepsilon \perp X$ given Y, as $\varepsilon \perp (X, Y)$ implies that $\varepsilon \perp X$ given Y. Moreover,

$$P(S = 1|Y = 1) = P(Y\varepsilon = 1|Y = 1) = P(\varepsilon = 1) = p.$$

Thus probability of success $P(\varepsilon = 1)$ coincides with c. Under SCAR assumption we have

$$P(Y = 1|X) = c^{-1}P(S = 1|X), \tag{2}$$

[1] https://github.com/teisseyrep/PUlogistic.

$$P(Y=1|S=0,X)=\frac{1-c}{c}\frac{P(S=1|X)}{P(S=0|X)} \tag{3}$$

[9] and

$$P(X=x|Y=1)=P(X=x|S=1). \tag{4}$$

[2]. Properties (2)–(4) are easily derivable when (1) is applied (see Sect. 7).

We also note that the assumed STD scenario should be distinguished from case-control scenario when two independent samples are observed: labeled sample consisting of independent observations drawn from distribution of X given $Y=1$ and the second drawn from distribution of X. This is carefully discussed in [1]. Both PU scenarios should be also distinguished from semi-supervised scenario when besides fully observable sample from distribution of (X,Y) we also have at our disposal sample from distribution of X [16] or, in extreme case, we have full knowledge of distribution of X, see [17] and references therein. One of the main goals of PU learning is to estimate the posterior probability $f(x) := P(Y=1|X=x)$. The problem is discussed in the following sections.

3 Logistic Regression for PU Data

In this section we present three different methods of estimating $f(x) := P(Y=1|X=x)$ using logistic loss. When data is fully observed the natural way to learn a model is to consider risk for logistic loss

$$R(b)=-E_{X,Y}[Y\log(\sigma(X^Tb))+(1-Y)\log(1-\sigma(X^Tb))], \tag{5}$$

where $\sigma(s)=1/(1+\exp(-s))$ and minimize its empirical version. This will be called oracle method. Note that using logistic loss function in the definition of $R(b)$ above corresponds to fitting logistic regression using Maximum Likelihood (ML) method. Obviously, for PU data, this approach is not feasible as we do not observe Y and inferential procedures have to be based on (S,X). The simplest approach (called naive estimation or non-traditional estimation) is thus to consider risk

$$R_1(b)=-E_{X,S}[S\log(\sigma(X^Tb))+(1-S)\log(1-\sigma(X^Tb))] \tag{6}$$

and the corresponding empirical risk

$$\hat{R}_1(b)=-\frac{1}{n}\sum_{i=1}^{n}[s_i\log(\sigma(x_i^Tb))+(1-s_i)\log(1-\sigma(x_i^Tb))],$$

which can be directly optimized. In Sect. 4 we study the relationship between minimizers of $R(b)$ and $R_1(b)$

$$b^*=\arg\min_b R(b),\quad b_1^*=\arg\min_b R_1(b).$$

It turns out that for certain, relatively large, class of distributions of X, $b_1^* = \eta b^*$, for some $\eta \in R$ (i.e. b_1^* and b^* are collinear). Moreover, when predictors X are normal and when (Y, X) corresponds to logistic model, we establish the relationship between η and label frequency c which shows that $\eta < 1$ and thus naive approach leads to shrinking of vector b^*. To estimate the posterior $f(x) = P(Y = 1|X = x)$ using naive estimation, we perform a two-step procedure, i.e. we first estimate $\hat{b}_{\text{naive}} = \arg\min_b \hat{R}_1(b)$ and then let $\hat{f}_{\text{naive}}(x) := c^{-1}\sigma(x^T\hat{b}_{\text{naive}})$, where unknown c has to be estimated using some external procedure. Note that even when (Y, X) corresponds to logistic regression model, b^* and whence posterior probability is not consistently estimated by naive method.

The second approach is based on weighted empirical risk minimization. As mentioned before, the empirical counterpart of risk $R(b)$ cannot be directly optimized as we do not observe Y. However it can be shown [1] that

$$R(b) = -P(S=1)E_{X|S=1}\left[\frac{1}{c}\log\sigma(X^Tb) + \left(1-\frac{1}{c}\right)\log(1-\sigma(X^Tb))\right]$$
$$+P(S=0)E_{X|S=0}\log(1-\sigma(X^Tb)).$$

The risk above is approximated by

$$\widehat{R}(b) = -\frac{1}{n}\sum_{i:s_i=1}\left[\frac{1}{c}\log\sigma(x_i^Tb) + \left(1-\frac{1}{c}\right)\log(1-\sigma(x_i^Tb))\right]$$
$$+\frac{1}{n}\sum_{i:s_i=0}\log(1-\sigma(x_i^Tb)).$$

This means that all unlabelled examples are assigned weight 1, whereas each labelled example is treated as a combination of positive example with weight $1/c$ and negative example with weight $(1-1/c)$. The posterior estimator is defined as $\widehat{f}_{\text{weighted}}(x) = \sigma(x^T\hat{b}_{\text{weighted}})$, where $\hat{b}_{\text{weighted}} = \arg\min_b \widehat{R}(b)$. The above idea of weighted empirical risk minimization was used in case-control scenario for which the above formulas have slightly different forms, see [12,13].

In the paper we propose a novel, intuitive approach, called joint method (name refers to joint estimation of b and c). In this method we avail ourselves of an important feature of logistic regression, namely that posterior probability is directly parametrized. This in turn allows to directly plug in the equation (2) into the risk function

$$R_2(b,c) = -E_{X,S}[S\log(c\sigma(X^Tb)) + (1-S)\log(1-c\sigma(X^Tb))].$$

The empirical counterpart of the above risk is

$$\widehat{R}_2(b,c) = -\frac{1}{n}\sum_{i=1}^{n}[s_i\log(c\sigma(x_i^Tb)) + (1-s_i)\log(1-c\sigma(x_i^Tb))].$$

The empirical risk $\widehat{R}_2(b,c)$ can be optimized with respect to b if c is assumed to be known or can be optimized simultaneously with respect to both b and c.

In the latter case the posterior estimator is $\hat{f}_{\text{joint}}(x) := \sigma(x^T \hat{b}_{\text{joint}})$ where $(\hat{b}_{\text{joint}}, \hat{c}_{\text{joint}}) = \arg\min_{b,c} \hat{R}_2(b,c)$. Note that when conditional distribution of Y given X is governed by logistic model i.e. $P(Y=1|X=x) = \sigma(\beta^T x)$, for some unknown vector β, then in view of (2) $P(S=1|X=x) = c\sigma(\beta^T x)$ and $\widehat{R}_2(b,c)$ is log-likelihood for observed sample (x_i, s_i). Whence under regularity conditions, maximisation of $\widehat{R}_2(b,c)$ yields consistent estimator of (β, c) in view of known results in consistency of maximum likelihood method. To optimize function $\widehat{R}_2$ we use BFGS algorithm, which requires the knowledge of functional form of gradient. The partial derivatives of $\widehat{R}_2$ are given by

$$\frac{\partial \widehat{R}_2(b,c)}{\partial b} = -\frac{1}{n}\sum_{i=1}^{n} x_i \sigma(x_i^T b)(1-\sigma(x_i^T b)) \left[\frac{s_i - c\sigma(x_i^T b)}{\sigma(x_i^T b)(1-c\sigma(x_i^T b))}\right],$$

$$\frac{\partial \widehat{R}_2(b,c)}{\partial c} = -\frac{1}{n}\sum_{i=1}^{n} \left[\frac{s_i}{c} - \frac{(1-s_i)\sigma(x_i^T b)}{1-c\sigma(x_i^T b)}\right].$$

For $c = 1$, the first equation above reduces to well-known formula for gradient of the maximum likelihood function for standard logistic regression. In general we observe quick convergence of BFGS algorithm. The proposed method is described by the following scheme.

Algorithm 1. Joint method for posterior estimation

Input : Observed data (x_i, s_i), $i = 1, \ldots, n$; new instance x
$(\hat{b}_{\text{joint}}, \hat{c}_{\text{joint}}) = \arg\min_{b,c} -\frac{1}{n}\sum_{i=1}^{n}[s_i \log(c\sigma(x_i^T b)) + (1-s_i)\log(1-c\sigma(x_i^T b))]$
Compute $\hat{f}_{\text{joint}}(x) := \sigma(x^T \hat{b}_{\text{joint}})$
Output : $\hat{f}_{\text{joint}}(x)$

Finally, we note that the joint method above is loosely related to non-linear regression fit in dose-response analysis when generalized logistic curve is fitted [18].

4 Naive Method as an Incorrect Specification of Logistic Regression

In this Section we show that naive method is related to incorrect specification of the logistic model and that the corresponding parameter vector will be shrunk towards zero for relatively large class of distributions of X. Moreover, we establish the relationship between the amount of shrinkage and label frequency.

Assume for simplicity of exposition that components of X are non-constant random variables (in the case when one of predictors is a dummy variable which allows for the intercept in the model, collinearity in (9) corresponds to vector of

predictors with dummy variable omitted) and assume that regression function of Y given X has the following form

$$P(Y = 1|X = x) = q(\beta^T X), \tag{7}$$

for a certain response function q taking its values in $(0, 1)$ and a certain $\beta \in R^p$. We note that when oracle method (5) is correctly specified, i.e. $q(\cdot) = \sigma(\cdot)$, then $\beta = b^*$ (cf [19]). Here we consider more general situation in which we may have $q(\cdot) \neq \sigma(\cdot)$. Under SCAR assumption, $P(S = 1|X = x) = cq(\beta^T X)$ and thus when $cq(\cdot) \neq \sigma(\cdot)$ then maximising $\widehat{R}_1(b)$ corresponds to fitting misspecified logistic model to (X, S). Importantly, this model is misspecified even if the oracle model is correctly specified. Observe that in this case shrinking of parameters is intuitive as they have to move towards 0 to account for diminished ($c < 1$) aposteriori probability. We explain in the following why misspecified fit, which occurs frequently in practice may still lead to reasonable results. Assume namely that distribution of X satisfies linear regression condition (LRC)

$$E(X|\beta^T X = x) = wx + w_0 \tag{8}$$

for a certain $w_0, w \in R^p$. Note that (8) has to be satisfied for a true β only. LRC is fulfilled (for all β) by normal distribution, and more generally, by a larger class of elliptically contoured distributions (multivariate t-Student distribution is a representative example). Then it follows (see e.g. [20])

$$b_1^* = \eta\beta \tag{9}$$

and $\eta \neq 0$ provided $\mathrm{Cov}(Y, X) \neq 0$. In this case true vector β and its projection on a logistic model are collinear which partly explains why logistic classification works even when data does not follow logistic model. When oracle method (5) is correctly specified, i.e. $q(\cdot) = \sigma(\cdot)$, then (9) can be written as

$$b_1^* = \eta b^* = \eta\beta, \tag{10}$$

i.e. risk minimizers corresponding to naive and oracle methods are collinear. In the following we investigate the relationship between label frequency c and collinearity factor η. Intuition suggests that small c should result in shrinking of estimators towards zero. First, we have a general formula (see [19] for derivation) describing the relationship between c and η when (7) holds

$$\frac{1}{c} = \frac{E_X[\sigma(X^T\beta)X_j]}{E_X[\sigma(X^T b_1^*)X_j]} = \frac{E_X[\sigma(X^T\beta)X_j]}{E_X[\sigma(X^T\eta\beta)X_j]},$$

for any j, where X_j is j-th coordinate of $X = (X_1, \ldots, X_p)$. Unfortunately, the above formula does not yield simple relationship between c and η. Some additional assumptions are needed to find more revealing one. In the case when X has normal distribution $N(0, \Sigma)$ it follows from [20] together with (2) that the following equality holds

$$\frac{E\sigma'(\beta^T X)}{E\sigma'(\eta\beta^T X)} = \frac{\eta}{c}, \tag{11}$$

where $\sigma'(s)$ denotes derivative of $\sigma(s)$ wrt to s. This is easily seen to be a corollary of Stein's lemma stating that $\text{Cov}(h(Z_1), Z_2) = \text{Cov}(Z_1, Z_2)Eh'(Z_1)$ for bivariate normal (Z_1, Z_2). Equation (11) can be used to find upper and lower bounds for η. Namely, we prove the following Theorem.

Theorem 1. *Assume that X follows normal distribution $N(0, \Sigma)$ and that linear regression condition holds (8). Then*

$$4cE\sigma'(\beta^T X) \leq \eta \leq c\frac{E\sigma'(\beta^T X)}{E\sigma'(c\beta^T X)} \leq c. \tag{12}$$

Note that RHS inequality in (1) yields the lower bound on the amount of shrinkage of true vector β^* whereas LHS gives a lower bound on this amount.

Proof. Let $Z = \beta^T X$ and note that Z has normal distribution $N(0, a^2)$ with $a^2 = \beta^T \Sigma \beta$. It follows from the fact that $\sigma'(s) = \sigma(s)(1 - \sigma(s))$ is nonincreasing for $s > 0$ that function $h(\lambda) = E\sigma'(\lambda Z)$ is non-increasing. This justifies the last equality on the right as $c \leq 1$. Define $g(\lambda) = h(1) - (\lambda/c)h(\lambda)$ and note that $g(0) = h(1) > 0$, $g(c) \leq 0$ and g is continuous. Thus for a certain $\lambda_0 \in [0, c]$ it holds that $g(\lambda_0) = 0$ and it follows from (11) and uniqueness of projection that $\eta = \lambda_0$. In order to prove the RHS inequality it is enough to prove that $g(\lambda)$ is convex as then $\lambda_0 \leq \lambda^*$, where λ^* is a point at which a line $h(1) - \lambda h(c)/c$ joining points $(0, g(0))$ and $(c, g(c))$ crosses x-axis. As $\lambda^* = (h(1)/h(c))c$ the inequality follows. Convexity of g follows from concavity of $\lambda h(\lambda)$ which is proved in Supplement. In order to prove the left inequality it is enough to observe that $\sigma'(x) \leq 1/4$ and use (11) again.

Note for $c \to 0$ the ratio of the lower and upper bound tends to 1 as $E\sigma'(c\beta^T X) \to 1/4$. To illustrate the above theoretical result we performed simulation experiment in which we artificially generated a sample of size $n = 10^6$ in such a way that X followed 3-dimensional standard normal distribution and Y was generated from (7) with $q(\cdot) = \sigma(\cdot)$, with known β. Then $Z = \beta^T X$ has $N(0, ||\beta||^2)$ distribution and the bounds in (12) depend only on c and $||\beta||$. Figure 1 shows how collinearity parameter η and the corresponding bounds depend on c, for three different norms $||\beta||$. Note that the bounds become tighter for smaller $||\beta||$ and smaller c. Secondly, for small c, the lower bound is nearly optimal.

5 Experiments

5.1 Datasets

We use 9 popular benchmark datasets from UCI repository[2]. To create PU datasets from the completely labelled datasets, the positive examples are selected to be labelled with label frequencies $c = 0.1, 0.2, \ldots, 0.9$. For each label frequency c

[2] https://archive.ics.uci.edu/ml/datasets.php.

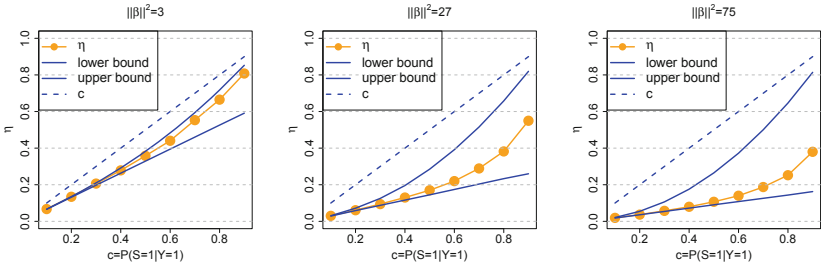

Fig. 1. Shrinkage parameter η wrt c for simulated dataset for $n = 10^6$.

we generated 100 PU datasets labelling randomly elements having $Y = 1$ with probability c and then averaged the results over 100 repetitions.

In addition, we consider one artificial dataset having n observations, generated as follows. Feature vector X was drawn from 3-dimensional standard normal distribution and Y was simulated from (7) with $q(\cdot) = \sigma(\cdot)$, with known $\beta = (1, 1, 1)$. This corresponds to correct specification of the oracle method. The observed variable S was labelled as 1 for elements having $Y = 1$ with probability c. Note however, that in view of discussion in Sect. 4, the naive model is incorrectly specified. Moreover, recall that in this case $\beta = b^* = \arg\min R(b)$. The main advantage of using artificial data is that β (and thus also b^*) is known and thus we can analyse the estimation error for the considered methods. For artificial dataset, we experimented with different values of c and n.

5.2 Methods and Evaluation Measures

The aim of the experiments is to compare the three methods of learning parameters in logistic regression: naive, weighted and joint. Our implementation of the discussed methods is available at https://github.com/teisseyrep/PUlogistic. Our main goal is to investigate how the considered methods relate to the oracle method, corresponding to idealized situation in which we have an access to Y. In view of this, as an evaluation measure we use approximation error for posterior defined as $AE = n^{-1} \sum_{i=1}^{n} |\hat{f}_{\text{oracle}}(x_i) - \hat{f}_{\text{method}}(x_i)|$, where 'method' corresponds to one of the considered methods (naive, weighted or joint), i.e. $\hat{f}_{\text{naive}}(x) := c^{-1}\sigma(x^T \hat{b}_{\text{naive}})$, $\hat{f}_{\text{weighted}}(x_i) := \sigma(x_i^T \hat{b}_{\text{weighted}})$ or $\hat{f}_{\text{joint}}(x_i) := \sigma(x_i^T \hat{b}_{\text{joint}})$. The oracle classifier is defined as $\hat{f}_{\text{oracle}}(x_i) := \sigma(x_i^T \hat{b}_{\text{oracle}})$, where $\hat{b}_{\text{oracle}}$ is minimizer of empirical version of (5). Estimation error for posterior, defined above, measures how accurate we can approximate the oracle classifier when using S instead of true class label Y. We consider two scenarios. In the first one we assume that c is known and we only estimate parameters corresponding to vector X. This setting corresponds to known prior probability $P(Y = 1)$ (c can be estimated accurately when prior is known via equation $c = P(S = 1)/P(Y = 1)$ by plugging-in corresponding fraction for $P(S = 1)$). In the second more realistic scenario, c is unknown and is estimated from data. For joint method we jointly

minimize empirical risk $\widehat{R}_2(b, c)$ with respect to b and c. For two remaining methods (naive and weighted) we use external methods of estimation of c. We employ two methods; the first one was proposed by Elkan and Noto [9] (called EN) is based on averaging predictions of naive classifier over labeled examples for validation data. The second method, described in recent paper [11], is based on optimizing a lower bound of c via top-down decision tree induction (this method will be called TI). In order to analyse prediction performance of the proposed methods, we calculate AUC (Area Under ROC curve) of classifiers based on $\hat{f}_{\text{method}}$ on independent test set.

For artificial datasets, the true parameter β is known so we can analyse mean estimation error defined as $EE = p^{-1} \sum_{j=1}^{p} |\hat{b}_j - \beta_j|$, where $\hat{b}$ corresponds to one of the considered methods. Moreover, we consider an angle between β and $\hat{b}$. In view of property (9) the angle should be small, for sufficiently large sample size. Finally, let us note, that some real datasets may contain large number of features, so to make the estimation procedures more stable, we first performed feature selection. We used filter method recommended in [21] based on mutual information and select top $t = 3, 5, 10$ features for each dataset (we present the results for $t = 5$, the results for other t are similar and are presented in Supplement). This step is common for all considered methods.

5.3 Results

First, we analyse how the approximation errors for posterior depend on c, for real datasets (Fig. 2). We show the results for unknown c, the results for known c are presented in Supplement https://github.com/teisseyrep/PUlogistic. For unknown c, estimation of label frequency plays an important role. We observe that the performance curves vary depending on the method used. For most datasets, TI method outperforms EN, which is consistent with experiments described in [11], an exception is spambase for which TI works poorly. Importantly, joint method is a clear winner for most of the datasets, what suggests that simultaneous estimation of c and b is more effective than performing these two steps separately. Its superiority is frequently quite dramatic (see diabetes, credit-g and spambase). For most datasets, we observe the deterioration in posterior approximation when c becomes smaller. This is concordant with expectations, as for small c, the level of noise in observed variable S increases (cf Eq. (1)) and thus the gap between oracle and naive methods increases.

Tables 1 and 2 show values of AUC, for cases of known and unknown c, respectively. The results are averaged over 100 repetitions. In each repetition, we randomly chose $c \in (0, 1)$, then generate PU dataset and finally split it into training and testing subsets. For naive and weighted methods, c is estimated using TI algorithm (the performance for EN algorithm is generally worse and thus not presented in the Table). The last row contains averaged ranks, the larger the rank for AUC the better. The best method from three (naive, weighted and joint method) is in bold. As expected, the oracle method is an overall winner. The differences between the remaining methods are not very pronounced. Surprisingly, naive and joint methods work in most cases on par, whereas weighted

Table 1. AUC, known c

	Oracle	Joint	Naive	Weighted
Breastc	0.993	0.981	**0.987**	0.974
Diabetes	0.821	0.805	**0.808**	0.805
Heart-c	0.879	0.847	0.849	**0.850**
Credit-a	0.914	0.875	**0.899**	0.891
Credit-g	0.740	0.726	**0.727**	0.725
Adult	0.874	**0.874**	0.869	0.874
Vote	0.973	**0.974**	0.968	0.970
Wdbc	0.987	**0.981**	0.971	0.970
Spambase	0.911	**0.914**	0.892	0.899
Rank	3.8	**2.4**	2.1	1.7

Table 2. AUC (est. c)

Oracle	Joint	Naive	Weighted
0.993	0.983	**0.988**	0.977
0.821	0.798	**0.805**	0.796
0.879	0.843	0.850	**0.853**
0.914	0.889	**0.899**	0.897
0.740	0.724	**0.730**	0.718
0.874	**0.872**	0.869	0.863
0.973	0.972	0.968	**0.977**
0.987	**0.981**	0.969	0.973
0.911	**0.913**	0.893	0.856
3.8	**2.2**	**2.2**	1.8

Table 3. $|c - \hat{c}|$

EN	TI	Joint
0.060	0.064	**0.030**
0.234	0.169	**0.071**
0.138	0.121	**0.043**
0.125	0.130	0.317
0.287	0.261	**0.143**
0.244	0.214	**0.059**
0.044	0.088	**0.024**
0.099	0.068	**0.033**
0.189	0.267	**0.033**
2.4	2.3	**1.2**

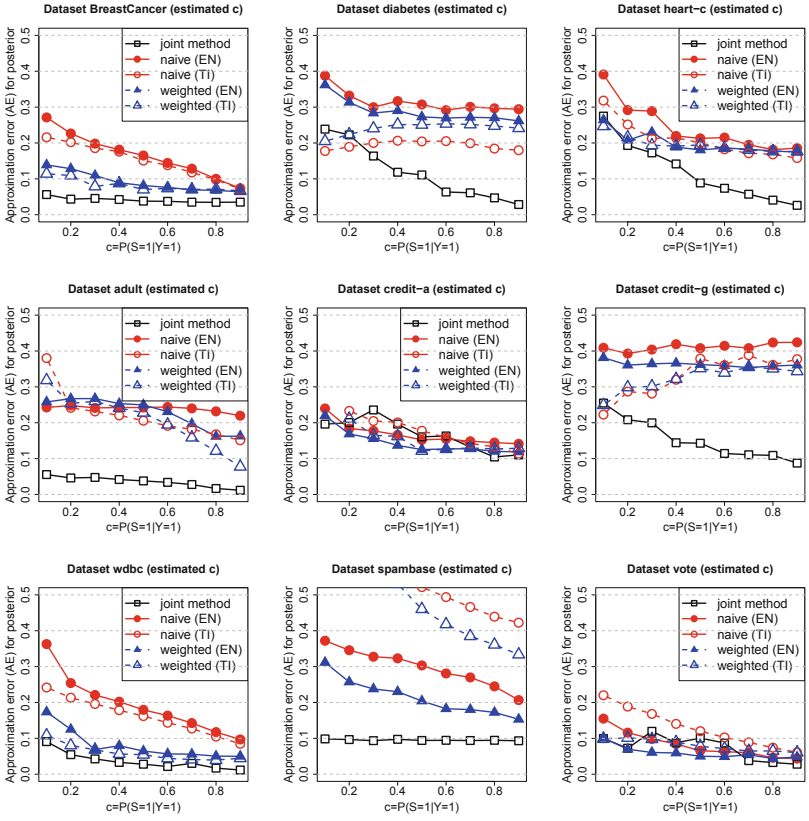

Fig. 2. Approximation error for posterior wrt to c, for estimated c.

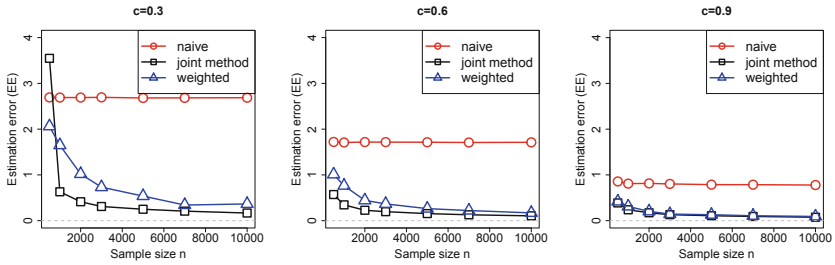

Fig. 3. Mean absolute error $p^{-1}\sum_{j=1}^{p}|\hat{b}_j-\beta|$ wrt to sample size n, where $\hat{b}$ corresponds to one of the methods: naive, weighted and joint method.

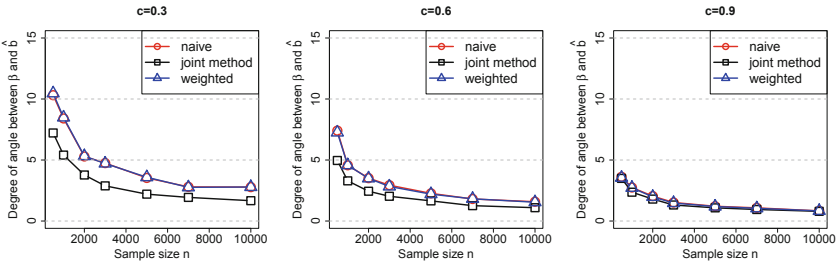

Fig. 4. Degree of angle between β and $\hat{b}$ wrt to sample size n, where $\hat{b}$ corresponds to one of the methods: naive, weighted and joint.

method performs slightly worse. The advantage of joint method is the most pronounced for spambase, for which we also observed superior performance of the joint method wrt approximation error (Fig. 2, bottom panel). Finally, joint method turns out to be effective for estimating c (Table 3)- the estimation errors for joint method are smaller than for TI and EN, for almost all datasets.

Figures 3 and 4 show results for artificial data, for $c = 0.3, 0.6, 0.9$, respectively. Mean estimation error converges to zero with sample size for weighted and joint methods (Fig. 3) and the convergence for joint method is faster. As expected, the estimation error for naive method is much larger than for joint and weighted methods, which is due to incorrect specification of the logistic regression. Note that weighted and joint methods account for wrong specification and therefore both methods perform better. Next we analysed an angle between true β (or equivalently b^*) and $\hat{b}$. Although the naive method does not recover the true signal β, it is able to consistently estimate the direction of β. Indeed the angle for naive method converges to zero with sample size (Fig. 4), which is in line with property (9). Interestingly the speed of converge for weighted method is nearly the same as for naive method, whereas the convergence for joint method is a bit faster.

6 Conclusions

We analysed three different approaches to fitting logistic regression model for PU data. We study theoretically the naive method. Although it does not estimate the true signal β consistently, it is able to consistently estimate the direction of β. This property can be particularly useful in the context of feature selection, where consistent estimation of the direction allows to discover the true significant features - this issue is left for future research. We have shown that under mild assumptions, risk minimizers corresponding to naive and oracle methods are collinear and the collinearity factor η is related to label frequency c. Moreover, we proposed novel method that allows to estimate parameter vector and label frequency c simultaneously. The proposed joint method achieves the smallest approximation error, which indicates that it is the closest to the oracle method among considered methods. Secondly, the joint method, unlike weighted and naive methods, does not require using external procedures to estimate c. Importantly, it outperforms the two existing methods (EN and TI) wrt to estimation error for c. In view of above, joint method can be recommended in practice, especially for estimating posterior probability and c; the differences in AUC for classifiers between the considered methods are not very pronounced.

7 Proofs

Equation (2) follows from

$$\begin{aligned} P(S=1|X=x) &= P(Y\varepsilon=1|X=x) = P(Y=1,\varepsilon=1|X=x) \\ &= P(Y=1|X=x)P(\varepsilon=1|X=x) = P(Y=1|X=x)P(\varepsilon=1) \\ &= P(Y=1|X=x)P(S=1|Y=1). \end{aligned}$$

The third equality follows from conditional independence of Y and ε given X.

To prove (3), note that $P(Y=1|S=0,X)$ can be written as

$$\begin{aligned} &\frac{P(Y=1,\varepsilon=0,X)}{P(S=0,X)} = \frac{P(\varepsilon=0)}{P(\varepsilon=1)}\frac{P(Y=1,X)P(\varepsilon=1)}{P(S=0,X)} \\ &= \frac{P(\varepsilon=0)}{P(\varepsilon=1)}\frac{P(Y=1,\varepsilon=1,X)}{P(S=0,X)}\frac{1-c}{c}\frac{P(S=1,X)}{P(S=0,X)} = \frac{1-c}{c}\frac{P(S=1|X)}{P(S=0|X)}, \end{aligned}$$

where the second to last equality follows from $P(\varepsilon=0)/P(\varepsilon=1) = (1-c)/c$. To prove (4) we write

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

The third equality follows from conditional independence of X and ε given Y.

References

1. Bekker, J., Davis, J.: Learning from positive and unlabeled data: a survey (2018)
2. Sechidis, K., Sperrin, M., Petherick, E.S., Luján, M., Brown, G.: Dealing with under-reported variables: an information theoretic solution. Int. J. Approx. Reason. **85**, 159–177 (2017)
3. Onur, I., Velamuri, M.: The gap between self-reported and objective measures of disease status in India. PLOS ONE **13**(8), 1–18 (2018)
4. Liu, B., Dai, Y., Li, X., Lee, W.S., Yu, P.S.: Building text classifiers using positive and unlabeled examples. In: Proceedings of the Third IEEE International Conference on Data Mining, ICDM 2003, p. 179 (2003)
5. Fung, G.P.C., Yu, J.X., Lu, H., Yu, P.S.: Text classification without negative examples revisit. IEEE Trans. Knowl. Data Eng. **18**(1), 6–20 (2006)
6. Li, X., Liu, B.: Learning to classify texts using positive and unlabeled data. In: Proceedings of the 18th International Joint Conference on Artificial Intelligence, pp. 587–592 (2003)
7. Mordelet, F., Vert, J.-P.: ProDiGe: prioritization of disease genes with multitask machine learning from positive and unlabeled examples. BMC Bioinformatics **12**(1), 389 (2011)
8. Cerulo, L., Elkan, C., Ceccarelli, M.: Learning gene regulatory networks from only positive and unlabeled data. BMC Bioinformatics **11**, 228 (2010)
9. Elkan, C., Noto, K.: Learning classifiers from only positive and unlabeled data. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2008, pp. 213–220 (2008)
10. du Plessis, M.C., Niu, G., Sugiyama, M.: Class-prior estimation for learning from positive and unlabeled data. Mach. Learn. **106**(4), 463–492 (2016). https://doi.org/10.1007/s10994-016-5604-6
11. Bekker, J., Davis, J.: Estimating the class prior in positive and unlabeled data through decision tree induction. In: Proceedings of the 32th AAAI Conference on Artificial Intelligence, February 2018
12. Steinberg, D., Cardell, N.S.: Estimating logistic regression models when the dependent variable has no variance. Commun. Stat. Theory Methods **21**(2), 423–450 (1992)
13. Lancaster, T., Imbens, G.: Case-control studies with contaminated controls. J. Econom. **71**(1), 145–160 (1996)
14. Kiryo, R., Niu, G., du Plessis, M.C., Sugiyama, M.: Positive-unlabeled learning with non-negative risk estimator. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS 2017, pp. 1674–1684 (2017)
15. Denis, F., Gilleron, R., Letouzey, F.: Learning from positive and unlabeled examples. Theoret. Comput. Sci. **348**(1), 70–83 (2005)
16. Chapelle, O., Schölkopf, B., Zien, A.: Semi-Supervised Learning. The MIT Press, Cambridge (2010)
17. Candès, E., Fan, Y., Janson, L., Lv, J.: Panning for gold: model-x knockoffs for high-dimensional controlled variable selection. Manuscript (2018)
18. Gottschalk, P.G., Dunn, J.R.: The five-parameter logistic: a characterization and comparison with the four-parameter logistic. Anal. Biochem. **343**(1), 54–65 (2005)

19. Mielniczuk, J., Teisseyre, P.: What do we choose when we err? Model selection and testing for misspecified logistic regression revisited. In: Matwin, S., Mielniczuk, J. (eds.) Challenges in Computational Statistics and Data Mining. SCI, vol. 605, pp. 271–296. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-18781-5_15
20. Kubkowski, M., Mielniczuk, J.: Active set of predictors for misspecified logistic regression. Statistics **51**, 1023–1045 (2017)
21. Sechidis, K., Brown, G.: Simple strategies for semi-supervised feature selection. Mach. Learn. **107**(2), 357–395 (2017). https://doi.org/10.1007/s10994-017-5648-2