细化了很多PU问题的细则，尤其是PU的训练损失目标。

LOSS是什么

正常的PU损失是;

$$R = \mathbb{E}_{(\boldsymbol{x},y)\sim p(\boldsymbol{x},y)}\left[\mathbb{1}\left(\hat{y},y\right)\right], \tag{4}$$

如果知道了先验概率π_P和π_N 损失就可以写作如下形式：

$$R = \pi_P\mathbb{E}_{\boldsymbol{x}\sim p_P(\boldsymbol{x})}\left[\mathbb{1}(\hat{y},1)\right] + \pi_N\mathbb{E}_{\boldsymbol{x}\sim p_N(\boldsymbol{x})}\left[\mathbb{1}(\hat{y},0)\right]. \tag{7}$$

拆开长这样：

$$\begin{aligned} R =& \pi_P\mathbb{E}_{\boldsymbol{x}\sim p_P(\boldsymbol{x})}\left[1-\hat{y}\right] + \pi_N\mathbb{E}_{\boldsymbol{x}\sim p_N(\boldsymbol{x})}\left[\hat{y}\right], \\ =& \pi_P \underbrace{\left|\mathbb{E}_{\boldsymbol{x}\sim p_P(\boldsymbol{x})}\left[\hat{y}\right]-1\right|}_{R_P} + \pi_N \underbrace{\left|\mathbb{E}_{\boldsymbol{x}\sim p_N(\boldsymbol{x})}\left[\hat{y}\right]\right|}_{R_N}, \end{aligned} \tag{8}$$

前面后面两部分分别为正样本上损失和负样本上损失，分别赋予不同的权重，组合成PU问题的损失。

带入公式化简：

$$\begin{aligned} & \mathbb{E}_{\boldsymbol{x}\sim p_U(\boldsymbol{x})}\left[\hat{y}\right] - \mathbb{E}_{(\boldsymbol{x},y)\sim p(\boldsymbol{x},y)}\left[y\right] \\ =& \pi_P\mathbb{E}_{\boldsymbol{x}\sim p_P(\boldsymbol{x})}\left[\hat{y}\right] - \pi_P + \pi_N\mathbb{E}_{\boldsymbol{x}\sim p_N(\boldsymbol{x})}\left[\hat{y}\right] \\ =& -\pi_P R_P + \pi_N R_N. \end{aligned} \tag{9}$$

这样我们就可以巧妙的将负样本损失隐式转换为未标记样本上的损失了。当然这些前提都是建立在数据集的分布符合先验概率的基础上。

$$\begin{aligned}\pi_N R_N =& \pi_P R_P + \mathbb{E}_{\boldsymbol{x}\sim p_U(\boldsymbol{x})}\left[\hat{y}\right] - \mathbb{E}_{(\boldsymbol{x},y)\sim p(\boldsymbol{x},y)}\left[y\right] \\ =& \pi_P R_P + \mathbb{E}_{\boldsymbol{x}\sim p_U(\boldsymbol{x})}\left[\hat{y}\right] - \pi_P. \end{aligned} \tag{10}$$

$$R = 2\pi_P R_P + \mathbb{E}_{\boldsymbol{x}\sim p_U(\boldsymbol{x})}\left[\hat{y}\right] - \pi_P. \tag{11}$$

后面那块由于为了保证未标记样本训练效果需要加上绝对值。加上绝对值之后我们不妨认为损失由前部分和后部分R_N组成，称为未标记样本损失。

LOSS的形式

理想中的PU问题损失应当是：

$$R = \mathbb{E}_{(\boldsymbol{x},y)\sim p(\boldsymbol{x},y)}\left[\mathbb{1}\left(\hat{y}, y\right)\right], \tag{4}$$

其中E函数为：

$E_{(x,y)\ p(x,y)}[1(\hat{y}, y)] = \hat{y} \oplus y$

但是这个函数不可微，在正常网络训练过程中无法进行优化。于是我们发明新的方法将损失函数转化为可以引用于损失的反向传播的形式。

我们想到sigmoid函数：

$$s = \frac{1}{1 + \exp\left[-f(\boldsymbol{x})\right]}. \tag{14}$$

至此就可以将模型输出的logits与先前的损失函数联系，将形式转换为可以反向传播的形式：

$$\hat{R}_{lab} = 2\pi_P \underbrace{\left| \frac{1}{n_L} \sum_{\boldsymbol{x} \in \boldsymbol{X_L}} s - 1 \right|}_{\hat{R}_L} + \underbrace{\left| \frac{1}{n_U} \sum_{\boldsymbol{x} \in \boldsymbol{X_U}} s - \pi_P \right|}_{\hat{R}_U}, \tag{15}$$

同时我们可以看到该方程有一个平凡解--如果对于任何模型输出$s == \pi_P$,那么后面的$\widehat{R}_U$就会一直为学不动。为了规避这一点我们希望sigmoid函数的输出是一个非常接近0或者1的值，因此引入熵最小化的概念。

在损失中增加训练目标;

$$L_{ent} = -\frac{1}{n_U} \sum_{\boldsymbol{x} \in \boldsymbol{X_U}} \left[(1-s) \log (1-s) + s \log s \right]. \tag{17}$$

除了最小化常规损失，还要令L_{ent}的值尽量大，确保s是一个接近0或1的值。

应对label bias

如果给样本打上的标签是错误的，在训练过程中由于一轮一轮的拟合以及熵最小化等技术的引入，会导致最初的训练误差不断积累，最后的误差会非常大。为了应对label bias我们引入mixup技术。

首先我们规定一个参数λ是一个遵循β 分布的超参数，并且规定λ'是一个由λ决定的参数：

$$\lambda' = \max(\lambda, 1-\lambda),$$

现在对于从原本的PU样本中的样本点x_1和x_2我们利用这些参数将他们线性插值为新的一个样本点：

$$\boldsymbol{x'} = \lambda' \boldsymbol{x_1} + (1-\lambda') \boldsymbol{x_2}. \tag{20}$$

针对新的样本点，我们对他的损失函数也做修改。对于新的经过融合之后的样本点，他的损失函数计算方法是;

$$L_{mix} = \frac{1}{n} \sum_{\boldsymbol{x_1}, \boldsymbol{x_2} \in \boldsymbol{X_{PU}}} [\lambda' l_{bce}(s', s_1) + (1 - \lambda') l_{bce}(s', s_2)], \quad (21)$$

可以看做是两个样本的交叉熵分别赋值权重的和。s'是组合样本的分数，$s_1 s_2$是两个样本点的原始分数。

对于之前所说的熵限制函数我们自然也可以做出更新：

$$L'_{ent} = -\frac{1}{n} \sum_{\boldsymbol{x_1}, \boldsymbol{x_2} \in \boldsymbol{X_{PU}}} l_{bce}(s', s'). \quad (22)$$

最后将前面的总体PU损失，熵最小化损失，与经过mixup之后的样本的熵最小化条件，以及mixup样本的损失结合起来就是dist-pu：

$$L_{dist} = \hat{R}_{lab} + \mu L_{ent} + \nu L_{mix} + \gamma L'_{ent}, \quad (23)$$

中间的都是权重。