# Note on Learning with Positive and Unlabeled Data

Haoyue Shi  
Data Intelligence Dept.  
April 10, 2017

## Contents

1. Concepts  
   1.1 MAR and MCAR .................................................. 2  
   1.2 Covariate Shift .................................................. 2  

2. PU Learning  
   2.1 Definition ...................................................... 2  
   2.2 Traditional PU Learning ........................................ 2  
   2.3 Applications of PU Learning .................................... 3  
      2.3.1 Identification of unexpected instances ................. 3  
      2.3.2 Dealing with the problem of covariate shift .......... 3  
      2.3.3 PU learning for matrix completion ...................... 4  
      2.3.4 Classifying networked text data with positive and unlabeled examples .......... 4  
   2.4 Theoretical Analysis of PU Learning ....................... 4  
      2.4.1 Problem Reformulation .................................... 5  
      2.4.2 Class Prior Estimation .................................... 5  
      2.4.3 Necessity of non-convex loss functions in PU classification ............ 7  
      2.4.4 A convex re-formulation of PU learning ............... 8  

3. Related Dataset .................................................. 8
1 Concepts

1.1 MAR and MCAR

**Missing Completely at random (MCAR)** There is no relationship between whether a data point is missing and any values in the data set, missing or observed.

**Missing at Random (MAR)** Missing at Random means the propensity for a data point to be missing is not related to the missing data, but it is related to some of the observed data. A better name would actually be Missing Conditionally at Random.

1.2 Covariate Shift

An assumption that one can make about the source and the target domains is that given the same observation \( X = x \), the conditional distributions of \( Y \) are the same in the two domains. However, the marginal distributions of \( X \) may be different in the source and the target domains. Formally, we assume that \( P_s(Y|X = x) = P_t(Y|X = x) \) for all \( x \in \mathcal{X} \), but \( P_s(X) \neq P_t(X) \). This difference between the two domains is called covariate shift (Shimodaira 2000)

Under a misspecified model family, the estimation of \( \Theta \) depends on \( P(X) \), rather than only samples themselves, where \( \Theta \) refers to the parameters of the distribution. However, we could weight each training sample on the training stage by:

\[
\frac{P_t(x, y)}{P_s(x, y)} = \frac{P_t(x)}{P_s(x)} \frac{P_t(y|x)}{P_s(y|x)} = \frac{P_t(x)}{P_s(x)}
\]

Many works focused on how to estimate this rate.

In machine learning, this problem is often solved by assigning each sample a different weight. In deep learning, batch normalization is a common method to solve the covariate shift problem.

2 PU Learning

2.1 Definition

PU learning refers to learning with positive and unlabeled samples. (Liu, Dai, Li, Lee, and Yu 2003) It distinguishes with semi-supervised learning by the lack of negative training samples.

2.2 Traditional PU Learning

According to (Liu, Dai, Li, Lee, and Yu 2003), conventional PU Learning contains two steps:

1. Reliable negative data mining: to find some reliable negative training samples from the unlabeled data set.
2. Classifier training: to train a classifier / classifiers using the positive and negative data extracted from the unlabeled data set.

To conduct the first step, we have 4 methods:

- **Naive Bayesian Classifier**: naive Bayesian classifier can produce the probability of each sample to be a positive/negative one. The system would choose the “most negative” samples as reliable ones.
- **the Rocchio Technique**: to find a representation vector for a class, which can be enhanced by clustering. While choosing reliable negative samples, the system selects those with the least similarity with each class.
- **the Spy Technique**: using positive samples as unlabeled ones, and decide threshold for reliable negative samples according to the spies.
- **1-DNF**: mine those positive features, and extract reliable negative samples (without such features). It’s different from Rocchio Technique by the difference between “features” and “vector of class”.

The second step can be:

- **SVM**: to minimize

\[
\frac{1}{2} w^T w \tag{1}
\]

\[
s.t. \ y_i (w^T x_i + b) \geq 1
\]
• EM algorithm: often collocates with the spy technique, and an NB classifier.
• Iterative SVM, combining the idea of EM algorithm with SVM.
• Iterative SVM Classifier with choosing algorithm - choose the best classifier generated from the previous iterations.

Note that there is a strong assumption behind NB classifier: features are independent with each other, due to the proof of Bayesian equation.

2.3 Applications of PU Learning

2.3.1 Identification of unexpected instances

(Li, Liu, and Ng 2007) formulated the unexpected instance identification problem into a PU learning problem. Using LGN (PU Learning with Generating Negative samples) technique, they obtained good results. For negative sample (document) generation, they treat each word as a feature. Each feature’s frequency in the negative document $A_N$ is randomly generated following a Gaussian distribution according to $q(w_i) = \frac{1 - \text{entropy}(w_i)}{\max_{w_j \in V} \text{entropy}(w_j)}$. What’s more, (Blanchard, Lee, and Scott 2010) have proposed a method and the analysis of semi-supervised novelty detection. This would be mentioned again in the section of semi-supervised learning.

2.3.2 Dealing with the problem of covariate shift

(Li, Liu, and Ng 2010) indicated that when the positive training and test samples have identical distributions, the negative training and test samples may have different distributions. (Note this is a common case for text binary classification.) In such cases, PU learning can perform better than those semi-supervised learning algorithm, with respect to negative training data. They evaluated their algorithm empirically.

In the work of (Fei and Liu 2015), they proposed center based space (CBS) features to replace traditional document space (DS), for the specific problem of covariate shift on social media text classification. They obtain the CBS in the following ways:

1. Generate the document space (e.g. tf-idf feature) with traditional methods.
2. Cluster documents into $k$ clusters.
3. Compute the CBS with $\text{sim}(x, c_k)$, and obtain a $k$-dimensional space.

They give out the following reason to explain why CBS works. The reason is that due to the use of similarity features, CBS-Learning is essentially trying to generate a boundary for the positive training data because similarity is not directional and thus covers all directions in a spherical shape in the space. In classification, the negative data from anywhere or direction outside the spherical shape can be detected. The covariate shift problem will not affect the classification much. Many types of documents that are not represented in the negative training data will still be detected due to their low similarity. For example, in Figure 1, we want to build a SVM classifier to separate positive data represented as black squares and negative data represented as empty circles.

![Figure 1: CBS vs DS.](image-url)
2.3.3 PU learning for matrix completion

Problem Formulation: given a uncompleted matrix $A$, we aim to find

$$X = \arg\min_X \sum_{i,j} \tilde{l}(X_{ij}, A_{ij})$$

$$s.t. \|X\|_* \leq t, 0 \leq X_{ij} \leq 1 \forall (i, j)$$

where

$$\tilde{l}(X_{ij}, A_{ij}) = \begin{cases} 
(X_{ij} - 1)^2 - \rho X_{ij}^2 & \text{if } A_{ij} = 1 \\
X_{ij}^2 & \text{if } A_{ij} = 0 
\end{cases}$$

and $\rho = P(A_{ij} = 0 | Y_{ij} = 1)$ is the missing rate, with the assumption $(1 - \rho) \times n \times n = O(\log n)$. (Hsieh, Natarajan, and Dhillon 2015) Actually, it is a subproblem of PU learning. The paper has given out a specific method for solving matrix completion problem only.

The bound constraint on $X$ in the above estimator ensures the loss has bounded Lipschitz constant. This optimization PU Learning for Matrix Completion problem is equivalent to the traditional trace-norm regularization method for solving matrix completion problem only. Let $F$ be a permissible class of functions with VC-dimension $d$ and let $f_1 \in F$ be the target function. Let $v = E[f_1(X)]$. Let $P = X_1, ..., X_{n_1}$ be drawn from the distribution of positive examples $D_{X|Y=1}$ where

$$n_1 > \frac{16}{\epsilon} \left( d \ln \left( \frac{16e}{\epsilon} \ln \frac{16e}{\epsilon} \right) + \ln \frac{24}{\delta} \right)$$

Let $M = Z_1, ..., Z_{n_2}$ be unlabeled examples drawn independently from $D_X$ where

$$n_2 > \frac{16}{\epsilon} \left( d \ln \left( \frac{16e}{\epsilon} \ln \frac{16e}{\epsilon} \right) + \ln \frac{24}{\delta} \right)$$

Let $\tilde{F}$ be the subset of $F$ that archives total recall on $P$ and $\hat{f} = \arg\min_{f \in \tilde{F}} \sum_{i=1}^{n_2} f(Z_i)$. Then, with probability at least $1 - \delta$, $ER(\hat{f}) > 1 - \epsilon$, $EP(\hat{f}) > \frac{1 - \epsilon}{1 + \log \frac{1}{\delta}}$ and $E[\hat{f}(X) \neq f_1(X)] < (10v + 4)\epsilon$.

Noisy case (false positive data in the training set) would perform like the noiseless one, w.r.t. the noise rate $r$. Please refer to (Liu, Lee, Yu, and Li 2002).
2.4.1 Problem Reformulation

According to (du Plessis, Niu, and Sugiyama 2014), PU Learning problems can be solve with cost-sensitive learning:

Ordinary classification aims to find the decision function \( F(X) \in \{1, -1\} \) that minimizes the expected misclassification rate w.r.t a class prior of \( \pi \):

\[
R(f) := \pi R_1(f) + (1 - \pi) R_{-1}(f)
\]

where \( R_1(f) \) and \( R_{-1}(f) \) denote the expected false positive rate and expected false negative rate.

Cost-sensitive classification: A cost-sensitive classifier selects a function \( f(X) \in \{1, -1\} \) in order to minimize the weighted expected misclassification rate:

\[
R(f) := \pi c_1 R_1(f) + (1 - \pi) c_{-1} R_{-1}(f)
\]

where \( c_1 \) and \( c_{-1} \) are per-class costs (Elkan 2001). Since scaling does not matter in (8), it is often useful to interpret the per-class costs as reweighting the problem according to new class priors proportional to \( \pi c_1 \) and \( (1 - \pi) c_{-1} \).

In PU classification, a classifier is learned using labeled data drawn from the positive class \( P_1 \) and unlabeled data that is a mixture of positive and negative samples with unknown class prior \( \pi \) (because of the lack of negative data):

\[
P_X = \pi P_1 + (1 - \pi) P_{-1}
\]

Note that \( P_X \) here can be intuitively treated as the distribution of unlabeled data.

Since negative samples are not available, we can train a classifier to minimize the expected misclassification rate between positive and unlabeled samples (Liu, Lee, Yu, and Li 2002). Thus, we rewrite the risk \( R(f) \) not to include \( R_{-1}(f) \). More specifically, let \( R_X(f) \) be the probability that the function \( f(X) \) gives the positive label over \( P_X \) (Blanchard, Lee, and Scott 2010):

\[
R_X(f) = P_X(f(X) = 1) = \pi P_1(f(X) = 1) + (1 - \pi) P_{-1}(f(X) = 1) = \pi (1 - R_1(f)) + (1 - \pi) R_{-1}(f)
\]

where \( R_1(f) = P_1(f(X) \neq 1) \) and \( R_{-1}(f) = P_{-1}(f(X) \neq -1) \).

Then the risk \( R(f) \) can be written as

\[
R(f) = \pi R_1(f) + (1 - \pi) R_{-1}(f)
\]

\[
= 2 \pi R_1(f) + R_X(f) - \pi
\]

\[
(11)
\]

Let \( \eta \) be the proportion of samples from \( P_1 \) compared to \( P_X \), which is empirically estimated by \( \frac{n}{n + n'} \) where \( n \) and \( n' \) denote the numbers of positive and unlabeled samples, respectively. The risk \( R(f) \) can be expressed as

\[
R(f) = c_1 \eta R_1(f) + c_X (1 - \eta) R_X(f) - \pi
\]

(12) where \( c_1 = \frac{2 \pi}{\eta} \) and \( c_X = \frac{1}{1 - \eta} \). Comparing this expression with (8), it can be solved by cost-sensitive classification, assigning weights (costs) \( c_1 \) and \( c_X \) to the two classes. In practice, the unknown class prior \( \pi \) can be estimated by the methods proposed in (Du Plessis and Sugiyama 2014; Elkan and Noto 2008; Blanchard, Lee, and Scott 2010).

2.4.2 Class Prior Estimation

(Du Plessis and Sugiyama 2014) proposed a method with the assumption of data is drawn according to

\[
(x, y, s) \sim^{i.i.d} p(x, y, s)
\]

(13) where \( x \) are the unlabeled features, \( y \in \{1, -1\} \) are the (unknown) class labels, and \( s \in \{0, 1\} \) determines whether the sample is labeled. Since we only have labeled data with \( y = 1 \), the dataset would typically be

\[
\mathcal{X} := \{(x_i, s_i)\}_{i=1}^n \sim^{i.i.d.} p(x, s)
\]

(14) (Elkan and Noto 2008) have proven that

\[
p(y = 1|x) = \frac{1}{e} p(s = 1|x)
\]

(15)
where \( c = p(s = 1 | y = 1) \).

The posterior \( p(s = 1 | x) \) is referred to in (Elkan and Noto 2008) as a “non-traditional” (i.e. the training set only contains positive and unlabeled data) classifier. This can be estimated from the training dataset by a probabilistic classification method such as kernel logistic regression (Friedman, Hastie, and Tibshirani 2001) or its squared-loss variant (Sugiyama 2010).

From the assumptions, the positive samples \( X := \{ x'_i \}_{i=1}^n \) are drawn according to \( p(x|y = 1) \).

We model the input density as

\[
q(x; \theta) = \theta p(x|y = 1) + (1 - \theta) p(x|y = -1)
\]

where \( \theta \in [0, 1] \) is a scalar value that represents the unknown class prior \( p(y = 1) \). (i.e. \( \pi \) in previous part) The above model \( q(x; \theta) \) would equal \( p(x) \) if \( \theta \) is the unknown class prior \( p(y = 1) \). Therefore, by selecting \( \theta \) so that the two distributions are equal, the class prior can be estimated. (left hand of Figure 2) However, due to the lack of the knowledge of \( p(x|y = -1) \), this algorithm does not perform well in PU learning tasks.

Nevertheless, if the class-conditional densities \( p(x|y = -1) \) and \( p(x|y = 1) \) are not overlapping strongly, we may estimate \( \theta \) so that \( \theta p(x|y = 1) \) is as similar to \( p(x) \) as possible (right hand of Figure 2). They use Pearson divergence to estimate \( \theta \).

\[
\theta^* = \arg \min_{\theta} PE(\theta)
\]

where \( PE(\theta) \) denotes the PE divergence from \( \theta p(x|y = 1) \) to \( p(x) \).

\[
PE = \frac{1}{2} \int \left( \frac{\theta p(x|y = 1)}{p(x)} - 1 \right)^2 p(x) dx = \frac{1}{2} \int \left( \frac{\theta p(x|y = 1)}{p(x)} \right)^2 p(x) dx - \theta + \frac{1}{2}
\]

The above PE divergence is defined in terms of unknown densities, but only samples drawn from these densities are available. A possible approach is to first estimate \( p(x|y = 1) \) and \( p(x) \) from the samples using, e.g., kernel density estimation and then plug these estimators into the above expression. This however does not work well since high-dimensional density estimation is a difficult problem. Furthermore, the division by an estimated density may exacerbate the estimation error.

Figure 2: Left hand side: full matching; Right hand side: partial matching

The proposed method by (Du Plessis and Sugiyama 2014) can avoid density estimation and directly minimize the PE divergence. Consider a lower bound which is linear in the unknown densities and can then be estimated from sample averages. Using inequality \( \frac{a^2}{2} \geq ty - t^2 \) which can be obtained from Fenchel duality. (Keziou 2003; Nguyen, Wainwright, and Jordan 2010), we can lower bound (18) in a pointwise manner as follows:

\[
\frac{1}{2} \left( \frac{\theta p(x|y = 1)}{p(x)} \right)^2 \geq \left( \frac{\theta p(x|y = 1)}{p(x)} \right) r(x) - \frac{1}{2} r(x)^2
\]

where \( r(x) \) performs as \( t \). This yields

\[
\frac{1}{2} \left( \frac{\theta p(x|y = 1)}{p(x)} \right)^2 p(x) \geq \theta p(x|y = 1) r(x) - \frac{1}{2} r(x)^2 p(x)
\]
Therefore,

\[ PE \geq \theta \int r(x)p(x|y=1)dx - \frac{1}{2} \int r(x)^2p(x)dx - \theta + \frac{1}{2} \]  

(21)

In practice, we could use a parametric model for \( r(x) \), replace the integrals with sample averages, and select the tightest bound via maximization of the right hand side.

In their paper, they proposed a linear-in-parameter model \( \hat{r}(x) = \alpha^T \phi(x) \) (Du Plessis and Sugiyama 2014), where \( \alpha \) are the parameters and \( \phi(x) = (\phi_1(x), ..., \phi_n(x))^T \) are the basis functions.

Hence, the class prior \( \theta \) should be

\[ \hat{\theta} = [\hat{h}^T \hat{G}^{-1} \hat{h} - \hat{h}^T \hat{G}^{-1} \hat{H} \hat{G}^{-1} \hat{h}]^{-1} \]  

(22)

where

\[ \hat{H} = \int \phi(x)\phi(x)^T p((x))dx, \quad \hat{h} = \int \phi(x)p(x|y=1)dx \]

\( \hat{H} \) and \( \hat{h} \) are the discrete version of \( H \) and \( h \), and \( \hat{G} = \hat{H} + \lambda I \) For detailed information, please see the paper (Du Plessis and Sugiyama 2014). Experiments have shown that the algorithm performs well in prior estimation tasks.

In the paper (du Plessis, Niu, and Sugiyama 2014), they also demonstrated that when the true class prior is known to be large (such as the proportion of inliers in inlier-based outlier detection), a rough class-prior estimator is sufficient to have a good classification performance. On the other hand, if the true class prior is small, PU classification tends to be hard and an accurate class-prior estimator is necessary.

### 2.4.3 Necessity of non-convex loss functions in PU classification

(du Plessis, Niu, and Sugiyama 2014) showed that solving PU classification problem with a convex loss function may lead to a biased solution, and the use of a non-convex loss function is essential to avoid this problem.

**Ramp loss (non-convex)**

\[ l_R(z) = \frac{1}{2} \max(0, \min(2, 1 - z)) \]  

(23)

gives the objective function of

\[ J_R(g) = \pi\mathbb{E}_1[l_R(g(X))] + (1 - \pi)\mathbb{E}_{-1}[l_R(-g(X))] \]  

(24)

**Hinge loss (convex)**

\[ l_H(z) = \frac{1}{2} \max(1 - z, 0) \]  

(25)

gives the objective function of

\[ J_H(g) = \pi\mathbb{E}_1[l_H(g(X))] + (1 - \pi)\mathbb{E}_{-1}[l_H(-g(X))] \]  

(26)

According to the previous equations, we can rewrite the objective functions into PU learning versions. In the case of PU classification, since hinge loss is not non-symmetric (but ramp loss is), it gives out superfluous penalty. The detailed analysis is in (du Plessis, Niu, and Sugiyama 2014). Note that there is a contradiction between symmetric and convex, however, it’s important to be symmetric (even more important than to be convex) to obtain a correct boundary.

(Collobert, Sinz, Weston, and Bottou 2006; Suzumura, Ogawa, Sugiyama, and Takeuchi 2014) give the methods to optimize a non-convex objective function with ramp loss.
2.4.4 A convex re-formulation of PU learning

(du Plessis, Niu, and Sugiyama 2015) gave a convex formulation version of PU learning problem, against their non-convex one, with changing the 0-1 objective function.

The initial objective function should be

\[ J_{0-1}(g) = \pi \mathbb{E}_1[l_{0-1}(g(X))] = (1 - \pi)\mathbb{E}_{-1}[l_{0-1}(-g(X))] - \pi \mathbb{E}_1[l_{0-1}(-g(X))] \]  

(27)

where the 0-1 loss is

\[ l_{0-1}(z) = \frac{1}{2} \text{sign}(z) + \frac{1}{2}. \]

Consider the objective function based on the following equation

\[ (1 - \pi)\mathbb{E}_{-1}[l_{0-1}(-g(X))] = \mathbb{E}_X[l_{0-1}(-g(X))] - \pi \mathbb{E}_1[l_{0-1}(-g(X))] \]  

(28)

By substituting this into Eq(27), we obtain

\[ J(g) = \pi \mathbb{E}_1[\tilde{l}(g(X))] + \mathbb{E}_X[l(-g(X))] \]  

(29)

where \( \tilde{l}(z) = l(z) - l(-z) \) is the composite loss. Then the problem comes to how to choose the loss function \( l(z) \) to make \( \tilde{l}(z) \) convex (therefore the whole objective function would be convex). They explored several loss functions, and showed that double hinge loss can obtain similar result with non-convex ramp loss (du Plessis, Niu, and Sugiyama 2014).

3 Related Dataset

Most of the paper published on NIPS and ICML only have numerical illustrations or simulated (ideal and perfect) data.

- Toy corpus from (Liu, Dai, Li, Lee, and Yu 2003)
- 20 Newsgroups.
  Link: http://qwone.com/˜jason/20Newsgroups/
  Used by (Li, Liu, and Ng 2010)
- ca-Hepth and ca-GrQc, graph completion problem
  Link: https://snap.stanford.edu/data/ca-HepPh.html
  https://snap.stanford.edu/data/ca-GrQc.html
  Used by (Hsieh, Natarajan, and Dhillon 2015)
- IJCAI 2016 challenge task
  Link: https://tianchi.aliyun.com/competition/introduction.htm?spm=5176.100068.5678.1.fuqdis&raceId=231532
References


