

# Analysis of Learning from Positive and Unlabeled Data

**Authored by:**

Masashi Sugiyama  
Gang Niu  
Marthinus C. du Plessis

## **Abstract**

Learning a classifier from positive and unlabeled data is an important class of classification problems that are conceivable in many practical applications. In this paper, we first show that this problem can be solved by cost-sensitive learning between positive and unlabeled data. We then show that convex surrogate loss functions such as the hinge loss may lead to a wrong classification boundary due to an intrinsic bias, but the problem can be avoided by using non-convex loss functions such as the ramp loss. We next analyze the excess risk when the class prior is estimated from data, and show that the classification accuracy is not sensitive to class prior estimation if the unlabeled data is dominated by the positive data (this is naturally satisfied in inlier-based outlier detection because inliers are dominant in the unlabeled dataset). Finally, we provide generalization error bounds and show that, for an equal number of labeled and unlabeled samples, the generalization error of learning only from positive and unlabeled samples is no worse than  $\sqrt{2}$  times the fully supervised case. These theoretical findings are also validated through experiments.

## **1 Paper Body**

Let us consider the problem of learning a classifier from positive and unlabeled data (PU classification), which is aimed at assigning labels to the unlabeled dataset [1]. PU classification is conceivable in various applications such as land-cover classification [2], where positive samples (built-up urban areas) can be easily obtained, but negative samples (rural areas) are too diverse to be labeled. Outlier detection in unlabeled data based on inlier data can also be regarded as PU classification [3, 4]. In this paper, we first explain that, if the class prior in the unlabeled dataset is known, PU classification can be reduced to the problem of cost-sensitive classification [5] between positive and unlabeled data. Thus, in principle, the PU classification problem can be solved by a standard cost-sensitive classifier such as the weighted support vector machine [6]. The goal

of this paper is to give new insight into this PU classification algorithm. Our contributions are three folds: ? The use of convex surrogate loss functions such as the hinge loss may potentially lead to a wrong classification boundary being selected, even when the underlying classes are completely separable. To obtain the correct classification boundary, the use of non-convex loss functions such as the ramp loss is essential. 1

? When the class prior in the unlabeled dataset is estimated from data, the classification error is governed by what we call the effective class prior that depends both on the true class prior and the estimated class prior. In addition to gaining intuition behind the classification error incurred in PU classification, a practical outcome of this analysis is that the classification error is not sensitive to class-prior estimation error if the unlabeled data is dominated by positive data. This would be useful in, e.g., inlier-based outlier detection scenarios where inlier samples are dominant in the unlabeled dataset [3, 4]. This analysis can be regarded as an extension of traditional analysis of class priors in ordinary classification scenarios [7, 8] to PU classification. ? We establish generalization error bounds for PU classification. For an equal number of positive and unlabeled samples, the convergence rate is no worse than 2 times the fully supervised case. Finally, we numerically illustrate the above theoretical findings through experiments.

2

#### PU classification as cost-sensitive classification

In this section, we show that the problem of PU classification can be cast as cost-sensitive classification. Ordinary classification: The Bayes optimal classifier corresponds to 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:  $R_{-1}(f) = P_{-1}(f(X) \neq -1)$  and

$$R_1(f) = P_1(f(X) \neq 1),$$

and  $P_1$  and  $P_{-1}$  denote the marginal probabilities of positive and negative samples. In the empirical risk minimization framework, the above risk is replaced with their empirical versions obtained from fully labeled data, leading to practical classifiers [9]. Cost-sensitive classification: A cost-sensitive classifier selects a function  $f(X) \in \{1, -1\}$  in order to minimize the weighted expected misclassification rate:  $R(f) := c_1 R_1(f) + (1 - c_1) R_{-1}(f)$ ,

(1)

where  $c_1$  and  $c_{-1}$  are the per-class costs [5]. Since scaling does not matter in (1), it is often useful to interpret the per-class costs as reweighting the problem according to new class priors proportional to  $c_1$  and  $(1 - c_1)$ . PU classification: 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  $\pi$ :  $P_X = \pi P_1 + (1 - \pi) P_{-1}$ . Since negative samples are not available, let us train a classifier to minimize the expected misclassification rate between positive and unlabeled samples. Since we do not have negative samples in the PU classification setup, we cannot directly estimate  $R_{-1}(f)$  and 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  $P_X$  [10]:  $R_X(f) = P_X(f(X) = 1) = \int P_1(f(X) = 1) + (1 - \int)P_0(f(X) = 1) = \int(1 - R_1(f)) + (1 - \int)R_0(f)$ .  $2$

(2)

Then the risk  $R(f)$  can be written as  $R(f) = \int R_1(f) + (1 - \int)R_0(f) = \int R_1(f) - \int(1 - R_1(f)) + R_X(f) = 2\int R_1(f) + R_X(f) - \int$ .

(3)

Let  $\int$  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 then be expressed as  $R(f) = c_1 \int R_1(f) + c_X (1 - \int)R_X(f) - \int$ ,

where  $c_1 =$

$2\int - \int$

and

$c_X =$

$1 - 1\int$

Comparing this expression with (1), we can confirm that the PU classification problem is solved by cost-sensitive classification between positive and unlabeled data with costs  $c_1$  and  $c_X$ . Some implementations of support vector machines, such as libsvm [6], allow for assigning weights to classes. In practice, the unknown class prior  $\int$  may be estimated by the methods proposed in [10, 1, 11]. In the following sections, we analyze this algorithm.

**3 Necessity of non-convex loss functions in PU classification** In this section, we show that solving the 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. Loss functions in ordinary classification: We first consider ordinary classification problems where samples from both classes are available. Instead of a binary decision function  $f(X) \in \{1, 1\}$ , a continuous decision function  $g(X) \in \mathbb{R}$  such that  $\text{sign}(g(X)) = f(X)$  is learned. The loss function then becomes  $J_{0-1}(g) = \int E_1[0-1(g(X))] + (1 - \int)E_0[0-1(g(X))]$ , where  $E_y$  is the expectation over  $P_y$  and  $0-1(z)$  is the zero-one loss:  $\begin{cases} 0 & z \geq 0 \\ 1 & z < 0 \end{cases}$ . Since the zero-one loss is hard to optimize in practice due to its discontinuous nature, it may be replaced with a ramp loss (as illustrated in Figure 1):  $\int R(z) =$

$\frac{1}{2} \max(0, \min(2, 1 + z))$ ,  $2$

giving an objective function of  $J_R(g) = \int E_1[\int R(g(X))] + (1 - \int)E_0[\int R(g(X))]$ .

(4)

To avoid the non-convexity of the ramp loss, the hinge loss is often preferred in practice:  $\int H(z) =$

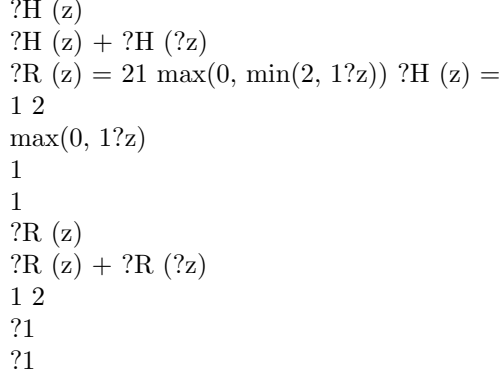
$\frac{1}{2} \max(1 - z, 0)$ ,  $2$

giving an objective of  $J_H(g) = \int E_1[\int H(g(X))] + (1 - \int)E_0[\int H(g(X))]$ .

(5)

One practical motivation to use the convex hinge loss instead of the non-convex ramp loss is that separability (i.e.,  $\min_g J_R(g) = 0$ ) implies  $\int R(z) = 0$  everywhere, and for all values of  $z$  for which  $\int R(z) = 0$ , we have  $\int H(z) =$

0. Therefore, the convex hinge loss will give the same decision boundary as the non-convex ramp loss in the ordinary classification setup, under the assumption that the positive and negative samples are non-overlapping. 3



1 (a) Loss functions

1 (b) Resulting penalties

Figure 1:  $R(z)$  denotes the ramp loss, and  $H(z)$  denotes the hinge loss.  $R(z) + R(-z)$  is constant but  $H(z) + H(-z)$  is not and therefore causes a superfluous penalty. Ramp loss function in PU classification: An important question is whether the same interpretation will hold for PU classification: can the PU classification problem be solved by using the convex hinge loss? As we show below, the answer to this question is unfortunately no. In PU classification, the risk is given by (3), and its ramp-loss version is given by  $JPU-R(g) = 2E[R(f)] + R(f) = 2E[R(g(X))] + [E[R(-g(X))] + (1 - E[R(-g(X))])]$   $= E[R(g(X))] + E[R(g(X)) + R(-g(X))] + (1 - E[R(-g(X))])$ , (6) (7) (8)

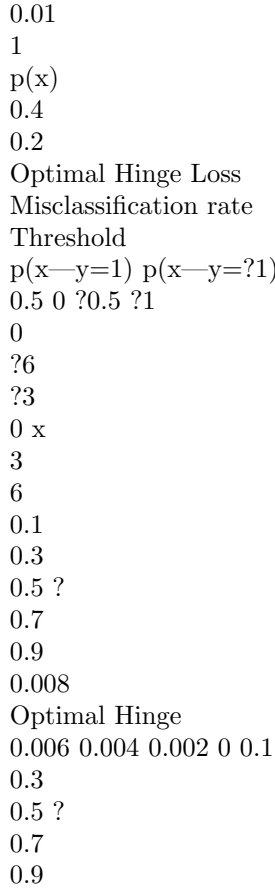
where (6) comes from (3) and (7) is due to the substitution of (2). Since the ramp loss is symmetric in the sense of  $R(-z) + R(z) = 1$ , (8) yields  $JPU-R(g) = E[R(g(X))] + (1 - E[R(-g(X))])$ . (9)

(9) is essentially the same as (4), meaning that learning with the ramp loss in the PU classification setting will give the same classification boundary as in the ordinary classification setting. For non-convex optimization with the ramp loss, see [12, 13]. Hinge loss function in PU classification: On the other hand, using the hinge loss to minimize (3) for PU learning gives  $JPU-H(g) = 2E[H(g(X))] + [E[H(-g(X))] + (1 - E[H(-g(X))])]$   $= E[H(g(X))] + (1 - E[H(-g(X))]) + E[H(g(X)) + H(-g(X))]$ .  $\{z\}$  — Ordinary error term, cf. (5)

Superfluous penalty

We see that the hinge loss has a term that corresponds to (5), but it also has a superfluous penalty term (see also Figure 1). This penalty term may cause an incorrect classification boundary to be selected. Indeed, even if  $g(X)$

perfectly separates the data, it may not minimize JPU-H (g) due to the superfluous penalty. To obtain the correct decision boundary, the loss function should be symmetric (and therefore non-convex). Alternatively, since the superfluous penalty term can be evaluated, it can be subtracted from the objective function. Note that, for the problem of label noise, an identical symmetry condition has been obtained [14]. Illustration: We illustrate the failure of the hinge loss on a toy PU classification problem with class conditional densities of:  $p(x|y=1) = N(-3, 12)$  and  $p(x|y=-1) = N(3, 12)$ , where  $N(\mu, \sigma^2)$  is a normal distribution with mean  $\mu$  and variance  $\sigma^2$ . The hinge-loss objective function for PU classification, JPU-H (g), is minimized with a model of  $g(x) = wx + b$  (the expectations in the objective function is computed via numerical integration). The optimal decision



(a) Class-conditional densities of (b) Optimal threshold and threshold (c) The misclassification rate for the problem using the hinge loss the optimal and hinge loss case

Figure 2: Illustration of the failure of the hinge loss for PU classification. The optimal threshold and the threshold estimated by the hinge loss differ significantly (Figure 2(b)), causing a difference in the misclassification rates

(Figure 2(c)). The threshold for the ramp loss agrees with the optimal threshold. threshold and the threshold for the hinge loss is plotted in Figure 2(b) for a range of class priors. Note that the threshold for the ramp loss will correspond to the optimal threshold. From this figure, we note that the hinge-loss threshold differs from the optimal threshold. The difference is especially severe for larger class priors, due to the fact that the superfluous penalty is weighted by the class prior. When the class-prior is large enough, the large hinge-loss threshold causes all samples to be positively labeled. In such a case, the false negative rate is  $R_1 = 0$  but the false positive rate is  $R_0 = 1$ . Therefore, the overall misclassification rate for the hinge loss will be  $1/2$ .

4

#### Effect of inaccurate class-prior estimation

To solve the PU classification problem by cost-sensitive learning described in Section 2, the true class prior  $\eta$  is needed. However, since it is often unknown in practice, it needs to be estimated, e.g., by the methods proposed in [10, 1, 11]. Since many of the estimation methods are biased [1, 11], it is important to understand the influence of inaccurate class-prior estimation on the classification performance. In this section, we elucidate how the error in the estimated class prior  $\hat{\eta}$  affects the classification accuracy in the PU classification setting. Risk with true class prior in ordinary classification: In the ordinary classification scenarios with positive and negative samples, the risk for a classifier  $f$  on a dataset with class prior  $\eta$  is given as follows ([8, pp. 26–29] and [7]):  $R(f, \eta) = \eta R_1(f) + (1 - \eta) R_0(f)$ . The risk for the optimal classifier according to the class prior  $\eta$  is therefore,  $R^*(\eta) = \min_{f \in \mathcal{F}} R(f, \eta)$ .

Note that  $R^*(\eta)$  is concave, since it is the minimum of a set of functions that are linear w.r.t.  $\eta$ . This is illustrated in Figure 3(a). Excess risk with class prior estimation in ordinary classification: Suppose we have a classifier  $f_{\hat{\eta}}$  that minimizes the risk for an estimated class prior  $\hat{\eta}$ :  $f_{\hat{\eta}} := \arg \min_{f \in \mathcal{F}} R(f, \hat{\eta})$ .

The risk when applying the classifier  $f_{\hat{\eta}}$  on a dataset with true class prior  $\eta$  is then on the line tangent to the concave function  $R^*(\eta)$  at  $\eta = \hat{\eta}$ , as illustrated in Figure 3(a):  $R(f_{\hat{\eta}}, \eta) = \eta R_1(f_{\hat{\eta}}) + (1 - \eta) R_0(f_{\hat{\eta}})$ . The function  $f_{\hat{\eta}}$  is suboptimal at  $\eta$ , and results in the excess risk [8]:  $E_{\hat{\eta}} = R(f_{\hat{\eta}}, \eta) - R^*(\eta)$ .

1 0.95 0.9

$R^*(\eta) = R(\eta)$

Risk

Effective prior  $\eta_e$

$R(\eta)$

0.2

$\eta = 0.95 \quad \eta = 0.9 \quad \eta = 0.7 \quad \eta = 0.5$

0.8

$E_{\hat{\eta}} 0.1$

0.7 0.6 0.5 0.4 0.3 0.2 0.1

$\eta$

$\eta_e$

0 0.2

1  
0.3  
0.4  
Class prior  
0.5 0.6 0.7 Estimated prior  $\eta$   $b$   
0.8  
0.9 0.95 1

(b) The effective class prior  $\eta_e$  vs. the estimated class prior  $\eta$   $b$  for different true class priors  $\eta$ .

(a) Selecting a classifier to minimize (11) and applying it to a dataset with class prior  $\eta$  leads to an excess risk of  $E$ ?

Figure 3: Learning in the PU framework with an estimated class prior  $\eta$   $b$  is equivalent to selecting a classifier which minimizes the risk according to an effective class prior  $\eta_e$ . (a) The difference between the effective class prior  $\eta_e$  and the true class prior  $\eta$  causes an excess risk  $E$ ?. (b) The effective class prior  $\eta_e$  depends on the true class prior  $\eta$  and the estimated class prior  $\eta$   $b$ . Excess risk with class prior estimation in PU classification: We wish to select a classifier that minimizes the risk in (3). In practice, however, we only know an estimated class prior  $\eta$   $b$ . Therefore, a classifier is selected to minimize  $R(f) = 2b \eta R_1(f) + R_X(f)$ ?

(11)

Expanding the above risk based on (2) gives  $R(f) = 2b \eta R_1(f) + (1 - \eta) R_1(f) + (1 - \eta) R_{-1}(f) = (2b \eta + 1 - \eta) R_1(f) + (1 - \eta) R_{-1}(f)$ ?. Thus, the estimated class prior affects the risk with respect to  $2b \eta$ ?.  $\eta$  and  $1 - \eta$ ?. This result immediately shows that PU classification cannot be performed when the estimated class prior is less than half of the true class prior:  $\eta < 1/2$ ?. We define the effective class prior  $\eta_e$  so that  $2b \eta_e$  and  $1 - \eta_e$  are normalized to sum to one:  $2b \eta_e + 1 - \eta_e = 1$ .  $2b \eta_e + 1 - \eta_e = 1$  Figure 3(b) shows the profile of the effective class prior  $\eta_e$  for different  $\eta$ ?. The graph shows that when the true class prior  $\eta$  is large,  $\eta_e$  tends to be flat around  $\eta$ ?. 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.  $\eta_e =$

We also see that when the true class prior is large, overestimation of the class prior is more attenuated. This may explain why some class-prior estimation methods [1, 11] still give a good practical performance in spite of having a positive bias.

5 Generalization error bounds for PU classification In this section, we analyze the generalization error for PU classification, when training samples are clearly not identically distributed. More specifically, we derive error bounds for the classification function  $f(x)$  of form  $f(x) =$

$$\frac{1}{n} \sum_{i=1}^n k(x_i, x) +$$

$x_1, \dots, x_n$  are positive training data and  $A = \{(x_1, y_1), \dots, (x_n, y_n)\}$  where  $y_i \in \{0, 1\}$ . Let  $\mathcal{F}$  be the set of all possible optimal solutions returned by the algorithm given some training data and test data according to  $p(x - y = +1)$  and  $p(x)$ . Then define the constants  $C = \sup_{f \in \mathcal{F}} \sum_{i=1}^n p(x_i - y_i = +1) k(x_i, x_i) + 2 \sum_{i=1}^n p(x_i) k(x_i, x_i)$  and define the function class  $F = \{f : x \mapsto \sum_{i=1}^n p(x_i - y_i = +1) k(x_i, x) + 2 \sum_{i=1}^n p(x_i) k(x_i, x)\}$ .

Let  $\ell(z)$  be a surrogate loss for the zero-one loss  $\ell_0(z) = 1 - z$ .

For any  $\epsilon > 0$ ,  $\ell(z)$  is lower bounded by  $\ell_0(z)$  and approaches  $\ell_0(z)$  as  $\epsilon$  approaches zero. Moreover, let  $e(x) = 2 \ell_0(yf(x))$  and  $e^*(x) = 2 \ell(yf(x))$ . Then we have the following theorems (proofs are provided in Appendix A). Our key idea is to decompose the generalization error as  $\mathbb{E} [e(x)] + \mathbb{E} [p(x, y) \ell(yf(x))]$ ,  $\mathbb{E} [p(x, y) \ell_0(yf(x))] = \mathbb{E} [p(x - y = +1) \ell(f)]$  where  $f := p(y = 1)$  is the true class prior of the positive class.

**Theorem 1.** Fix  $f \in \mathcal{F}$ , then, for any  $0 \leq \epsilon \leq 1$ , with probability at least  $1 - \epsilon$  over the repeated sampling of  $\{x_1, \dots, x_n\}$  and  $\{(x_1, y_1), \dots, (x_n, y_n)\}$  for evaluating the empirical error,
 
$$\mathbb{E} [e(x)] + \mathbb{E} [p(x, y) \ell(yf(x))] \leq \mathbb{E} [p(x, y) \ell_0(yf(x))] + \frac{1}{n} \sum_{j=1}^n \sum_{i=1}^n \mathbb{E} [k(x_j, x_i)] + 2 \sum_{i=1}^n p(x_i) \mathbb{E} [k(x_i, x_i)] \quad (12)$$

**Theorem 2.** Fix  $\epsilon > 0$ , then, for any  $0 \leq \epsilon \leq 1$  with probability at least  $1 - \epsilon$  over the repeated sampling of  $\{x_1, \dots, x_n\}$  and  $\{(x_1, y_1), \dots, (x_n, y_n)\}$  for evaluating the empirical error, every  $f \in \mathcal{F}$  satisfies
 
$$\mathbb{E} [e(x)] + \mathbb{E} [p(x, y) \ell(yf(x))] \leq \mathbb{E} [p(x, y) \ell_0(yf(x))] + \frac{1}{n} \sum_{j=1}^n \sum_{i=1}^n \mathbb{E} [k(x_j, x_i)] + 2 \sum_{i=1}^n p(x_i) \mathbb{E} [k(x_i, x_i)] + \frac{1}{n} \ln(2/\epsilon) + \frac{1}{n} \ln(2/\epsilon) + 2 \sum_{i=1}^n p(x_i) \mathbb{E} [k(x_i, x_i)]$$

In both theorems, the generalization error bounds are of order  $O(1/n + 1/n^2)$ . This order is optimal for PU classification where we have  $n$  i.i.d. data from a distribution



and  $n^?$  i.i.d. data from  $\mathcal{D}^?$  another distribution. The error bounds for fully supervised classification, by assuming these  $n + n^?$  data are all i.i.d., would be of order  $O(1/(n + n^?))$ . However, this assumption is unreasonable for PU classification, and we cannot train fully supervised classifiers using these  $n + n^?$  samples. Although the orders (and the losses) differ slightly,  $O(1/(n + 1/n^?))$  for PU classification is no worse than 2 times  $O(1/(n + n^?))$  for fully supervised classification (assuming  $n$  and  $n^?$  are equal). To the best of our knowledge, no previous work has provided such generalization error bounds for PU classification. The empirical error that we cannot evaluate in practice is in the left-hand side of (13), and the empirical error and confidence terms that we can evaluate in practice are in the right-hand side of (13).

7  
Table 1: Misclassification rate (in percent) for PU classification on the USPS dataset. The best, and equivalent by 95% t-test, is indicated in bold. ?

0.2	0.4	0.6	0.8	0.9	0.95	Ramp	Hinge	Ramp	Hinge	Ramp	Hinge	Ramp	Hinge
Ramp	Hinge	Ramp	Hinge	3.36	4.40	4.85	4.78	5.48	5.18	4.16	4.00	2.68	9.86
1.71	4.94	5.15	6.20	6.96	8.67	7.22	8.79	5.90	14.60	4.12	9.92	2.80	4.94
3.49	5.52	4.72	8.08	5.02	8.52	4.06	16.51	2.89	9.92	2.12	4.94	1.68	2.83
2.05	4.00	2.21	3.99	2.00	3.03	1.70	9.92	1.42	4.94	5.21	7.42	7.22	11.16
7.46	12.04	6.16	19.78	4.36	9.92	3.21	4.94	11.47	11.61	19.87	19.59	22.58	22.94
15.13	19.83	8.86	9.92	5.29	4.94	1.89	3.55	2.55	4.61	2.64	3.70	2.31	2.49
1.78	9.92	1.39	4.94	3.98	5.09	4.81	7.00	4.75	6.85	3.74	11.34	2.79	9.92
2.11	4.94	1.22	2.76	1.60	3.86	1.73	3.56	1.61	2.24	1.38	9.92	1.13	4.94

0 vs 1	0 vs 2	0 vs 3	0 vs 4	0 vs 5	0 vs 6	0 vs 7	0 vs 8	0 vs 9	3
5	5	5	5	5	5	5	5	5	5
Positive loss	Negative loss	Positive loss	Negative loss	Positive loss	Negative loss	Positive loss	Negative loss	Positive loss	Negative loss
5	5	5	5	5	5	5	5	5	5
Hinge	Ramp	Hinge	Ramp	Hinge	Ramp	Hinge	Ramp	Hinge	Ramp
Positive	Negative	Positive	Negative	Positive	Negative	Positive	Negative	Positive	Negative
0	0	0	0	0	0	0	0	0	0
x2	x2	x2	x2	x2	x2	x2	x2	x2	x2
Loss	Loss	Loss	Loss	Loss	Loss	Loss	Loss	Loss	Loss
2	0	2	0	2	0	2	0	2	0
0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1
Negative	0	?	2	?	2	?	2	?	2
?	1	?	1	?	1	?	1	?	1
0	z	0	z	0	z	0	z	0	z
1	1	1	1	1	1	1	1	1	1
(a)	Loss	functions	(a)	Loss	functions	(a)	Loss	functions	(a)
2	2	2	2	2	2	2	2	2	2
?	5	?	6	?	6	?	6	?	6
?	4	?	4	?	4	?	4	?	4

2  
 0 x  
 2  
 4  
 5 6  
 6  
 1  
 4  
 2  
 0 x1  
 2  
 4  
 6  
 5 6  
 4  
 2  
 0 x1  
 2  
 4  
 6

(b) Class prior is  $\gamma = 0.2$  (c) Class prior is  $\gamma = 0.6$  (d) Class prior is  $\gamma = 0.9$  .

Figure 4: Examples of the classification boundary for the 0? vs. 7? digits, obtained by PU learning. The unlabeled dataset and the underlying (latent) class labels are given. Since discriminant function for the hinge loss case is constant 1 when  $\gamma = 0.9$ , no decision boundary can be drawn and all negative samples are misclassified.

6

#### Experiments

In this section, we experimentally compare the performance of the ramp loss and the hinge loss in PU classification (weighting was performed w.r.t. the true class prior and the ramp loss was optimized with [12]). We used the USPS dataset, with the dimensionality reduced to 2 via principal component analysis to enable illustration. 550 samples were used for the positive and mixture datasets. From the results in Table 1, it is clear that the ramp loss gives a much higher classification accuracy than the hinge loss, especially for large class priors. This is due to the fact that the effect of the superfluous penalty term in (10) becomes larger since it scales with  $\gamma$ . When the class prior is large, the classification accuracy for the hinge loss is often close to 1. This can be explained by (10): collecting the terms for the positive expectation, we get an effective loss function for the positive samples (illustrated in Figure 4(a)). When  $\gamma$  is large enough, the positive loss is minimized, giving a constant 1. The misclassification rate becomes 1 since it is a combination of the false negative rate and the false positive rate according to the class prior. Examples of the discrimination boundary for digits 0? vs. 7? are given in Figure 4. When the class-prior is low (Figure 4(b) and Figure 4(c)) the misclassification rate of

the hinge loss is slightly higher. For large class-priors (Figure 4(d)), the hinge loss causes all samples to be classified as positive (inspection showed that  $w = 0$  and  $b = 1$ ).

7

## Conclusion

In this paper we discussed the problem of learning a classifier from positive and unlabeled data. We showed that PU learning can be solved using a cost-sensitive classifier if the class prior of the unlabeled dataset is known. We showed, however, that a non-convex loss must be used in order to prevent a superfluous penalty term in the objective function. In practice, the class prior is unknown and estimated from data. We showed that the excess risk is actually controlled by an effective class prior which depends on both the estimated class prior and the true class prior. Finally, generalization error bounds for the problem were provided. Acknowledgments MCdP is supported by the JST CREST program, GN was supported by the 973 Program No. 2014CB340505 and MS is supported by KAKENHI 23120004.

8

## 2 References

- [1] C. Elkan and K. Noto. Learning classifiers from only positive and unlabeled data. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD2008), pages 213?220, 2008.
- [2] W. Li, Q. Guo, and C. Elkan. A positive and unlabeled learning algorithm for one-class classification of remote-sensing data. *IEEE Transactions on Geoscience and Remote Sensing*, 49(2):717?725, 2011.
- [3] S. Hido, Y. Tsuboi, H. Kashima, M. Sugiyama, and T. Kanamori. Inlier-based outlier detection via direct density ratio estimation. In F. Giannotti, D. Gunopulos, F. Turini, C. Zaniolo, N. Ramakrishnan, and X. Wu, editors, Proceedings of IEEE International Conference on Data Mining (ICDM2008), pages 223? 232, Pisa, Italy, Dec. 15?19 2008.
- [4] C. Scott and G. Blanchard. Novelty detection: Unlabeled data definitely help. In Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS2009), pages 464?471, Clearwater Beach, Florida USA, Apr. 16-18 2009.
- [5] C. Elkan. The foundations of cost-sensitive learning. In Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI2001), pages 973?978, 2001.
- [6] C.C. Chang and C.J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1?27:27, 2011.
- [7] H.L. Van Trees. *Detection, Estimation, and Modulation Theory, Part I. Detection, Estimation, and Modulation Theory*. John Wiley and Sons, New York, NY, USA, 1968.
- [8] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. John Wiley & Sons, 2nd edition, 2001.
- [9] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 2000.
- [10] G. Blanchard, G. Lee, and C. Scott. Semi-supervised novelty detection. *The Journal of Machine Learning Research*, 11:2973?3009, 2010.
- [11] M. C. du Plessis and M. Sugiyama. Class

prior estimation from positive and unlabeled data. IEICE Transactions on Information and Systems, E97-D:1358–1362, 2014. [12] R. Collobert, F.H. Sinz, J. Weston, and L. Bottou. Trading convexity for scalability. In Proceedings of the 23rd International Conference on Machine learning (ICML2006), pages 201–208, 2006. [13] S. Suzumura, K. Ogawa, M. Sugiyama, and I. Takeuchi. Outlier path: A homotopy algorithm for robust SVM. In Proceedings of 31st International Conference on Machine Learning (ICML2014), pages 1098–1106, Beijing, China, Jun. 21–26 2014. [14] A. Ghosh, N. Manwani, and P. S. Sastry. Making risk minimization tolerant to label noise. CoRR, abs/1403.3610, 2014. [15] M. Mohri, A. Rostamizadeh, and A. Talwalkar. Foundations of Machine Learning. MIT Press, 2012.