1 Analysis of Learning from Positive and Unlabeled Data

In this paper [2] by du Plessis et al., the authors analyse the problem of trying to learn a binary classifier of positive and negative labels given only positively-labelled and unlabelled data points. An example application is given: discriminating between built-up urban areas (with positive labels) and rural areas (negative labels), the latter being much more diverse therefore unlikely to be fully labelled. The authors determine that non-convex loss functions (such as *ramp loss*) lead to better results than convex loss functions (such as *hinge loss*) and compare learning from positive and unlabelled data to fully supervised learning.

1.1 PU Classification

First we should recall the basics of binary classification. Suppose that we have a class prior π and marginal probabilities of positive and negative samples P_1 and P_{-1} respectively. Then note that the *Bayes optimal classifier* is defined by the decision function $f(X) \in \{-1, 1\}$ that minimises the expected misclassification rate

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

where $R_{-1}(f)$ denotes the expected false positive rate

$$R_{-1}(f) = P_{-1}(f(X) \neq -1)$$

and $R_1(f)$ denotes the expected false positive rate

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

We can extend this to a *cost-sensitive classifier* by choosing per-class costs c_1 and c_{-1}, in which case we attempt to minimise the weighted expected misclassification rate:

$$R(f) := \pi c_1 R_1(f) + (1-\pi)c_{-1}R_{-1}(f). \tag{1}$$

Now we consider PU (positive and unlabelled) classification, in which we have data from P_1 and from a mixture probability from positive and negative samples with an unknown class prior π:

$$P_X = \pi P_1 + (1-\pi)P_{-1}.$$

Denoting the true labels as Y we may consider this as a marginalisation where

$$P_X = \sum_{y \in \{-1,1\}} \mathbb{P}(Y=y)\mathbb{P}(X|Y=y).$$

We would like to write the risk $R(f)$, but since we have no negatively-labelled data we cannot directly estimate R_{-1}. Instead, we consider $R_X(f)$; the probability that the function $f(X)$ gives the positive label over P_X:

$$\begin{aligned} R_X(f) &= P_X(f(X)=1) \\ &= \pi P_1(f(X)=1) + (1-\pi)P_{-1}(f(X)=1) \\ &= \pi(1-R_1(f)) + (1-\pi)R_{-1}(f). \end{aligned}$$

Thus we can rewrite the risk $R(f)$ by substituting $-\pi(1-R_1(f)) + R_X(f)$ for $(1-\pi)R_{-1}(f)$:

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

If we have n positively labelled samples and n' unlabelled samples, we can estimate the proportion of samples from P_1 compared to P_X as $\eta = \frac{n}{n+n'}$. This then allows us to express the risk $R(f)$ as

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

where

$$c_1 = \frac{2\pi}{\eta} \text{ and } c_X = \frac{1}{1-\eta}.$$

That is, we can solve PU classification through cost-sensitive analysis with costs c_1 and c_X (as in 1).

1.2 Loss Functions

Suppose now that $f(X)) = \text{sign}(g(X)) \in \{-1, 1\}$ for some continuous decision function $g(X) \in \mathbb{R}$. We then obtain the loss function

$$J_{0-1}(g) = \pi\mathbb{E}_1[l_{0-1}(g(X))] + (1-\pi)\mathbb{E}_{-1}[l_{0-1}(-g(X))] \tag{2}$$

where $\mathbb{E}_1$ and $\mathbb{E}_{-1}$ are the expectations over P_1 and P_{-1} respectively and $l_{0-1}(z)$ is the zero-one loss:

$$l_{0-1}(z) = \begin{cases} 0 & z > 0, \\ 1 & z \leq 0. \end{cases}$$

However, $l_{0-1}(z)$ has a discontinuity at $z = 0$ and so can be difficult to optimize. Instead we might consider two alternatives: *ramp loss* l_r

$$l_r(z) = \frac{1}{2}\max(0, \min(2, 1-z)) = \begin{cases} 1 & z \leq -1, \\ \frac{1}{2} - z & -1 < z < 1, \\ 0 & z \geq 1, \end{cases}$$

and *hinge loss* l_h

$$l_h(z) = \frac{1}{2}\max(1-z, 0) = \begin{cases} \frac{1}{2} - z & z < 1, \\ 0 & z \geq 1. \end{cases}$$

The first of these leads to the ramp-loss risk:

$$\begin{aligned} R_r(g) &= 2\pi R_1(f) + R_X(f) - \pi \\ &= 2\pi\mathbb{E}_1[l_r(g(X))] + [\pi\mathbb{E}_1[l_r(-g(X))] + (1-\pi)\mathbb{E}_{-1}[l_r(-g(X))]] - \pi \\ &= \pi\mathbb{E}_1[l_r(g(X))] + \pi\mathbb{E}_1[l_r(g(X)) + l_r(-g(X))] + (1-\pi)\mathbb{E}_{-1}[l_r(-g(X))] - \pi \\ &= \pi\mathbb{E}_1[l_r(g(X))] + \pi + (1-\pi)\mathbb{E}_{-1}[l_r(-g(X))] - \pi \\ &= \pi\mathbb{E}_1[l_r(g(X))] + (1-\pi)\mathbb{E}_{-1}[l_r(-g(X))]. \end{aligned}$$

(Here we have used the fact that $l_r(z) + l_r(-z) = 1$.) Note that this is essentially the same as 2, and so employing ramp loss will lead to the same decision boundary as in ordinary classification. The hinge-loss risk, however, includes an extra penalty that may cause incorrect classification:

$$\begin{aligned} R_h(g) &= 2\pi R_1(f) + R_X(f) - \pi \\ &= 2\pi\mathbb{E}_1[l_h(g(X))] + [\pi\mathbb{E}_1[l_h(-g(X))] + (1-\pi)\mathbb{E}_{-1}[l_h(-g(X))]] - \pi \\ &= \pi\mathbb{E}_1[l_h(g(X))] + \underbrace{\pi\mathbb{E}_1[l_h(g(X)) + l_h(-g(X))]}_{\text{extra (unnecessary) penalty}} + (1-\pi)\mathbb{E}_{-1}[l_h(-g(X))] - \pi. \end{aligned}$$

Hence the authors conclude this section by noting that the loss function should be symmetric (and therefore non-convex), before moving onto further PU classification analysis and numerical examples.

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

This paper [1] by Liu and Wang presents a method for learning a probability distribution by minimising the KL divergence between the learnt distribution and the target distribution through gradient descent. We will first discuss regular gradient descent before examining how the authors of this paper apply it to variational inference.

2.1 Gradient Descent

In gradient descent we minimise a function $L(\theta)$ via iterative updates of the parameters:

$$\theta_{t+1} = \theta_t - \eta \nabla L(\theta_t) \tag{3}$$

with some positive learning rate $\eta > 0$. Note that we can derive this by considering a general update of the form

$$\theta_{t+1} = \theta_t - \eta v$$

for some unit vector $v \in \mathbb{R}^d$. If we want to find an optimal $\hat{v}$ that maximises the decrease in L (whilst keeping $||v|| = 1$) we see that:

$$\begin{aligned}\hat{v} = \underset{v:||v||=1}{\operatorname{argmin}} \partial_\varepsilon L(\theta_t + \varepsilon v)|_{\varepsilon=0} = \underset{v:||v||=1}{\operatorname{argmin}} \langle \nabla L(\theta_t + \varepsilon v), v\rangle|_{\varepsilon=0} &= \underset{v:||v||=1}{\operatorname{argmin}} \langle \nabla L(\theta_t), v\rangle \\ &= \frac{-\nabla L(\theta_t)}{||\nabla L(\theta_t)||}.\end{aligned}$$

This agrees with our update rule in 3 up to a normalising factor.

2.2 Stein Variational Gradient Descent

In regular variational inference we attempt to estimate a target distribution p by a distribution q from some (often parameterised) family of distributions F by minimising the KL divergence between the two distributions, that is, we attempt to find $\operatorname{argmin}_{q \in F} KL[q||p]$. The choice of F here is very important and can be very restrictive when it comes to successfully estimating p, in particular MCMC methods can become very expensive depending on F, so instead the authors suggest representing q by a number of particles and using gradient descent to iteratively move these particles into the shape of p.

We may derive this version of gradient descent very similarly to the derivation in the previous section. First suppose $X \sim Q$ (with $X \in \mathbb{R}^d$) has density $q(X)$ and note that we shall update X in the form

$$X' = X + \varepsilon V(X)$$

where $V \in \mathbb{R}^d$ is a unit vector and we say $X' \sim Q'$ has density $q'(X)$. As before, we want to choose the $\hat{V}(X)$ that will result in a maximal decrease in the KL divergence, i.e.:

$$\hat{V}(X) = \underset{V:||V||=1}{\operatorname{argmin}} \partial_\varepsilon KL[q'||p]|_{\varepsilon=0}.$$

We may rewrite the KL divergence above as

$$\begin{aligned} KL[q'||p] = \mathbb{E}_{X'}\left[\log \frac{q'(X')}{p(X')}\right] &= \mathbb{E}_X\left[\log \frac{q'(X+\varepsilon V(X))}{p(X+\varepsilon V(X))}\right] \\ &= \mathbb{E}_X\left[\log\left(\frac{q'(X+\varepsilon V(X))}{p(X+\varepsilon V(X))} \cdot \frac{q(X+\varepsilon V(X))}{q'(X+\varepsilon V(X))} \cdot \frac{q'(X+\varepsilon V(X))}{q(X+\varepsilon V(X))}\right)\right] \\ &= \mathbb{E}_X\left[\log\left(\frac{q(X+\varepsilon V(X))}{p(X+\varepsilon V(X))}\right) + \log q'(X+\varepsilon V(X)) - \log q(X+\varepsilon V(X))\right]. \end{aligned}$$

In Appendix A we show that $\partial_\varepsilon \mathbb{E}_X \left[\log q'(X+\varepsilon V(X)) - \log q(X+\varepsilon V(X))\right]|_{\varepsilon=0} = 0$, which allows us to observe that

$$\partial_\varepsilon KL[q'||p]|_{\varepsilon=0} = \partial_\varepsilon \left(\mathbb{E}_X[\log q(X+\varepsilon V(X))] - \mathbb{E}_X[\log p(X+\varepsilon V(X))]\right)|_{\varepsilon=0}.$$

The first term in this can be rewritten as

$$\begin{aligned}
\partial_\varepsilon \left(\mathbb{E}_X[\log q(X+\varepsilon V(X))]\right)|_{\varepsilon=0} &= \mathbb{E}_X[\partial_\varepsilon \log q(X+\varepsilon V(X))]|_{\varepsilon=0} \\
&= \mathbb{E}_X\left[\frac{\partial_\varepsilon q(X+\varepsilon V(X))}{q(X+\varepsilon V(X))}\right]\Bigg|_{\varepsilon=0} \\
&= \mathbb{E}_X\left[\frac{\langle \nabla q(X+\varepsilon V(X)), V(X)\rangle}{q(X+\varepsilon V(X))}\right]\Bigg|_{\varepsilon=0} \\
&= \mathbb{E}_X\left[\frac{\langle \nabla q(X), V(X)\rangle}{q(X)}\right] \\
&= -\mathbb{E}_X\left[\sum_i \partial_{X_i} V_i(X)\right],
\end{aligned}$$

where the last equality is achieved via integration by parts and through the definition of expectation. We may similarly rewrite the second term to obtain

$$\begin{aligned}
-\partial_\varepsilon \left(\mathbb{E}_X[\log p(X+\varepsilon V(X))]\right)|_{\varepsilon=0} &= -\mathbb{E}_X\left[\frac{\langle \nabla p(X+\varepsilon V(X)), V(X)\rangle}{p(X+\varepsilon V(X))}\right]\Bigg|_{\varepsilon=0} \\
&= -\mathbb{E}_X\left[\frac{\langle \nabla p(X), V(X)\rangle}{p(X)}\right] \\
&= -\mathbb{E}_X\left[\sum_i \partial_{X_i} \log p(X) V_i(X)\right].
\end{aligned}$$

(The $\log p(X)$ occurs during integration by parts since the expectation is with respect to $X \sim Q$ and so we cannot cancel out the $p(X)$ term as we did before with the $q(X)$ term.) If we now model $V(X) \in \mathbb{R}^d$ via $V_i(X) := \langle v, \phi_i(X)\rangle$ for some $v, \phi_i(X) \in \mathbb{R}^b$ then

$$\begin{aligned}
\partial_\varepsilon KL[q'||p]|_{\varepsilon=0} &= -\mathbb{E}_X\left[\langle v, \sum_i \partial_{X_i} \log p(X)\phi_i(X)\rangle\right] - \mathbb{E}_X\left[\langle v, \sum_i \partial_{X_i}\phi_i(X)\rangle\right] \\
&= \left\langle v, -\mathbb{E}_X\left[\sum_i \partial_{X_i}\log p(X)\phi_i(X) + \sum_i \partial_{X_i}\phi_i(X)\right]\right\rangle.
\end{aligned}$$

Finally, note that this is minimised (with $||v|| = 1$) at

$$v* = \frac{\mathbb{E}_X[\sum_i \partial_{X_i}\log p(X)\phi_i(X) + \sum_i \partial_{X_i}\phi_i(X)]}{||\mathbb{E}_X[\sum_i \partial_{X_i}\log p(X)\phi_i(X) + \sum_i \partial_{X_i}\phi_i(X)]||}. \tag{4}$$

Hence this leads into the algorithm presented in the paper, where an initial set of n particles $\{x_i^0\}_{i=1}^n$ gradually change to approximate the target distribution's density function $p(x)$ via a series of updates (indexed by t) given by:

$$x_j^{t+1} \leftarrow x_j^t + \eta_t \cdot \frac{1}{n}\sum_i \left[\partial_{x_i^t}\log p(x_i^t)\phi_i(x_j^t) + \partial_{x_i^t}\phi_i(x_j^t)\right]$$

where η_t is the step size at the t-th iteration (and we omit the normalising factor in 4).

The authors go on to analyse the algorithm's complexity and evaluate its performance through numerical experiments against similar algorithms, through which it is shown to be a powerful and scalable tool for variational Bayesisan inference.

A Result For The Rewritten KL Divergence

Here we show that $\partial_\varepsilon \mathbb{E}_X \left[\log q'(X+\varepsilon V(X)) - \log q(X+\varepsilon V(X))\right]|_{\varepsilon=0} = 0$.

To see this, first note that (with $|\cdot|$ representing the determinant)[1]

$$q'(X+\varepsilon V(X)) = \frac{q(X)}{|\mathbf{I}+\varepsilon\cdot\nabla V(X)|}.$$

Hence we may rewrite the relevant expectation as

$$\begin{aligned}
\mathbb{E}_X\left[\log\frac{q'(X+\varepsilon V(X))}{q(X+\varepsilon V(X))}\right] =& \mathbb{E}_X\left[\log\frac{q(X)}{q(X+\varepsilon V(X))\cdot|\mathbf{I}+\varepsilon\cdot\nabla V(X)|}\right]\\
=& \mathbb{E}_X\left[\log\frac{q(X)}{q(X)}\right] - \mathbb{E}_X\left[\varepsilon\cdot\langle\nabla\log q(X), V(X)\rangle + o(\varepsilon^2)\right]\\
&- \mathbb{E}_X\left[\log|\mathbf{I}+\varepsilon\cdot\nabla V(X)|\right]\\
=& -\mathbb{E}_X\left[\varepsilon\cdot\langle\nabla\log q(X), V(X)\rangle + o(\varepsilon^2)\right] - \mathbb{E}_X\left[\log|\mathbf{I}+\varepsilon\cdot\nabla V(X)|\right].
\end{aligned}$$

From this we may deduce the desired result:

$$\begin{aligned}
\partial_\varepsilon\mathbb{E}_X\left[\log\frac{q'(X+\varepsilon V(X))}{q(X+\varepsilon V(X))}\right]\Bigg|_{\varepsilon=0} &= -\mathbb{E}_X\left[\langle\nabla\log q(X), V(X)\rangle\right]] - \partial_\varepsilon\mathbb{E}_X\left[\log|\mathbf{I}+\varepsilon\cdot\nabla V(X)|\right]\\
&= -\mathbb{E}_X\left[\langle\nabla\log q(X), V(X)\rangle\right]] - \mathbb{E}_X\left[tr[\nabla V(X)]\right]\\
&= +\mathbb{E}_X\left[\sum_i \partial_i V_i(X)\right] - \mathbb{E}\left[\sum_i \partial_i V_i(X)\right]\\
&= 0.
\end{aligned}$$

References

[1] Qiang Liu and Dilin Wang. Stein Variational Gradient Descent: A General Purpose Bayesian Inference Algorithm, September 2019. arXiv:1608.04471 [cs, stat].

[2] Marthinus C. du Plessis, Gang Niu, and Masashi Sugiyama. Analysis of learning from positive and unlabeled data. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'14, pages 703–711, Cambridge, MA, USA, December 2014. MIT Press.

[1] See, for example, https://statproofbook.github.io/P/pdf-invfct .