




Bayesian Logistic Model for Positive and Unlabeled Data

Małgorzata Łazęcka^{1,2}(✉) 

¹ Faculty of Mathematics, Informatics and Mechanics University of Warsaw,
Banacha 2, 02-097 Warsaw, Poland

² Institute of Computer Science Polish Academy of Sciences, Jana Kazimierza 5,
01-248 Warsaw, Poland
`malgorzata.lazeczka@ipipan.waw.pl`

Abstract. In the paper, we introduce a novel method of estimating label frequency and parameters of the logistic model for positive and unlabeled (PU) data. Our approach is based on Gibbs sampler that uses Pólya-Gamma latent variables for Bayesian logistic model. In the paper, we focus on estimating label frequency, but the proposed method also provides estimated probabilities of being positive observation among the unlabeled ones.

Keywords: positive and unlabeled data · Selected Completely At Random · Bayesian logistic regression · Gibbs sampling · graphical model

1 Introduction

In standard binary classification, the data consists of positive and negative examples. However, in many applications, the assumption that the class is known for all observations might not be realistic. Consider e.g. medical data, in which usually one has information about patients with the diagnosed disease and the rest of patients might either be healthy or have a disease and remain undiagnosed. In positive and unlabeled (PU) learning we model that situation by assuming that we have access to some positive examples (diagnosed patients), and we do not know the true class of the others (undiagnosed) - they might be either positive or negative. Another example is a survey with sensitive questions e.g. about illegal behavior. Some people who broke the law would answer to that question truthfully, but among those, who would answer “no”, there might be a group that actually broke the law but would not admit that in the survey. The next example considers advertisements e.g. the ads that appear on the visited websites. Positive and labeled examples in that scenario are clicks on the ads. However, the remaining ads also might be interesting to the user even though the user has not clicked on them. In the paper we focus on estimating the frequency of such events that the user clicks on the ad, thus we estimate how many of the positive examples are labeled. As a by-product of the proposed method, we obtain probabilities indicating which of the unlabeled observations might be positive.

The main contribution of this paper is to adapt the model introduced in [8] in such a way that it can be applied to PU data. In [8] the authors propose a framework for Bayesian inference for the logistic model using the idea of data augmentation. Since the vector of classes is not observed in PU setting, the model proposed in [8] cannot be used directly, thus we extend it with new variables so the new model can cope with data censored as described above. That approach allows for estimating label frequency with high accuracy.

1.1 Notation and Assumptions in PU Learning

In PU learning, we consider a triple of variables (X, Y, S) , where $X \in \mathbb{R}^p$ is a random variable corresponding to a feature vector, $Y \in \{0, 1\}$ is a true class and $S \in \{0, 1\}$ is an indicator, whether the observation is positive or unlabeled. In PU setting all labeled observations are positive. In this article, we consider single-case scenario, in which we assume that there is a common distribution of (X, Y, S) and the sample $(x_i, y_i, s_i)_{i=1}^n$ consists of independently drawn observations from that distribution. In standard classification the available data is $(x_i, y_i)_{i=1}^n$, whereas in PU learning we observe only $(x_i, s_i)_{i=1}^n$. Notice that some values of the vector $y = (y_1, y_2, \dots, y_n)$ are known as when $s_i = 1$ then $y_i = 1$, but when $s_i = 0$ then y_i can be either 0 or 1.

A common assumption in PU learning is the *Selected Completely At Random* (SCAR) assumption, which states that the labeled examples are selected randomly from a set of positive examples independently of X , i.e.

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

Constant $c := P(S = 1|Y = 1)$ is called *label frequency*. Note that the condition (1) is equivalent to conditional independence of S and X given Y . A common task in PU learning under SCAR assumption is estimation of the parameter c and in this paper we also focus on that problem. We briefly describe some of the existing methods of estimation of c in Sect. 1.3.

1.2 Logistic Model Assumption for PU Data

In logistic model, in which we observe a class indicator Y , we assume that probability of the event $Y = 1$ is logit function in a linear combination of the variables X , namely

$$P(Y = 1|X = x) = \frac{e^{x'\beta}}{1 + e^{x'\beta}} =: \sigma(x'\beta), \quad (2)$$

where $\sigma(t) = e^t/(1 + e^t)$ is a standard logistic function and a symbol $'$ denotes transposition. In PU learning assuming SCAR we have

$$P(S = 1|X = x) = cP(Y = 1|X = x) \quad (3)$$

as LHS equals

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

Hence, using (2) we obtain that

$$P(S = 1|X = x) = c \times \sigma(x'\beta), \quad (4)$$

where c is label frequency. The model for PU data has an additional parameter c in comparison to the standard logistic regression. The parameters (c, β) are identifiable in view of Theorem 1 in [12], which is not true for c in general setting for PU learning without some additional assumptions. In the proposed method we use an assumption that (Y, X) follow the logistic model as in (2). Other methods of estimation of parameters (c, β) are discussed in Sect. 1.3.

1.3 Methods of Label Frequency Estimation

In this section, we introduce methods of label frequency estimation, some of which will be used in Sect. 3.2. For a comprehensive survey, we refer to [6].

Elkan-Noto and Tlce Estimator. In a method proposed by Elkan and Noto [4] we divide the dataset into two subsets: a training set, on which the classifier $\hat{P}(S = 1|x)$ is trained and a validation set used to compute an estimator of c . The estimator of c is defined as

$$\hat{c}_{EN} = \frac{1}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} \hat{P}(S = 1|X_i),$$

where \mathcal{A} is a set of indices of observations in the validation set that are labeled. The method uses the fact, that

$$c = \frac{P(S = 1|X = x)}{P(Y = 1|X = x)}, \quad (5)$$

and thus if the classes are separable and we compute the denominator for a labeled example, then it equals 1. The method introduced in [1] is based on similar observation, namely that

$$\begin{aligned} c &= P(S = 1|Y = 1) = P(S = 1|Y = 1, X \in \mathcal{A}) \\ &= \frac{P(S = 1, Y = 1, X \in \mathcal{A})}{P(Y = 1, X \in \mathcal{A})} = \frac{P(S = 1|X \in \mathcal{A})}{P(Y = 1|X \in \mathcal{A})}. \end{aligned}$$

Next, we look for a so-called anchor set \mathcal{A} , for which $P(Y = 1|X \in \mathcal{A}) \approx 1$ using induction trees on a training set and on a test set we estimate $P(S = 1|X \in \mathcal{A})$.

KM Estimators. The estimators proposed in [10] are based on representing the distribution of unlabeled observations as a mixture of the distributions corresponding to $S = 0, Y = 1$ and $S = 0, Y = 0$. In [10] the authors estimate mixing proportion of the latter two distributions. Then, after the mixing proportion is estimated, the class frequency $P(Y = 1)$ can be easily computed, and as $P(Y = 1) = P(S = 1)/c$ we also obtain the estimator of c . Two ways of estimating mixing proportion are proposed, thus two estimators \hat{c}_{KM1} and \hat{c}_{KM2} are obtained.

JOINT and CD+MM Estimators. In view of (4), in order to obtain estimators of (c, β) the following log-likelihood function is maximised

$$l(c, b) = \sum_{i=1}^n (s_i \log(c\sigma(x'_i\beta)) + (1 - s_i) \log(1 - c\sigma(x'_i\beta)))$$

with respect to c and β simultaneously. JOINT method [11] optimize $l(c, b)$ using simple gradient algorithm. CD+MM accounts for the fact that $l(c, b)$ is not a concave function and thus it may have multiple local minima. CD+MM algorithm [12] consists of two steps in each iteration i : first, using the fact that $l(c, b)$ is concave with respect to c , finds a maximizer \hat{c}_i of $l(c, \hat{b}_{i-1})$. Next, using Minorization-Maximization algorithm (see [7]) maximizes $l(\hat{c}_i, b)$ with respect to b . The optimization algorithm is run until it converges to the local minimum.

MLR Estimator. [5] Note that from (5) it follows that $c \leq \max_x P(S = 1|X = x)$ and if $\max_x P(Y = 1|X = x) = 1$, then we obtain equality. In MLR the following model is fitted

$$g(x, b, \gamma) = \frac{1}{1 + b^2 + \exp(\gamma'x)},$$

where $b > 0$ and $\gamma \in \mathbb{R}^p$. By noting that c can be estimated as $\max_x \hat{P}(S = 1|X = x)$ and $\max_x g(x, b, \gamma) = \frac{1}{1+b^2}$, we obtain $\hat{c} = \frac{1}{1+\hat{b}^2}$.

2 Gibbs Sampler for Estimation of Label Frequency

First, we give a brief description of Gibbs sampler and Bayesian logistic regression introduced in [8], and then in Sect. 2.3 we present our adaptation to PU setting.

2.1 Gibbs Sampling

Gibbs sampling is a Markov chain Monte Carlo algorithm for obtaining a sequence of observations from a multivariate joint probability distribution. The algorithm is especially useful in the cases when direct sampling from the joint distribution of variables (X_1, X_2, \dots, X_p) is difficult, whereas sampling from conditional distributions $X_i|X_{-i} = x_{-i}$, where $X_{-i} = (X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_p)$ is relatively simple. The output of the algorithm, among other applications, can be used to approximate marginal distribution of a chosen subset of variables or to compute their expected value. The variables (X_1, X_2, \dots, X_p) might represent latent variables of the model we want to sample from or parameters in Bayesian approach. Below we give a brief description of the algorithm.

Suppose we want to sample from the distribution $p(x_1, x_2, \dots, x_p)$ and sampling from conditional distribution $p(x_j|x_{-j})$ for $j = 1, 2, \dots, p$ is feasible. Then to obtain N observations from the distribution of $p(x_1, x_2, \dots, x_p)$, one can proceed as follows:

- (1) Set $x^{(0)} = (x_1^{(0)}, x_2^{(0)}, \dots, x_p^{(0)})$ to a starting value.
- (2) Sample $x_j^{(i)} \sim p\left(x_j | x_1^{(i)}, \dots, x_{j-1}^{(i)}, x_{j+1}^{(i-1)}, \dots, x_p^{(i-1)}\right)$ for $j = 1, 2, \dots, p$.

Repeat (2) for $i = 1, 2, \dots, N$, where N is a number of samples required.

Ideally, the initial value $x^{(0)}$ should be chosen from a region of high probability $p(x_1, x_2, \dots, x_p)$, but as it is difficult, it is common to sample $N + B$ samples instead of N and discard B samples from the beginning.

Gibbs sampling is frequently used in Bayesian inference. In that approach, prior distribution $\pi(\theta)$ of the vector of parameters $\theta = (\theta_1, \theta_2, \dots, \theta_p)$ is given and we assume that observations y come from the distribution $p(y|\theta)$, where p is known. In this case, the aim is to sample from the posterior distribution $p(\theta|y)$, as we are interested in the distribution of the parameters given the information about θ from the observed sample. In each step i of (2) of the Gibbs sampler we sample from the distribution $p(\theta_j | \theta_1^{(i)}, \dots, \theta_{j-1}^{(i)}, \theta_{j+1}^{(i-1)}, \theta_p^{(i-1)}, y)$ for $j \in 1, 2, \dots, p$.

2.2 Gibbs Sampler for Bayesian Logistic Regression

We describe now an algorithm introduced in [8] for sampling from the posterior distribution of the parameters β from the logistic model (cf. (2))

$$P(Y = y|\beta) = p(y|\beta) = \frac{(e^{x'\beta})^y}{1 + e^{x'\beta}}.$$

Gibbs sampler given in [8] uses latent variables following Pólya-Gamma distribution to enable efficient sampling from conditional distributions. The densities of distributions in Pólya-Gamma family $PG(1, a)$ with parameter $a > 0$ are defined as

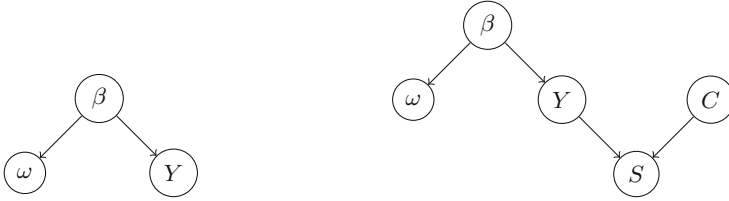
$$f(x|a) = \cosh(a/2)e^{-\frac{a^2x}{2}}g(x),$$

where g is a density of an infinite sum of properly scaled i.i.d. exponential variables (the definition of the density g is given in [8], p. 1340). We do not provide the formula for g , as in the following only the terms containing a will be used. For details see [2, 8].

To construct a Gibbs sampler for Bayesian logistic regression, latent variables ω are used, thus we estimate (β, ω) . The step (2) of Gibbs algorithm has two sub-steps:

- Sample $\omega^{(i)} \sim p(\omega|\beta^{(i-1)}, y)$,
- Sample $\beta^{(i)} \sim p(\beta|\omega^{(i)}, y)$.

Using notation of the previous section we have $(\theta_1, \theta_2) = (\beta, \omega)$ (the first parameter is a p -dimensional vector, where p denotes the number of predictors and the second parameter is n -dimensional, where n denotes the number of observations) and y is observed. The dependence structure of variables (β, ω, Y) is represented by probabilistic graphical model shown in Fig. 1a, in which the vertices denote random variables and the orientation of the edges determines the direction of



(a) Graphical model for Bayesian logistic sampler (b) Graphical model for Bayesian logistic sampler for PU data

Fig. 1. Graphical models indicating dependence structure of the considered variables

dependence. The joint distribution corresponding to a graphical model is the product of the conditional probabilities for every node given its parents, thus the joint distribution of (β, ω, Y) factorizes in the following way

$$p(\beta, \omega, Y) = \pi(\beta)p(\omega|\beta)p(y|\beta). \tag{6}$$

Note that from (6) it follows that ω and Y are independent given β . We also assume that observations $(\omega_i, Y_i)_{i=1}^n$ are independent given β , hence we have $p(\omega_i|\beta, y) = p(\omega_i|\beta)$ and $p(\omega|\beta) = \prod_{i=1}^n p(\omega_i|\beta)$. Moreover for a given β the distribution of ω_i is $PG(1, |x'_i\beta|)$ and the prior for β is $\mathcal{N}(b_\beta, B_\beta)$, where b_β and B_β are fixed and we show that conditional distribution of β is also normal. In the following, we will use \propto to denote equality up to multiplication by a constant. We have

$$\begin{aligned} p(\beta|\omega, y) &\propto p(\beta, \omega, y) = \pi(\beta)p(\omega|\beta)p(y|\beta) \\ &= \pi(\beta) \prod_{i=1}^n \left(\cosh\left(\frac{|x'_i\beta|}{2}\right) e^{-\frac{(x'_i\beta)^2\omega_i}{2}} g(\omega_i) \right) \prod_{i=1}^n \left(\frac{(e^{x'_i\beta})^{y_i}}{1 + e^{x'_i\beta}} \right) \\ &\propto 2^{-n} \pi(\beta) \prod_{i=1}^n \exp\left(y_i x'_i\beta - \frac{x'_i\beta}{2} - \frac{\omega_i(x'_i\beta)^2}{2}\right) \\ &\propto \pi(\beta) \prod_{i=1}^n \exp\left(-\frac{\omega_i}{2} \left(x'_i\beta - \frac{y_i - 1/2}{\omega_i}\right)^2\right) \end{aligned}$$

where in the third expression we omitted the terms $g(\omega_i)$, as they do not depend on β and we used the fact that $\cosh(x) = \frac{e^x + e^{-x}}{2}$. Thus, after further transformations, we obtain that conditional distribution of β is $N(\mu(\omega), \Sigma(\omega))$, where $\Sigma(\omega) = (X'\Omega(\omega)X + B_\beta^{-1})^{-1}$, $\mu(\omega) = \Sigma(\omega)(X'(y - \frac{1}{2}\mathbb{1}_n) + B_\beta^{-1}b_\beta)$, $\Omega(\omega) = \text{diag}(\omega)$ and $\mathbb{1}_n$ is a n -dimensional vector of 1.

2.3 Gibbs Sampler for PU Data

In PU learning the true class Y is not always observed, thus the procedure needs to be modified. We use two additional variables in the model for PU

data: observed vector of labels S and unobserved label frequency C treated as a random variable. We assume that the dependency structure is defined by Fig. 1b, thus the joint density of all considered in the model variables can be factorized in the following way (cf. (6))

$$p(\omega, \beta, y, s, c) = \pi(\beta)p(\omega|\beta)p(y|\beta)\pi(c)p(s|y, c). \quad (7)$$

We now compute the conditional distributions of all the variables, which will be sampled, given the remaining ones. The conditional distributions for β and ω are described in Sect. 2. Below we compute conditional densities $p(y|\beta, \omega, c, s)$ and $p(c|\beta, \omega, y, s)$.

Note, that from (7) it easily follows that $p(y|\beta, \omega, c, s) = p(y|\beta, c, s)$. We also have

$$P(Y_i = y_i | S = s, \beta = b, C = c) \propto P(Y_i = y_i | \beta = b)P(S_i = s | Y = y, C = c). \quad (8)$$

We assume that variables (Y, X) satisfy (2). On the other hand, from SCAR assumption it follows that $P(S_i = 1 | Y_i = 1, C = c) = 1 - P(S_i = 0 | Y_i = 1, C = c) = c$ and if $Y_i = 0$, we have $P(S_i = 0 | Y_i = 0, C = c) = P(S_i = 0 | Y_i = 0) = 1$. Hence for $S_i = 1$ we obtain

$$\begin{aligned} P(Y_i = 1 | S_i = 1, \beta, C) &\propto c \times \sigma(x'_i \beta), \\ P(Y_i = 0 | S_i = 1, \beta, C) &= 0, \end{aligned} \quad (9)$$

and for $S_i = 0$ we have

$$\begin{aligned} P(Y_i = 1 | S_i = 0, \beta, C) &\propto (1 - c)\sigma(x'_i \beta), \\ P(Y_i = 0 | S_i = 0, \beta, C) &\propto 1 - \sigma(x'_i \beta). \end{aligned} \quad (10)$$

Equations (9) and (10) lead to

$$P(Y_i = 1 | S_i = s, \beta = b, C = c) = \frac{(1 - c)\sigma(x'_i \beta)}{(1 - c)\sigma(x'_i \beta) + (1 - s)(1 - \sigma(x'_i \beta))}. \quad (11)$$

Now we derive the formula for $p(c|\beta, \omega, y, s)$. Prior density $\pi(c)$ of C is $Beta(\alpha_c, \beta_c)$. From (7) we obtain that C is independent of β and ω given S and Y . Thus we consider conditional distribution of C given only S and Y

$$p(c|S_i = s, Y_i = 1) \propto \pi(c)P(S_i = s | Y_i = 1, C = c).$$

Hence, assuming that the pairs $(S_i, Y_i)_{i=1}^n$ are independent given C , we obtain

$$p(c|S_i, Y_i = 1, i = 1, \dots, n) \propto \pi(c)c^{\sum_{i=1}^n \mathbb{I}(S_i=1, Y_i=1)}(1 - c)^{\sum_{i=1}^n \mathbb{I}(S_i=0, Y_i=1)},$$

thus $C|S, Y \sim Beta(\alpha_c + \sum_{i=1}^n \mathbb{I}(S_i = 1, Y_i = 1), \beta_c + \sum_{i=1}^n \mathbb{I}(S_i = 0, Y_i = 1))$. Note that proposed prior distribution of C is conjugate for the likelihood which is Bernoulli distribution (for success being $S_i = 1, Y_i = 1$ and the failure $S_i = 0, Y_i = 1$). Thus the posterior is also Beta distribution with modified parameters according to the data.

Below we summarise the above derivations. We use the following prior distributions for β and C with hyperparameters (b_β, B_β) and (α_c, β_c)

$$\beta \sim \mathcal{N}(b_\beta, B_\beta), \quad C \sim \text{Beta}(\alpha_c, \beta_c).$$

Then in step (2) of the Gibbs sampler we sample from the following distributions (the definitions of $\mu(\omega)$ and $\Sigma(\omega)$ are given at the end of Sect. 2.2):

$$\begin{aligned} \omega_i | \beta &\sim PG(1, |x'_i \beta|) \text{ for } i = 1, 2, \dots, n, \\ Y_i | \beta, S, C &\sim \text{Bern} \left(\frac{(1-C)\sigma(x'_i \beta)}{(1-C)\sigma(x'_i \beta) + (1-S_i)(1-\sigma(x'_i \beta))} \right) \text{ for } i = 1, 2, \dots, n, \\ \beta | \omega, Y &\sim \mathcal{N}(\mu(\omega), \Sigma(\omega)), \\ C | Y, S &\sim \text{Beta} \left(\alpha_c + \sum_{i=1}^n \mathbb{I}(S_i = 1, Y_i = 1), \beta_c + \sum_{i=1}^n \mathbb{I}(S_i = 0, Y_i = 1) \right). \end{aligned}$$

Algorithm 1. One step of Gibbs sampler for PU data

Input: $X, S, b_\beta, B_\beta, \beta_{\text{old}}, \alpha_c, \beta_c, c_{\text{old}}$

Output: $\beta_{\text{new}}, c_{\text{new}}$

- 1: **for** $i \in \{1, 2, \dots, n\}$ **do**
 - 2: $\omega_{i,\text{new}} \leftarrow$ a sample from $PG(1, |X'_i \beta_{\text{old}}|)$
 - 3: $\sigma_i \leftarrow \sigma(X'_i \beta_{\text{old}})$
 - 4: $p_{Y_i=1} \leftarrow (1 - c_{\text{old}})\sigma_i / [(1 - c_{\text{old}})\sigma_i + (1 - S_i)(1 - \sigma_i)]$
 - 5: $y_{i,\text{new}} \leftarrow$ a sample from $\text{Bern}(p_{Y_i=1})$
 - 6: **end for**
 - 7: $\Omega_{\text{new}} \leftarrow \text{diag}(\omega_{\text{new}})$
 - 8: $\Sigma_\beta \leftarrow (X' \Omega_{\text{new}} X + B_\beta^{-1})^{-1}$
 - 9: $\mu_\beta \leftarrow \Sigma_\beta (X' (y_{\text{new}} - \frac{1}{2} \mathbb{1}_n) + B_\beta^{-1} b_\beta)$
 - 10: $\beta_{\text{new}} \leftarrow$ a sample from $\mathcal{N}_p(\mu_\beta, \Sigma_\beta)$
 - 11: $n_{\text{pl}} \leftarrow \sum_{i=1}^n \mathbb{I}(S_i = 1, y_{i,\text{new}} = 1)$
 - 12: $n_{\text{pu}} \leftarrow \sum_{i=1}^n \mathbb{I}(S_i = 0, y_{i,\text{new}} = 1)$
 - 13: $c_{\text{new}} \leftarrow$ a sample from $\text{Beta}(\alpha_c + n_{\text{pl}}, \beta_c + n_{\text{pu}})$
-

3 Numerical Experiments

In this section, we first present an illustrative example showing how the proposed method works. Next, we briefly describe methods of label frequency estimation existing in the literature and at the end we compare the accuracy of our method with other methods on real datasets. The R code is available on Github¹.

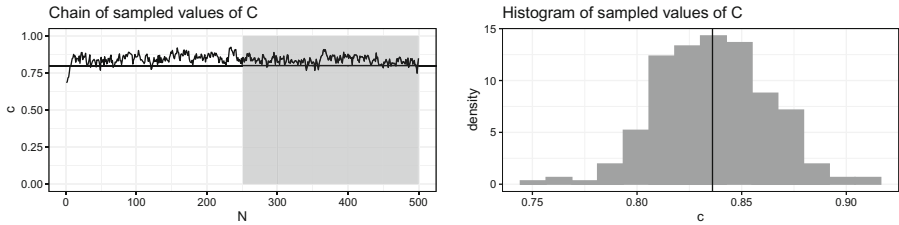
In Algorithm 1 we present one step of Gibbs sampler for PU learning. The input consists of a matrix of predictors X , a vector of labels S , hyperparameters

¹ github.com/lazeckam/PU_BayesLogistic.

of normal distribution b_β and B_β , hyperparameters of Beta distribution α_c and β_c and initial values or values from the previous step of β and c , namely β_{old} and c_{old} .

3.1 Example

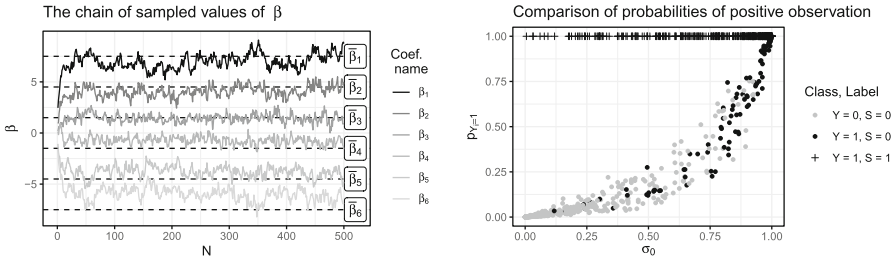
Let $X_i = (X_{i1}, X_{i2}, \dots, X_{ip})$ for $i \in \{1, 2, \dots, n\}$. We sample observations $X_{i,j}$ independently from uniform $U([0, 1])$ distribution for $i \in \{1, 2, \dots, n\}$, $j \in \{1, 2, \dots, p\}$ and let $n = 1000$, $p = 6$. For each row $X_{i,\cdot} = x_i$ of the matrix X we sample Y_i according to the distribution $P(Y_i = 1 | \bar{\beta}, x_i) = \sigma(x_i' \bar{\beta}) =: \sigma_{0,i}$, where $\bar{\beta} = (7.5, 4.5, 1.5, -1.5, -4.5, -7.5)$ with intercept being 0. We fix $c = 0.8$ and sample S_i according to Bernoulli distribution with probability of success c for positive observations ($Y_i = 1$) and for the remaining ones $S_i = 0$. To run the simulations we use the following hyperparameters and initial values: $b_\beta = 0_p$, $B_\beta = 10 \cdot I_p$, $\alpha_c = 1$, $\beta_c = 1$ and $\beta_{start} = 0_p$, $c_{start} = 0.5$, where 0_p denotes a vector of p zeros and I_p is a $p \times p$ identity matrix. Next, we repeat $B + N = 500$ times the step of the Gibbs sampler algorithm described in Algorithm 1.



(a) The chain of sampled values of the variable C . The true value is marked with the horizontal line, the gray background shows which values are used to compute the estimator \hat{c} . (b) The histogram of the values of C for iterations 251-500, which approximates marginal distribution of C and the point estimate $\hat{c} \approx 0.836$ marked with vertical line.

Fig. 2. Estimation of label frequency c .

The obtained chains of values of C and β are shown in Figs. 2a and 3a. We obtain a point estimate of c by discarding the first $B = 250$ values and averaging the remaining ones. In Fig. 2b the histogram of the estimator \hat{c} is presented for the last 250 samples. Figure 3b shows scatterplot of estimated posterior values of probability of $Y = 1$, where posterior distribution of $Y_i | \beta, C, S$ follows $Bern\left(\frac{(1-C)\sigma(x_i'\beta)}{(1-C)\sigma(x_i'\beta) + (1-S_i)(1-\sigma(x_i'\beta))}\right)$, which corresponds to $p_{Y_i=1}$ from line 4 in Algorithm 1 (the values from the 500th iteration of Algorithm 1 are used) against $\sigma_{0,i} = \sigma(x_i'\beta)$ values for $i \in \{1, 2, \dots, n\}$. Both Fig. 3b and Fig. 3a indicating accurate estimation of β show, that the unlabeled observations with high probability of being positive can be detected based on the proposed method as they have also high values of $p_{Y_i=1}$.



(a) The chains of sampled values for vector β compared with true values $\bar{\beta}$ marked with vertical lines. (b) Comparison of true probabilities of $Y = 1$ and values of the posterior probabilities of $Y = 1$ for the last iteration of Algorithm 1.

Fig. 3. The chains of β parameters and posterior probabilities of positive class obtained by the proposed method.

3.2 Real Data Simulations

In this section, we artificially created PU datasets using the labeled benchmark 11 datasets from UCI Machine Learning Repository [3] and one from the IJCNN 2001 competition [9]. Detailed information about datasets is in Table 1, in which the number of observations and predictors is given as well as fraction of positive observations α .

Table 1. Information about datasets

Dataset	n	p	α	Dataset	n	p	α
BreastCancer	683	9	0.35	pop_failure	540	18	0.91
diabetes	768	8	0.35	SPECTF	79	44	0.49
heart-c	303	19	0.46	vote	435	32	0.39
ijcnn2001	35000	22	0.10	wdbc	569	31	0.37
mushroom	8124	21	0.48	Wholesale	440	7	0.32
parkinsons	195	22	0.75	wpbc	198	33	0.24

We run simulations to compare the proposed method (**PGPU**) with the existing ones listed in Sect. 1.3. Due to the lack of space, we present the results only for some of the methods. Extended results are available on Github. For each dataset, we select positive examples to be labeled with probability $c = 0.1, 0.2, \dots, 0.9$ and for each c we repeat the experiment 100 times. All predictors are scaled to $[0, 1]$ suggested in [1]. Due to the computational costs of KM methods, for large dataset *ijcnn2001* we subsampled the original dataset 5 times to obtain $n = 2000$ and we averaged results obtained on the subsamples. In our method, we use the same hyperparameters and initial values as in

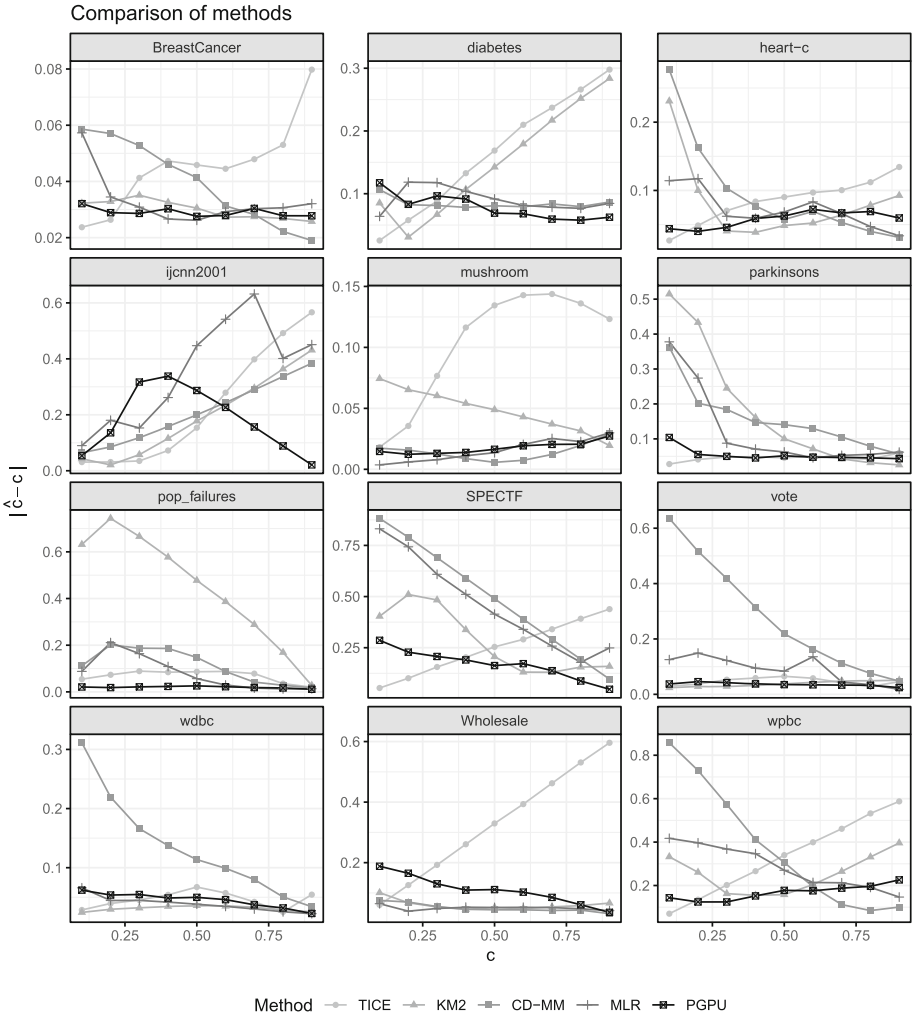


Fig. 4. Comparison of label frequency estimation methods.

the example from Sect. 3.1. For *ijcnn2001* PGPU uses the same subsampling approach as for KM described above. PGPU is also computationally expensive for large datasets as in each iteration we generate n samples from Pólya-Gamma distribution and we take an inverse of $p \times p$ matrix to obtain Σ_β .

Figure 4 shows the results of the experiments. Each point on the plot is an average of 100 results of $|\hat{c} - c|$ for fixed method, dataset and label frequency c . The proposed method for all datasets except for *ijcnn2001* and *Wholesale* outperforms or is as accurate as other methods for almost all c values in terms of the accuracy of c estimation. In the cases, when another method performs better

for some limited range of c , then it works significantly worse for other values of the label frequency (see e.g. `diabetes` and compare PGPU with KM2 for $c = 0.2$ and $c = 0.9$). We stress that achieving small errors over whole range $c \in [0, 1]$ is particularly important in that task and PGPU meets that requirement. PGPU fails this criterion only on `Wholesale` and `ijcnn2001` for small c values, but we note that PGPU might perform better for a different choice of parameters.

4 Conclusions

We establish that the proposed method based on a simple graphical model and Gibbs sampler works well in comparison to other methods. Parametric assumption on the distribution of (Y, X) makes it possible to detect positive and unlabeled observations. Using more elaborate graphical model the method can be naturally extended to situations when the SCAR assumption fails. This is a subject of ongoing research. The method also will be further developed to be feasible for large datasets.

References

1. 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
2. Choi, H.M., Hobert, J.P.: The Pólya-Gamma Gibbs sampler for Bayesian logistic regression is uniformly ergodic. *Electron. J. Statist.* **7**, 2054–2064 (2013)
3. Dua, D., Graff, C.: UCI machine learning repository (2017). <http://archive.ics.uci.edu/ml>
4. 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)
5. Jaskie, K., Elkan, C., Spanias, A.: A modified logistic regression for positive and unlabeled learning. In: 53rd Asilomar Conference on Signals, Systems, and Computers, pp. 2007–2011 (2020)
6. Jaskie, K., Spanias, A.: Positive and unlabeled learning algorithms and applications: a survey. In: IEEE IISA, Patras, Greece, July 2019, pp. 1–8 (2019)
7. Lange, K.: Numerical Analysis for Statisticians. Springer Verlag New-York (2010). <https://doi.org/10.1007/978-1-4419-5945-4>
8. Polson, N.G., Scott, J.G., Windle, J.: Bayesian inference for logistic models using Pólya-Gamma latent variables. *J. Am. Stat. Assoc.* **108**(504), 1339–1349 (2013)
9. Prokhorov, D.: IJCNN 2001 neural network competition. Slide presentation in IJCNN 2001, Ford Research Laboratory (2001)
10. Ramaswamy, H., Scott, C., Tewari, A.: Mixture proportion estimation via kernel embeddings of distributions. In: Proceedings of The 33rd International Conference on Machine Learning, vol. 48, pp. 2052–2060 (2016)
11. Teisseyre, P., Mielniczuk, J., Łazęcka, M.: Different strategies of fitting logistic regression for positive and unlabelled data. In: Proceedings of the International Conference on Computational Science. ICCS 2020 (2020)
12. Łazęcka, M., Mielniczuk, J., Teisseyre, P.: Estimating the class prior for positive and unlabelled data via logistic regression. *Adv. Data Anal. Class.* **15**(4), 1039–1068 (2021). <https://doi.org/10.1007/s11634-021-00444-9>