Surrogate Assisted Positive-Unlabelled Learning

Nigel Petersen
University of Toronto

Abstract

In many Machine Learning applications, obtaining a fully labelled set of training data is often infeasible and labour-intensive, suggesting a need to generalize existing techniques to learn from data in a more restricted setting. Many methods in Semi-Supervised Learning aim to solve this problem, and in the particular case of binary classification, Positive-Unlabelled Learning is a sub-discipline that aims to solve this problem in a setting in which only positively labelled outcomes are observed, a setting which is especially present in Health Care, with Electronic Health Record Data. We propose a Positive-Unlabelled Learning method making use of a single continuous surrogate feature, under the assumption of conditional independence of the remaining features given the response. Our proposed method performs very well in comparison to benchmark methods in simulation studies under a correctly specified model, and shows potential for robustness in a setting where independence assumptions are violated.

1 Introduction

Over the last several decades, improvements in Machine Learning methods have made a significant impact on many aspects of our society, from finance to healthcare and even entertainment. As Machine Learning continues to grow in popularity, new problems in many adjacent disciplines continue to present themselves, and Machine learning is becoming a more popular first choice for tackling them. In particular, Health Care is one such field that has enjoyed many of the advancements in Machine Learning over the past several decades, particularly as a result of the introduction of Electronic Health Record (EHR) data. With Health Care being one of the most integral parts of our society, advancements in the Health Care system are among the most crucial. Unfortunately, many techniques in Supervised Learning fail at the hands of Health Care because of the difficulty of obtaining a sufficiently large sample of well-labelled data. The difficulty of this is present in other applications as well, all encompassed in the area on Semi-Supervised Learning (SSL), a branch of Supervised Learning in which a fully labelled data set is not available, and learning must be done on one where only a subset of observations have labels. A particularly important context within SSL is that of Positive-Unlabelled (PU) Learning, where the outcome of interest is binary, and in addition to only having a subset of labelled observations, only labels in the positive class are present. Many applications in Health Care, particularly disease prediction/phenotyping, can be phrased in a PU Learning framework, but applications in other disciplines, such as personalized advertising and recommender systems can also be thought of in a PU Learning setting. Over the last two decades, there have been several methods and approaches introduced to solve problems in PU Learning, and particularly in the Health Care setting, several have made use of Anchor and Surrogate variables. Surrogate variables are variables known to be highly predictive of the response, and can potentially be used as an auxiliary response when fully labelled data is difficult or labour-intensive to obtain. In such settings, we seek to bridge the gap between the responses we do not have access to, and the data that we observe, using surrogate variables.

2 Materials and Methods

Traditionally, we consider triples of the form $(\mathbf{X}, \mathbf{S}, Y)$ where $\mathbf{X} \in \mathbb{R}^p$ is a vector of features, $\mathbf{S} \in \mathbb{R}^q$ is a vector of surrogate features, and $Y \in \{0, 1\}$ is the binary response. In the case that $q = 1$, we say that S is an anchor variable, and if $S \in \{0, 1\}$, a binary anchor variable. There are a number of existing methods that make use of surrogate and anchor variables, here we introduce the Maximum Likelihood (ML) algorithm [3] making use of binary anchors and an automated feature selection algorithm [1] making use of surrogate features.

2.1 Prior Methods

2.1.1 ML Algorithm

Assume observations consist of triples $(\mathbf{X}, S, Y)$ where $\mathbf{X} \in \mathbb{R}^p$ is a vector of features, $S \in \{0, 1\}$ is a binary anchor variable, and $Y \in \{0, 1\}$ is the binary response. Further assume that we observe an independent and identically distributed collection of random variables $\{(\mathbf{X}_i, S_i)\}_{i=1}^N$, and only a subset of $\{Y_i\}_{i=1}^N$, each with the positive label. Define the anchor sensitivity $c = \mathbb{P}(S = 1 \mid Y = 1)$, the phenotype prevalence $q = \mathbb{P}(Y = 1)$ and the anchor prevalence by $h = \mathbb{P}(S = 1)$. Impose the assumptions that the anchor S is chosen to be highly predictive of the response, namely that $\mathbb{P}(Y = 1 \mid S = 1) = 1$, and further that S is chosen such that the anchor sensitivity is independent of the features, namely

$$c = \mathbb{P}(Y = 1 \mid S = 1) = \mathbb{P}(Y = 1 \mid S = 1, \mathbf{X}) \tag{1}$$

Finally, Fit a working logistic regression model; $\text{logit}\mathbb{P}(Y = 1 \mid \mathbf{X}, \beta) = \mathbf{X}^T\beta$, by maximum likelihood, where it follows from (1) that $c\mathbb{P}(Y = 1 \mid \mathbf{X}) = \mathbb{P}(S = 1 \mid \mathbf{X})$, and hence the likelihood of $\{(\mathbf{X}_i, S_i)\}_{i=1}^N$ can be written as

$$\begin{aligned} L(\beta, c) &= \prod_{i=1}^N \mathbb{P}(\mathbf{X}_i, S_i = 1)^{S_i} \mathbb{P}(\mathbf{X}_i, S_i = 0)^{1-S_i} \\ &\propto \prod_{i=1}^N \left[c\mathbb{P}(Y = 1 \mid \mathbf{X}_i, \beta)\right]^{S_i} \left[1 - c\mathbb{P}(Y = 1 \mid \mathbf{X}_i, \beta)\right]^{1-S_i} \end{aligned}$$

Prediction is then based on the relationship $c\mathbb{P}(Y = 1 \mid \mathbf{X}) = \mathbb{P}(S = 1 \mid \mathbf{X})$ and the maximum likelihood estimates $\hat{\beta}_{\text{mle}}$ and $\hat{c}_{\text{mle}}$ of β and c obtained by maximizing the above likelihood. Furthermore, additional quantities of interest, like h and q can be estimated using plug-in estimators involving $\hat{\beta}_{\text{mle}}$ and $\hat{c}_{\text{mle}}$.

2.1.2 Automated Feature Selection Algorithm

Assume observations consist of triples $(\mathbf{X}, \mathbf{S}, Y)$ where $\mathbf{X} \in \mathbb{R}^p$ is a vector of features, $S \in \mathbb{R}^q$ is a vector of surrogate features, and $Y \in \{0, 1\}$ is the binary response. Further assume that we observe an independent and identically distributed collection of random variables $\{(\mathbf{X}_i, \mathbf{S}_i)\}_{i=1}^N$, and only a subset of $\{Y_i\}_{i=1}^N$, each with the positive label. Finally, assume that $\mathbf{S} \perp \mathbf{X} \mid Y$, and that $Y \mid \mathbf{X}$ follows a GLM with a known, smooth link function. The model consists of two main steps, clustering and regularized estimation. In the clustering step, impose a parametric mixture model

$$\mathbf{S} \sim \tau f_{\theta_1}(\mathbf{S} \mid Y = 1) + (1 - \tau) f_{\theta_0}(\mathbf{S} \mid Y = 0) \qquad \tau = \mathbb{P}(Y = 1)$$

By obtaining maximum likelihood estimators $\hat{\theta}_i$ of each θ_i, estimate $\pi_S = \mathbb{P}(Y = 1 \mid \mathbf{S})$ by

$$\hat{\pi}_S = \frac{\tau f_{\theta_1}(\mathbf{S} \mid Y = 1, \theta_1 = \hat{\theta}_1)}{\tau f_{\theta_1}(\mathbf{S} \mid Y = 1, \theta_1 = \hat{\theta}_1) + (1 - \tau) f_{\theta_0}(\mathbf{S} \mid Y = 1, \theta_0 = \hat{\theta}_0)}$$

Using $\hat{\pi}_S$ as a response, the regularized estimation step consists of fitting a penalized quasi-logistic regression of $\hat{\pi}_S$ against the features $\mathbf{X}$ by maximum likelihood, using Adaptive LASSO (ALASSO). Namely, the penalty function for regression is

$$R(\beta) = \sum_{j=1}^p |\beta_j| / |\tilde{\beta}_j|$$

where the $\tilde{\beta}_j$'s are the estimated non-intercept coefficients obtained by fitting an unpenalized quasi-logistic regression of $\hat{\pi}_S$ against the features $\mathbf{X}$, and the hyperperameter λ^* in the penalized regression is chosen so that $\sqrt{N}\lambda^*(N) \to \infty$ and $\lambda^*(N) \to 0$ as $N \to \infty$. The estimated active set is then taken as $\hat{\mathcal{A}} = \{j \in [N] : \hat{\beta}_j \neq 0\}$. The final selected set of coefficients is then determined by a resampling of the data to improve estimation. Define by $\mathcal{R}_m$ the indices corresponding to the m^{th} resample of the data, N_b the size of each resample, namely $|\mathcal{R}_m| = N_b$ for all $m \in [M]$, where M is the total number of resamples, and $\hat{\beta}_j^{(m)}$ the estimated j^{th} coefficient corresponding to the m^{th} resample. For a fixed cutoff $\rho_{\text{cut}} \in (0, 1)$, feature selection is then based on

$$\frac{1}{M}\sum_{i=1}^{M} \mathbb{I}(\hat{\beta}_j^{(m)} = 0) < \rho_{\text{cut}} \tag{2}$$

namely feature j is selected when (2) is satisfied.

2.2 The SAPUL Method

Assume our data consists of triples $(\mathbf{X}, S, Y)$, where $\mathbf{X} \in \mathbb{R}^p$ is a vector of features, $S \in \mathbb{R}$ is a surrogate feature, and $Y \in \{0, 1\}$ is the binary response. Further assume that we observe an independent and identically distributed collection of random variables $\{(\mathbf{X}_i, S_i)\}_{i=1}^N$, and only a subset of $\{Y_i\}_{i=1}^N$, each with the positive label. Finally, assume that $S \perp \mathbf{X} \mid Y$, and that $Y \mid \mathbf{X}, S$ is logistic with

$$\text{logit}\mathbb{P}(Y = 1) = \beta_0 + \beta \begin{bmatrix} \mathbf{X}^T & S \end{bmatrix}$$

The method consists of two steps, initial estimation and positive-labelled regression. First, we obtain an estimator $\hat{\beta}^{\text{surr}}$ by fitting a penalized regression of S onto $\mathbf{X}$ by maximum likelihood using ALASSO. To use ALASSO, we obtain an initial estimator $\hat{\beta}^{\text{init}}$ by fitting a ridge regression of S on $\mathbf{X}$ by maximum likelihood, with penalty hyperparameter $\lambda_{\text{init}} = \frac{p}{N}$, namely

$$\hat{\beta}^{\text{init}} = \underset{\beta \in \mathbb{R}^{p+1}}{\text{argmin}} \left\{ \frac{1}{N}\sum_{i=1}^{N} \ell_i(\beta, \mathbf{x}_i) + \lambda_{\text{init}}\|\beta\|_2^2 \right\}$$

where $\ell_i(\beta; \mathbf{X}_i)$ denotes the negative log-likelihood of the i^{th} observation. Then, using the non-intercept estimates $\hat{\beta}_j^{\text{init}}$, we compute $\hat{\beta}^{\text{surr}} \in \mathbb{R}^{p+1}$as

$$\hat{\beta}^{\text{surr}} = \underset{\beta \in \mathbb{R}^{p+1}}{\text{argmin}} \left\{ \frac{1}{N}\sum_{i=1}^{N} \ell_i(\beta, \mathbf{x}_i) + \lambda^* \sum_{j=1}^{p} \frac{\beta_j}{\hat{\beta}^{\text{init}}} \right\} \tag{3}$$

where λ^* is chosen to minimize the adjusted Bayes Information Criterion, defined by

$$\text{BIC}_{\text{adjusted}} = -2\ell(\hat{\beta}) + p \min\{N^B, \log(N)\}$$

where $p = \sum_{i=1}^{p} \mathbb{I}(\hat{\beta}_i \neq 0)$ is the degrees of freedom, and B, the BIC factor, is a hyperparameter. Often times obtaining a solution to (3) can be difficult in practice, and so as per [2], we may employ a quadratic approximation to the likelihood as done in [1]. We begin the second step by obtaining the linear predictor $\mathbf{L} = D\hat{\beta}_{-0}^{\text{surr}}$ from $\hat{\beta}^{\text{surr}}$, where $\hat{\beta}_{-0}^{\text{surr}}$ is the vector of non-intercept coefficient estimates, and D is the design matrix of the observations, namely

$$L_i = X_{i1}\hat{\beta}_1^{\text{surr}} + \cdots + X_{ip}\hat{\beta}_p^{\text{surr}}$$

Lastly, define $\mathbf{d}_i = (S_i, L_i)^T$ and a_i to be the i^{th} observed label, we regress $\mathbf{d}_i$ onto a_i by maximum likelihood. We impose a logistic regression model $\mathbb{P}(a_i = 1 \mid \mathbf{d}_i, \theta) = c_0 \cdot \text{logit}^{-1}(\alpha_0 + \mathbf{d}_i^T\alpha)$ where $\alpha = (\alpha_0, \alpha_1, \alpha_2)^T$ and $\theta = (c_0, \alpha^T)^T$ as per [3]. Write

$$\hat{\theta} = \underset{\theta \in \mathbb{R}^4}{\text{argmin}} \left\{ \sum_{i=1}^{N} \ell_i(\theta; \mathbf{d}_i) \right\}$$

so that as per [4], the final estimator $\hat{\beta}^{\text{SAPUL}} \in \mathbb{R}^{p+2}$ is given by

$$\hat{\beta}^{\text{SAPUL}} = \begin{bmatrix} \hat{\theta}_1 & \hat{\theta}_2 & \hat{\theta}_3\hat{\beta}_1^{\text{surr}} & \cdots & \hat{\theta}_3\hat{\beta}_p^{\text{surr}} \end{bmatrix}$$

3 Numerical Studies

We test the performance of the SAPUL method in a range of simulated settings, where all model assumptions hold, and in settings were the conditional independence of $\mathbf{X}$ and S fails. In all settings, the simulated data consists of $N = 20,000$ unlabelled observations, and varying values of n and p, the number of labelled observations, and the dimension of the features, namely we consider all combinations with $n \in \{50, 100, 200\}$ and $p \in \{50, 100\}$. The response Y is generated from a Bernoulli distribution with success rate 0.3, and labelled responses are randomly selected from $\{Y_i\}_{i=1}^N$. The features $\mathbf{X} \in \mathbb{R}^p$ are generated from a multivariate normal distribution as follows

$$\mathbf{X}_i \sim N_p(0, \Sigma) + \mu_i \qquad \mu_i^T = [1 + \beta_{01} y_i \quad \cdots \quad 1 + \beta_{0p} y_i]$$

where $\beta_0 = (\beta_0^{(1)^T}, \beta_0^{(2)^T})^T \in \mathbb{R}^p$ is initialized in block form with blocks

$$\beta_0^{(1)^T} = [-0.6 \quad 0.6 \quad 0.3 \quad -0.3 \quad 0.3]$$

and $\beta_0^{(2)} = \mathbf{0} \in \mathbb{R}^{p-5}$, and the covariance matrix Σ satisfies $\Sigma_{ij} = \rho^{|i-j|}$ where $\rho = 0.3$. Similarly, the surrogate S is generated from a univariate normal distribution as follows

$$S_i \sim N(1 + \gamma_0 y_i, 1) \qquad \gamma_0 = 1.5$$

Finally, all models fit based on the adjusted BIC will use hyperparameter $B = 0.1$. In simulation settings where the conditional independence assumption is removed, S will be transformed using one of the columns of $\mathbf{X}$, namely we will have either $S_i = \mathbf{X}_{i1}$ or $S_i = \mathbf{X}_{i2}$ (in simulation settings 7 and 8 respectively, as in Table 1)

3.1 Simulation Settings

We consider comparing the SAPUL method to several other methods, some of which making use of the underlying labels we assume to not have full access to. Some of the additional methods are that of an ideal setting, where we have access to all of the underlying labels, and a surrogate only setting, where we have access to none of the labels, and treat the surrogate as a response variable. Each of the methods used are described below in more detail.

3.1.1 Ideal Method

Assuming access to all of the underlying labels, we fit a penalized logistic regression of the responses Y against the features $\mathbf{X}$ and the surrogate S by maximum likelihood, using ALASSO and an a quadratic approximation to the likelihood as in [2]

3.1.2 Surrogate-only Method

We assume no access to any of the labels, and obtain an estimator $\hat{\beta}^{\text{surr}}$ as done in the SAPUL method.

3.1.3 Surrogate Assisted Method

We follow a similar approach to the proposed SAPUL method, but makes use of the underlying labels in the data, effectively an ideal SAPUL setting, rather than strictly an ideal setting. We begin by obtaining $\hat{\beta}^{\text{surr}}$ in the same way as in the SAPUL method, but we obtain an additional estimator $\gamma^{\text{init}} \in \mathbb{R}^3$ by fitting a standard logistic regression of the first 100 observations against $(\mathbf{d}_i, S_i)$, and obtain a final estimator

$$\hat{\beta}^{\text{sass}} = \left[\hat{\gamma}_0^{\text{init}} \quad \hat{\gamma}_1^{\text{init}} \quad \hat{\beta}_1^{\text{surr}}\hat{\gamma}_2^{\text{init}} \quad \cdots \quad \hat{\beta}_p^{\text{surr}}\hat{\beta}_3^{\text{init}}\right] \in \mathbb{R}^{p+2}$$

3.1.4 Semi-Supervised Method

Assuming access to a subset of the underlying observations, namely we observe $\{(\mathbf{X}_i, S_i, Y_i)\}_{i=1}^n$ for $n < N$, we obtain an estimate $\hat{\beta}^{(n)} \in \mathbb{R}^{p+2}$ by fitting a penalized logistic regression of Y_i against $\mathbf{X}_i$ and S_i by maximum likelihood using ALASSO.

With the semi-supervised method, we take values $n = 100$ and $n = 200$, giving us 6 total methods to compare.

3.2 Simulation Results

The metric used for comparing performance is the Area Underneath the Receiver Operating Characteristic curve (AUC), which is computed based on the 200 simulations run in each particular setting. The averaged AUC estimates among all simulations across all models are summarized in Table 1.

Setting			AUC					
n	p	corr	true	surr	sass	sup-100	sup-200	sapul
100	50	FALSE	0.9184	0.8136	0.9139	0.7395	0.8158	0.9120
100	100	FALSE	0.9184	0.8090	0.9121	0.5000	0.7369	0.9103
200	50	FALSE	0.9185	0.8132	0.9143	0.7321	0.8104	0.9043
200	100	FALSE	0.9183	0.8086	0.9118	0.5000	0.7349	0.9006
50	50	FALSE	0.9184	0.8135	0.9144	0.7365	0.8158	0.9017
50	100	FALSE	0.9185	0.8093	0.9124	0.5000	0.7371	0.9031
200	50	TRUE	0.9182	0.7234	0.8523	0.5000	0.7349	0.8260
50	100	TRUE	0.9183	0.4329	0.7767	0.5000	0.7366	0.7263

Table 1: Averaged AUC estimates across simulation settings

Among the settings where CORR, an indicator for the conditional independence assumption to be violated, is FALSE, we can see that the performance of the SAPUL method is highly comparable to the ideal method (denoted TRUE) and the surrogate assisted setting (denoted SASS). As both make use of the underlying labels, it is expected that both out perform the SAPUL method, but the difference is small enough to suggest SAPUL method performs very well in comparison.

In the last two settings, where we violate the conditional independence of the features and surrogate, we have moderate reductions in averaged AUC across all non-ideal methods, and the performance of the SAPUL method is less comparable to the surrogate assisted method as in the previous settings, but the difference is still small enough to be comparable. In the second to last setting in particular, we can see some evidence of robustness of the method under violated assumptions.

4 Conclusion

We proposed a new PU Learning method leveraging surrogate features, and building off of results presented in [3] and [1]. We tested the performance of our proposed method using the AUC metric in several simulation settings, working under our method assumptions and with violated assumptions, against several other methods. With our assumptions in tact, we found that the performance of the SAPUL method was highly comparable with both of the other ideal methods in the simulation study, with averaged AUC estimates consistently in the $[0.9, 0.92]$ range. When the conditional independence of $\mathbf{X}$ and S was violated, we saw a reduction in averaged AUC estimates, falling withing the $[0.72, 0.83]$ range, showing potentially some robustness, but nothing significant. There is opportunity in future to expand on the proposed SAPUL method, potentially leveraging higher dimensional surrogate features rather than a single surrogate variable, and a generalized approach that behaves more robustly when faced with violated assumptions.

References

[1] J. Gronsbell, J. Minnier, S. Yu, K. Liao, and T. Cai. Automated feature selection of predictors in electronic medical records data. *Biometrics*, 2019.

[2] H. Wang and C. Leng. Unified lasso estimation by least squares approximation. *Journal of the American Statistical Association*, 2007.

[3] L. Zhang, X. Ding, Y. Ma, N. Muthu, I. Ajmal, J. H. Moore, D. S. Herman, and J. Chen. A maximum likelihood approach to electronic health record phenotyping using positive and unlabeled patients. *Journal of the American Medical Informatics Association*, 2020.

[4] Y. Zhang, M. Liu, M. Neykov, and T. Cai. Prior adaptive semi-supervised learning with application to ehr phenotyping. 2020.