Statistical Methods: Portfolio 5

By Henry Bourne

1 Analysis of Learning from Positive and Unlabeled Data [1]

This paper looks at **PU classification** the problem of learning a (binary) classifier from positive and unlabelled data. This is where we have access to labels for some of the positive data, however, no labels for the rest of our data including all the negative samples [1]. What the paper [1] covers includes:

- Casting PU classification as cost-sensitive classification.
- Exploring how convex surrogate loss functions such as the hinge loss can lead to an incorrect classification boundary, even when the classes are entirely separable. It then shows that non-convex loss functions, such as the ramp loss, lead to the correct classification boundaries.
- When the class prior in the unlabelled dataset is estimated from the data the classification error is governed by the effective class prior. Where the effective class prior depends on the true and estimated class priors. This analysis then shows that the classification error is not sensitive to class-prior estimation error if the unlabelled data has substantially more positive data.
- Generalization error bounds for PU classification are established: "For an equal number of positive and unlabeled samples, the convergence rate is no worse than $2\sqrt{2}$ times the fully supervised case" [1].

Here we will discuss the first two bullet points.

PU classification as cost-sensitive classification

The first thing we will do is show that the PU classification problem can be cast as a cost-sensitive classification problem. Let's begin by describing an **ordinary classification problem**, specifically Bayes optimal classifier (which we saw in Portfolio 1). Let Bayes optimal classifier corresponds to the decision function $f(X) \in \{1, -1\}$ that minimizes the expected misclassification rate wrt. a class prior π. Then our misclassification rate, or our risk, can be written $R(f) := \pi R_1(f) + (1-\pi)R_{-1}(f)$, where $R_1(f), R_{-1}(f)$ are the FN and FP rates, ie. $R_1(f) = \mathbb{P}(f(X) \neq 1 | y = 1)$ and $R_{-1}(f) = \mathbb{P}(f(X) \neq -1 | y = -1)$ where y is the class label. What we typically do is then calculate the empirical risk using our fully labeled data.

Now, a **cost-sensitive classification problem** again selects a function $f(X) \in \{1, -1\}$, however, now we would like to minimize the <u>weighted</u> expected misclassification rate: $R(f) := \pi c_1 R_1(f) + (1-\pi)c_{-1}R_{-1}(f)$, where c_1, c_{-1} are the costs for associated with each class. We can also interpret the costs as reweighing the problem according to new class priors proportional to πc_1 and $(1-\pi)c_{-1}$.

In **PU classification** we learn a classifier from labelled data belonging to the positive class and unlabelled data that is a mixture of samples from both the positive and negative class, which has some unknown class prior π, ie. $P_X(A) = \pi P(A|y = 1) + (1-\pi)P(A|y = -1)$. Since we don't have access to negative labels we will train the classifier to minimize the expected misclassification rate between the positive (labelled) and unlabelled samples. Since we don't have labelled negative samples we cannot directly estimate $R_{-1}(f)$ here, so instead we will rewrite $R(f)$ such that it doesn't appear. Let $R_X(f)$ be the probability that the function $f(X)$ gives a positive label over

[1]Outlier detection in unlabelled data based on labelled inlier data can also be regarded as PU classification.

P_X, then:

$$\begin{aligned} R_X(f) &= P_X(f(X) = 1) && (1)\\ &= \pi P(f(X) = 1|y = 1) + (1-\pi)P(f(X) = 1|y = -1) && (2)\\ &= \pi(1 - R_1(f)) + (1-\pi)R_{-1}(f) && (3) \end{aligned}$$

Then we have that the risk, $R(f)$, can be written as:

$$\begin{aligned} R(f) &= \pi R_1(f) + (1-\pi)R_{-1}(f) && (4)\\ &= \pi R_1(f) - \pi(1 - R_1(f)) + R_X(f) && (5)\\ &= 2\pi R_1(f) + R_X(f) - \pi && (6) \end{aligned}$$

We now will cast this to a cost-sensitive classification problem. First, let η be the proportion of labelled samples from class 1 compared to the whole dataset, which we can empirically estimate as $n/(n+n')$ where n is the number of positive samples and n' is the number of unlabeled samples. We can then express the risk as: $R(f) = c_1\eta R_1(f) + c_X(1-\eta)R_X(f) - \pi$, where $c_1 = 2\pi/\eta$ and $c_X = 1/(1-\eta)$. Comparing this with the weighted expected misclassification rate for cost-sensitive classification we wrote above we see that the PU classification problem is the same as a cost-sensitive classification problem between positive and unlabelled data with costs c_1 and c_X.

1.1 Investigating the affect of convexity of the loss function on PU classification

We now will investigate how wether or not a loss function is convex affects our solution to PU classification. But let's first consider an ordinary classification problem where instead of using our binary decision function $f(X) \in \{1, -1\}$, this time let us have a continuous decision function $g(X) \in \mathbb{R}$ such that $\text{sign}(g(X)) = f(X)$. Then our loss function becomes: $\mathcal{L}_{0,1}(g) = \pi\mathbb{E}_1(l_{0,1}(g(X))) + (1-\pi)\mathbb{E}_{-1}(l_{0,1}(-g(X)))$ where $\mathbb{E}_x(A)$ is the expectation over $P(A|y = x)$ and $l_{0,1}$ is the **zero-one loss** (outputs zero when its argument is strictly bigger than zero and one otherwise). The problem here is that the zero-one loss is hard to optimize as its discontinuous. To fix this it can be replaced by the:

Definition 1.1 ***Ramp-Loss:***
$l_R(z) = \frac{1}{2}\max(0, \min(2, 1-z))$

However, the ramp-loss is non-convex and therefore we would prefer a convex loss function as it will be easier to optimize, in practice we often use the:

Definition 1.2 ***Hinge Loss:***
$l_H(z) = \frac{1}{2}\max(1-z, 0)$

Provided the positive and negative samples are non-overlapping the hinge loss gives the the same decision boundary as the ramp loss [2].

Now, does this hold for PU classification? Let's first look at what happens when we use the ramp-loss, our loss function for PU classification becomes:

$$\begin{aligned} \mathcal{L}_{PU-R}(g) &= 2\pi R_1(f) + R_X(f) - \pi && (7)\\ &= 2\pi\mathbb{E}_1(l_R(g(X))) + (\pi\mathbb{E}_1(l_R(-g(X)))) + (1-\pi)\mathbb{E}_{-1}(l_R(-g(X))) - \pi && (8)\\ &= \pi\mathbb{E}_1(l_R(g(X))) + \pi\mathbb{E}_1(l_R(g(X)) + l_R(-g(X))) + (1-\pi)\mathbb{E}_{-1}(l_R(-g(X))) - \pi && (9) \end{aligned}$$

Now, since $l_R(-z) + l_R(z) = 1$, we have $\mathcal{L}_{PU-R}(g) = \pi\mathbb{E}_1(l_R(g(X))) + (1-\pi)\mathbb{E}_{-1}(l_R(-g(X)))$. Which is the same as the objective function for an ordinary classification problem (when using the ramp loss).

When using the hinge loss our objective function becomes:

$$\begin{aligned} \mathcal{L}_{PU-H}(g) &= 2\pi\mathbb{E}_1(l_H(g(X))) + (\pi\mathbb{E}_1(l_H(-g(X)))) + (1-\pi)\mathbb{E}_{-1}(l_H(-g(X))) - \pi && (10)\\ &= \pi\mathbb{E}_1(l_H(g(X))) + (1-\pi)\mathbb{E}_{-1}(l_H(-g(X))) + \pi\mathbb{E}_1(l_H(g(X)) + l_H(-g(X))) - \pi && (11) \end{aligned}$$

[2] As separability implies for optimal g that $l_R(z) = 0$ everywhere and for all values of z for which $l_R(z) = 0$ we have $l_H(z) = 0$.

In red is the ordinary error term, however, we also have a superfluous penalty in orange . This superfluous penalty term may cause us to obtain an incorrect classification boundary as our objective function, as a $g(X)$ that perfectly separates the data may not now minimize our objective function.

Therefore, to obtain a correct decision boundary the loss function should be symmetric and therefore non-convex [3].

2 Stein Variational Gradient Descent: A General Purpose Bayesian Inference Algorithm [2]

This paper proposes a general purpose variational inference algorithm that in empirical studies performs competitively with the existing state-of-the-art methods at the time (2019). The derivation of the method is based on a new theoretical result that connects the derivative of the KL divergence under smooth transforms with Steins identity aswell as a kernelized Stein discrepancy.

We will begin by first asking what is variational inference? **Variational inference** concerns methods that approximately infer a probability distribution. In particular we seek to infer a distribution that is close to the true posterior distribution of the model but is tractable and easy to work with. This is typically done by defining a family of distributions and then finding the member of that family that is "closest" to the true posterior in some sense.

More mathematically: we want to find a distribution q that is approximately p, ie. $q \approx p$, where p is our target distribution. Let x be a continuous random variable or parameter of interest and D be our dataset consisting of iid observations. We would like to find the posterior of x, ie. $p(x|D)$. One potential way to find the posterior would be using sampling methods such as Markov Chain Monte Carlo (MCMC) methods. With these methods we generate samples from the posterior distribution, ie. we generate $x_i \sim p(x|D)$, and then use these samples to approximate the posterior. However, these methods can be very computationally expensive, especially for high-dimensional models amongst other disadvantages.

Another way of finding the posterior is to find some q that minimizes the Kullback-Leibler divergence: $\min_{q \in F} \mathrm{KL}(q,p)$, where F is some set of distributions that we restrict ourselves to. This approach is defined by variational inference.

However, does there exist a more flexible approach? one where we are not restricted to F (is non-parametric)? let $X \sim Q$ where Q has density $q(x)$ and let $X' = X + \epsilon V(X)$, where $V \in \mathbb{R}^d \sim Q'$ has density $q'(x)$. What we are doing here is we are saying x comes from some proposal distribution Q, we are then "moving" our samples (or particles) by some magnitude ϵ in direction given to us by V in the hope that the distribution of our now moved samples is closer to that of the true distribution. To "move our samples" in the optimal direction we would like to choose V such that $\mathrm{KL}(q',p) < \mathrm{KL}(q,p)$ and the optimal V will decrease the KL divergence by the maximum amount (ie. minimize it). What we are doing in this process is carrying out gradient descent.

So, we would like to find $\min_{V,||V||=1} \frac{\partial}{\partial \epsilon}\mathrm{KL}(q',p)|_{\epsilon=0}$. First, note that by the law of the unconscious statistician,

$$\mathrm{KL}(q',p) = \mathbb{E}_{x'}\left(\log\left(\frac{q'(x')}{p(x')}\right)\right) = \mathbb{E}_{x}\left(\log\left(\frac{q(x+\epsilon V(x))}{p(x+\epsilon V(x))}\right)\right) \tag{12}$$

We then have that,

$$\mathbb{E}_x\left(\log\left(\frac{q(x+\epsilon V(x))}{p(x+\epsilon V(x))}\right)\right) = E_x\left(\frac{\log(q(x+\epsilon V))}{\log(p(x+\epsilon V))}\right) + E_x(\log q'(x+\epsilon V(x))) - E_x(\log q(x+\epsilon V(x))) \tag{13}$$

We will now state a lemma which will get rid of the added cost (the last two terms):

Lemma 2.1 $\frac{\partial}{\partial \epsilon}\mathbb{E}_x(\log \frac{q'(x+\epsilon V(x))}{q(x+\epsilon V(x))})|_{\epsilon=0} = 0$

The proof of this lemma is in A.1. We can then split the first term on the RHS in 13 and take the derivative to get:

$$\frac{\partial}{\partial \epsilon}[\mathbb{E}_x[\log q(x+\epsilon V(x))] - \mathbb{E}_x[\log p(x+\epsilon V(x))]] \tag{14}$$

[3]Note that an alternative to this would be to evaluate the superfluous penalty term and subtract it from the objective function.

First lets look at the first term above:

$$\frac{\partial}{\partial \epsilon}\mathbb{E}_x[\log q(x+\epsilon V(x))]|_{\epsilon=0} = \mathbb{E}_x[\frac{\partial}{\partial \epsilon}\log q(x+\epsilon V(x))]|_{\epsilon=0} \tag{15}$$

$$= \mathbb{E}_x\left[\frac{\frac{\partial}{\partial \epsilon}q(x+\epsilon V(x))}{q(x+\epsilon V(x))}\right]\Bigg|_{\epsilon=0} \tag{16}$$

$$= \mathbb{E}_x\left[\frac{\langle \frac{\partial}{\partial \boldsymbol{x}}q(x+\epsilon V(x)), V(x)\rangle}{q(x+\epsilon V(x))}\right]\Bigg|_{\epsilon=0} [4] \tag{17}$$

$$= \mathbb{E}_x\left[\frac{\langle \frac{\partial}{\partial \boldsymbol{x}}q(x), V(x)\rangle}{q(x)}\right] \tag{18}$$

$$= -\mathbb{E}\left[\sum_i \frac{\partial}{\partial x_i}V_i(x)\right] \tag{19}$$

For the second term we have:

$$-\frac{\partial}{\partial \epsilon}\mathbb{E}_x[\log p(x+\epsilon V(x))] = -\mathbb{E}_x\left[\frac{\langle \frac{\partial}{\partial \boldsymbol{x}}p(x+\epsilon V(x)), V(x)\rangle}{p(x+\epsilon V(x))}\right]\Bigg|_{\epsilon=0} \tag{20}$$

$$= -\mathbb{E}_x\left[\frac{\langle \frac{\partial}{\partial \boldsymbol{x}}p(x), V(x)\rangle}{p(x)}\right] \tag{21}$$

$$= -\mathbb{E}_x\left[\sum_i \frac{\partial}{\partial x_i}\log p(x)V_i(x)\right] \tag{22}$$

Where $V(x) \in \mathbb{R}^d$ with $V_i(x) = \langle V, \phi_i(x)\rangle$ with $\phi_i \in \mathbb{R}^b$. Where our $\phi_i' s$ are the feature transforms induced by a kernel we choose, this is where the Reproducing Kernel Hilbert Space (RKHS) comes in (which is mentioned many times in the paper) as its induced by the kernel. Therefore,

$$\frac{\partial}{\partial \epsilon}\mathrm{KL}(q', p) = -\mathbb{E}_x[\langle V, \sum_i \frac{\partial}{\partial x_i}\log p(x)\phi_i(x)\rangle] - \mathbb{E}_x[\langle V, \sum_i \frac{\partial}{\partial x_i}\phi_i(x)\rangle] \tag{23}$$

$$= \langle V, -\mathbb{E}_x[\sum_i \log p(x)\phi_i(x) + \sum_i \frac{\partial}{\partial x_i}\phi_i(x)]\rangle \tag{24}$$

We want to minimize $\frac{\partial}{\partial \epsilon}\mathrm{KL}(q', p)$ subject to $||V|| = 1$. So we take our optimal value for v, V^*, to be:

$$V* = \frac{\mathbb{E}_x[\sum_i \log p(x)\phi_i(x) + \sum_i \frac{\partial}{\partial x_i}\phi_i(x)]}{||\mathbb{E}_x[\sum_i \log p(x)\phi_i(x) + \sum_i \frac{\partial}{\partial x_i}\phi_i(x)]||} \tag{25}$$

Where we obtain the expectation using MCMC (note its cheaper here than it would be using it directly on the posterior). Now we can construct an algorithm:

1. $\{x_0^n\}_{n=1}^N \sim Q_0$
2. For $i = 1, ..., 100$
    (a) $x_{i+1}^n = x_i^n + \epsilon \cdot V^*(x_i^n)$

Where we find V^* as we defined above. Using this we can then perform variational inference.

[4]We get this step by using the directional derivative, that is we have by the chain rule that $\frac{\partial}{\partial \epsilon}q(x+\epsilon V(x)) = \frac{\partial q}{\partial x}\frac{\partial x}{\partial \epsilon}(x+\epsilon V(x)) = V(x)\frac{\partial q}{\partial x}(x+\epsilon V(x))$

Appendices

A Proofs

A.1 Proof of 2.1

First note that,

$$q'(x + \epsilon V(x)) = q(x) \cdot \frac{1}{|I + \epsilon \nabla V(x)|} \tag{26}$$

Where $|\cdot|$ is the determinant and $\nabla V(x)$ is the jacobian of $V(x)$. Then we have,

$$\begin{aligned}
\mathbb{E}_x \left[\log \frac{q'(x+\epsilon V(x))}{q(x+\epsilon V(x))} \right] |_{\epsilon=0} &= \mathbb{E}_x \left[\log \frac{q(x)}{q(x+\epsilon V(x)) \cdot |I + \epsilon \nabla V(x)|} \right] |_{\epsilon=0} && (27)\\
&= \mathbb{E}_x \left[\log \frac{q(x)}{q(x)} \right] - \mathbb{E}_x \left[\epsilon \cdot \langle \nabla \log q(x), V(x) \rangle + O(\epsilon^2) \right] && (28)\\
&- \mathbb{E}_x \left[\log |I + \epsilon \nabla V(X)| \right] && (29)
\end{aligned}$$

Now taking $\frac{\partial}{\partial \epsilon}$ and evaluating at $\epsilon = 0$, we get,

$$\begin{aligned}
&= 0 - \mathbb{E}_x[\langle \nabla \log q(x), V(x) \rangle] - \frac{\partial}{\partial \epsilon} \mathbb{E}_x[\log |I + \epsilon \nabla V(x)|]|_{\epsilon=0} && (30)\\
&= -\mathbb{E}_x[\langle \nabla \log q(x), V(x) \rangle] - \mathbb{E}_x[\mathrm{tr}(I^{-1} \nabla V(x))] && (31)\\
&= \mathbb{E}_x[\sum_i \frac{\partial}{\partial \boldsymbol{x}} V_i(x)] - \mathbb{E}_x[\sum_i \frac{\partial}{\partial \boldsymbol{x}} V_i(x)] && (32)\\
&= 0 && (33)
\end{aligned}$$

References

[1] Marthinus C Du Plessis, Gang Niu, and Masashi Sugiyama. Analysis of learning from positive and unlabeled data. *Advances in neural information processing systems*, 27, 2014.

[2] Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. *Advances in neural information processing systems*, 29, 2016.