不平衡的标签

Zhexiang ZHANG

2025 年 11 月 28 日

Table of Contents

Table of Contents

What for?

The selection of presentations features carefully chosen publications that demonstrate **SOLID** research, **STRONG** baseline performance, and **SIMPLE BUT NON-TRIVIAL** mathematics, with the collective aim of deepening the understanding of machine learning concepts in medical area.

Table of Contents

长尾分布

类别不平衡的是长尾分布的特征。在这种数据上做分类，需要帮助模型对每个类都学习到合理的表示:

$$z = f(x;\theta), g(z) = W^T z + b$$

Note

从直觉上来说，不均衡的分布会导致模型在不同的分类上表现不同。实际上这里有两个可能得潜在因素：（1）不均衡的分布导致编码器无法学习到好的特征，隐空间性质很差（2）不均衡的分布导致分类头的性质很差

论文结论

论文认为，不均衡的分布不会导致编码器无法学习到好的特征。（这句话的意思不是说编码器能够在少样本上学到与多样本差不多的高质量的特征；而是说编码器在少样本上特征提取的表现比较鲁棒。）论文认为，更大的问题出现在分类头上，在传统的交叉熵损失中:

$$\mathcal{L}_{\mathrm{CE}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log(\hat{y}_{i,c})$$

作者认为，数量占比显著的样本会在反向传播中收到更多的梯度更新。比如如果$c=1$这一类占据了采样的大部分，那么大部分梯度$-\nabla\log(w_1^T z)$都会被传递到w_1上去。

论文结论

作者认为，大部分梯度都会被传递到w_1上去的结果是，w_1对分类边界产生了显著的影响力。论文使用了不同类的长度$\|\mathbf{w_i}\|_2, 1 \leq i \leq C$来量化这种影响力。作者发现$\|\mathbf{w_i}\|_2$ 与类的样本数n_i呈负相关关系。基于这种观察，作者给出了一些实验。

Note

然而后来的文献证明这与优化器有关。

基于采样

假设我们有一个长尾分布的采样$\mathcal{S} = \{x_n, y_n\}_{n=1}^{N}$有$L$个类别。用 n_j表示类别为j的样本数量，也就是$\sum_{j=1}^{L} n_j = N$。某个类别j被采样的概率被修改成：

$$p_j = \frac{n_j^q}{\sum_{i=1}^{L} n_i^q}, q \in [0, 1]$$

可以看到，$q = 1$的时候是instance-balanced采样，$q = 0$的时候是class-balanced采样，$q = 0.5$ 是square-root采样。也有根据epoch进行差值的方法。

Note

论文指出这种方法得到的编码器泛化性差于使用instance-balanced采样方法得到的编码器。

Focal Loss

我们基于前面的结果来理解一下Focal Loss。我们不希望分类头中的权重出现有大有小的分布,为了阻止这种情况（大的权重通常意味着分类器的过度自信，得到了过大的权重），我们可以给过大的权重施加惩罚。观察Focal Loss当中的第一项：

$$-\alpha_1(1-\log(\hat{y}_{1,c}))^{\gamma}\log(\hat{y}_{1,c})$$

越自信（$\log(\hat{y}_{1,c})\to 0$）的样本，要遏制他的损失 $(1-\log(\hat{y}_{1,c}))^{\gamma}\to 1$，减少向它的梯度传递，让他与越不自信的样本的梯度的量相近。这样训出来的模型的权重相较之前均一了一些，理论上在长尾上的表现会更好。

尚未验证

Focal Loss在有噪声的标注环境中表现很差。这可能是模型被强迫去学习噪声导致的。

Baselines

论文提出了一些Strong Baseline：

- Classifier Retraining：E2E使用instance-balanced sampling炼出一个Encoder之后，冻住Encoder使用class-balanced sampling重炼分类头
- Nearest Class Mean Classifier:计算每个类别的特征平均，然后使用cos-similarity做分类。
- τ-normalized Classifier: 使用$\frac{w_i}{\|w_i\|_2^\tau}$ normalize参数后再送入分类头。可以使用trainable的$\tau \in [0, 1]$使用cross-validation选择。
- Learnable Weight Scaling: 对每一个w_i学一个缩放参数f_i。

Table of Contents

问题定义

设有一个样本空间$\mathcal{X}$与一个离散的标签空间$\mathcal{Y} = \{1, \cdots, L\}$。 $\mathbb{P}$是某个未知的定义在$\mathcal{X} \times \mathcal{Y}$上的分布。设有从$\mathbb{P}$中的$N$次采样：

$$S = \{(x_n, y_n)\}_{n=1}^{N} \sim \mathbb{P}^N$$

，长尾分布是指，分布$\mathbb{P}(y) = \int_x \mathbb{P}(x, y)dx$是极其不均衡的。

问题定义

设有一个模型$f : \mathcal{X} \to \mathbb{R}^L$从$S$中学习分布$\mathbb{P}(y|x)$。$f(x;\theta)$表示一个长为$L$的logits，$f_y(x;\theta)$表示label为$y$的单个logit。随后通过softmax得到分布。通常，我们只关心f在原始分布$\mathbb{P}$上的准确率：

$$ACC(f) = \mathbb{P}_{x,y}(y \notin \arg\max_{y' \in \mathcal{Y}} f_{y'}(x))$$

我们有时候不关心模型在原始分布上的准确率，而是关心他在均衡分布测试集上的准确率：

$$BER(f) = \frac{1}{L} \sum_{y \in \mathcal{Y}} \mathbb{P}_{x|y}(y \notin \arg\max_{y' \in \mathcal{Y}} f_{y'}(x))$$

Example

	Class 1	Class 2	Class 3
# of Instances	10	20	30
# of Correctly Classified	1	10	20

均衡分布

回忆一下先前的记号，$\mathbb{P}$是长尾的，而$\mathbb{P}^{\texttt{bal}}$是它均衡分布版本。更一般地说：

$$\mathbb{P}^{\texttt{bal}}(x, y) = \frac{1}{L}\mathbb{P}(x|y)$$

我们可以看到，正是由于采样S来自不均衡的分布，模型学习的是：

$$f^* \leftarrow \arg\max_f \mathbb{P}_{(x,y)\in S, S\sim\mathbb{P}}(y \notin \arg\max_{y'\in\mathcal{Y}} f_{y'}(x))$$

但是我们更关心模型在另一个均衡分布上的结果：

$$BER(f^*) = \mathbb{P}_{(x,y)\in V, V\sim\mathbb{P}^{\texttt{bal}}}(y \notin \arg\max_{y'\in\mathcal{Y}} f^*_{y'}(x))$$

从而导致了bias。

Note

S是训练集，V是验证集。两者采样方式不同，才从根本上导致了前者学习的模型在BER下表现不佳。

主定理

可以看到，我们的最终目标是，从一个$(x, y) \in S, S \sim \mathbb{P}$这样一个头重脚轻的数据集中学习到该目标：

$$f^* \leftarrow \arg\max_f BER(f)$$

论文给出的最重要的结果是，这个在均衡验证集上表现最佳的f^*满足以下性质：

$$\arg\max_{y \in \mathcal{Y}} f_y^*(x) = \arg\max_{y \in \mathcal{Y}} \mathbb{P}^{\mathtt{bal}}(y|x)$$

Note

上式左边是最优f^*对输入x的标签预测结果，右边是等待建模的分布。

主定理

回忆一下，网络曾经的建模对象是$\mathbb{P}(y|x)$。由上文的结论，为了最优化$BER(f)$，我们需要建模的对象是：

$$\arg\max_{y\in\mathcal{Y}} f_y^*(x) = \arg\max_{y\in\mathcal{Y}} \mathbb{P}^{\mathtt{bal}}(y|x)$$

在arg max的意义下：

$$\mathbb{P}^{\mathtt{bal}}(y|x) = \frac{\mathbb{P}^{\mathtt{bal}}(x,y)}{\mathbb{P}^{\mathtt{bal}}(x)} \propto \mathbb{P}^{\mathtt{bal}}(x,y) = \frac{1}{L}\mathbb{P}(x|y)$$

$$\propto \mathbb{P}(x|y) = \frac{\mathbb{P}(y|x)\mathbb{P}(x)}{\mathbb{P}(y)} \propto \frac{\mathbb{P}(y|x)}{\mathbb{P}(y)}$$

建模方式

由之前的结果

$$\arg\max_{y\in\mathcal{Y}} f_y^*(x) = \arg\max_{y\in\mathcal{Y}} \mathbb{P}^{\texttt{bal}}(y|x) = \arg\max_{y\in\mathcal{Y}} \frac{\mathbb{P}(y|x)}{\mathbb{P}(y)}$$

我们可以知道存在两种建模方式，分别是让网络拟合$\mathbb{P}^{\texttt{bal}}(y|x)$或$\frac{\mathbb{P}(y|x)}{\mathbb{P}(y)}$。

建模$\mathbb{P}(y|x)$

第一种办法是建模$\mathbb{P}(y|x)$.

$$\arg\max_{y\in\mathcal{Y}} \mathbb{P}^{\texttt{bal}}(y|x) = \arg\max_{y\in\mathcal{Y}} \frac{\mathbb{P}(y|x)}{\mathbb{P}(y)} = \arg\max_{y\in\mathcal{Y}} \frac{softmax_y(f(x;\theta))}{\pi_y}$$

其中，f是我们从长尾分布中训练出来的模型（logits），
$\pi_y = \int_x \mathbb{P}(x,y)dx$是标签为$y$的样本的分布。在实际中，我们可以引入$\tau > 0$（类似蒸馏中的温度参数）让最后一项变成$\frac{softmax_y(f(x;\theta))}{\pi_y^\tau}$。

建模$\mathbb{P}(y|x)$

Bias Correction

这立刻就引出了最好的模型，首先我们从长尾分布中训练出来模型预测$\mathbb{P}(y|x)$，设最后一层的分类头输出的logis 是$W^T\Phi(x)$，原本的预测是

$$\arg\max_{y\in\mathcal{Y}} f_y(x;\theta) = \arg\max_{y\in\mathcal{Y}}(\exp\left(w_y^T\Phi(x)\right))$$

现在为了在均衡数据集上得到最好的效果，只需要在执行预测的时候修改分类头即可：

$$\arg\max_{y\in\mathcal{Y}} \frac{softmax_y(f(x;\theta))}{\pi_y^\tau} = \arg\max_{y\in\mathcal{Y}}(\exp\left(w_y^T\Phi(x)\right) - \tau\log(\pi_y))$$

前面的论文中给分类头施加约束（如正则化权重）、修改分类边界的思想与此处是一致的，不过此处是严格的推导结果，证明了具体需要进行如何的修改才能得到最优的结果。

关于温度参数

In principle, the outputs of a sufficiently high-capacity neural network aim to mimic these probabilities. In practice, these estimates are often uncalibrated [Guo et al., 2017]. One may thus need to first calibrate the probabilities before applying logit adjustment. Temperature scaling is one means of doing so, and is often used in the context of distillation [Hinton et al., 2015]. One may treat τ as a tuning parameter to be chosen based on some measure of holdout calibration, e.g., the expected calibration error [Murphy and Winkler, 1987, Guo et al., 2017], probabilistic sharpness [Gneiting et al., 2007, Kuleshov et al., 2018], or a proper scoring rule such as the log-loss or squared error [Gneiting and Raftery, 2007]. One may alternately fix $\tau = 1$ and aim to learn inherently calibrated probabilities, e.g., via label smoothing [Szegedy et al., 2016, Müller et al., 2019].

建模$\mathbb{P}^{\mathtt{bal}}(y|x)$

我们还有一种办法，就是直接让模型：$f : \mathcal{X} \to \mathbb{R}^L$学这个分布$\mathbb{P}^{\mathtt{bal}}(y|x)$。

$$\arg\max_{y\in\mathcal{Y}} f_y^*(x) = \arg\max_{y\in\mathcal{Y}} \mathbb{P}^{\mathtt{bal}}(y|x)$$

由前面的推导我们可以知道

$$\arg\max_{y\in\mathcal{Y}} \pi_y^\tau \mathbb{P}^{\mathtt{cal}}(y|x) = \arg\max_{y\in\mathcal{Y}} \mathbb{P}(y|x)$$

对前者使用交叉熵损失（其中一项）：

$$\mathcal{L} = -\log \frac{e^{f_y(x;\theta)+\tau\log\pi_y}}{\sum_{y'\in\mathcal{Y}} e^{f_{y'}(x;\theta)+\tau\log\pi_{y'}}} = \log(1+\sum_{y\neq y'}(\frac{\pi_{y'}}{\pi_y})^\tau e^{f_{y'}(x;\theta)-f_y(x;\theta)})$$

在这个修改过后的损失下训练的模型可以直接被用于预测均衡分布的数据而无需修改分类头。

建模$\mathbb{P}^{\text{bal}}(y|x)$

下面是我们需要优化的损失：

$$\mathcal{L} = \sum_{(x,y)\sim S} \log(1 + \sum_{y \neq y'} (\frac{\pi_{y'}}{\pi_y})^{\tau} e^{f_{y'}(x;\theta) - f_y(x;\theta)})$$

实际上，论文给出了一种更广泛的损失：

$$\mathcal{L} = \sum_{(x,y)\sim S} \alpha_y \log(1 + \sum_{y \neq y'} e^{\Delta_{yy'}} e^{f_{y'}(x;\theta) - f_y(x;\theta)})$$

可以根据需要设计（炼出）任何合理的损失。常见的有$\alpha_y = \frac{1}{\pi_y}, \Delta_{yy'} = 0$；　$\alpha_y = 1, \Delta_{yy'} = pi_y^{-\frac{1}{4}}$；
$\alpha_y = 1, \Delta_{yy'} = \log F(\pi_{y'}), F' \geq 0$；

Quiz

Quiz Theorem (互信息)

设随机变量$\mathcal{X}, \mathcal{Y}$的联合概率分布为$\mathbb{P}$，$\mathbb{P}^{bal}$是本节中提到的分布$\mathbb{P}$的均衡的版本。证明学习$\mathbb{P}^{bal}(y|x)$等价于学习$\mathcal{X}, \mathcal{Y}$ 的互信息$I(\mathcal{X}, \mathcal{Y})$

Quiz Theorem (非均衡验证集)

设随机变量$\mathcal{X}, \mathcal{Y}$的联合概率分布为$\mathbb{P}$，$\mathbb{P}^{any}$是另一个非均衡的验证分布满足$\mathbb{P}^{any}(x|y) = \mathbb{P}(x|y), \mathbb{P}^{any}(y)$是我们希望关注给予不同类关注的已知权重系数。证明为了最大化在验证集上的精度，*i.e.*,学习$\mathbb{P}^{any}(y|x)$ 是等价于学习$\frac{\mathbb{P}(y|x)\mathbb{P}^{any}(y)}{\mathbb{P}(y)}$

Proof

Quiz Theorem (互信息)

显然我们有

$$I = \frac{\mathbb{P}(x,y)}{\mathbb{P}(x)\mathbb{P}(y)} = \frac{\mathbb{P}(y|x)}{\mathbb{P}(y)}$$

Quiz Theorem (非均衡验证集)

显然我们有

$$\mathbb{P}^{any}(y|x) \propto \mathbb{P}^{any}(x|y)\mathbb{P}^{any}(y) = \mathbb{P}(x|y)\mathbb{P}^{any}(y) \propto \frac{\mathbb{P}(y|x)\mathbb{P}^{any(y)}}{\mathbb{P}(y)}$$

讨论

无内容

Table of Contents

连续标签

标签是连续的可能会导致许多问题。首先标签会有内在的bias，例如年龄标签的分布是连续的，并且相近年龄临域内样本很可能也是相似的。离散的标签没有这种性质；
正是因为这种bias，导致即使两个标签具有相同的样本数，由于其临域内的数据丰度不同，他们实际上有着不同的imbalance，这种问题导致我们需要注入额外的偏差信息来纠正这种imbalance。

问题定义

设有一个样本空间$\mathcal{X}$与一个连续的标签空间$\mathcal{Y}$。 $\mathbb{P}$是某个未知的定义在$\mathcal{X} \times \mathcal{Y}$上的分布。设有从$\mathbb{P}$中的$N$次采样：

$$S = \{(x_n, y_n)\}_{n=1}^{N} \sim \mathbb{P}^N$$

通常处理这种数据是把$\mathcal{Y}$拆成有限个不相交的区间$\mathcal{Y} = \cup_{i=1}^{B} \mathcal{Y}_i$，这样我们就将标签离散化了：$\mathcal{Y} \to \{1, \cdots, B\}$.

标签分布平滑

一个很直接的想法是，某个标签的真实imbalance属性不止和自身有关，还应该与自身周围的数据也有关（换言之，学习该标签周围的标签对识别该标签也是有利的，如果周围的数据较多，那么该标签的imbalance程度不应该太低）。作者使用了最简单的核密度估计来达成该目标：

$$q(y) = \int_{y' \in \mathcal{Y}} \kappa(y, y') p(y') dy$$

Note

$\kappa(y, y')$是核$\mathcal{K} \times \mathcal{K} \to \mathbb{R}$满足$\kappa(x, y) = \kappa(y, x), \nabla_x \kappa(x, y) = \nabla_y \kappa(y, x)$。$q(y)$就是纠正过之后的标签概率。通过核的度量引入了对bias的纠正。在获得了纠正后的概率后，常用于长尾的办法可以立刻使用。如直接加权损失$\frac{\mathcal{L}(y_i, \hat{y}_i)}{q(y_i)}$。

特征分布平滑

不只是标签，模型学到的特征也应该有类似的性质。换言之，提取该标签周围的样本的features，对提取该标签也应该是有益的（比如学习到55岁与56岁的人的特征，对学习到55岁的特征也应该是有好处的）。对于每一个区间$b \in \{1, \cdots, B\}$统计其区间内均值$\mu_b = \frac{1}{N_b} \sum_i z_i$与方差$\Sigma_b = \frac{1}{N_b - 1} \sum_i (z_i - \mu_b)(z_i - \mu_b)^T$，对这两个统计量做核密度估计：

$$\mu'_b = \int_{b' \in [B]} \kappa(b, b') \mu_{b'} d\mu$$

$$\Sigma'_b = \int_{b' \in [B]} \kappa(b, b') \Sigma_{b'} d\Sigma$$

使用这两个统计量去更新$z' = \Sigma_b'^{\frac{1}{2}} \Sigma_b^{-\frac{1}{2}} (z - \mu_b) + \mu'_b$. 对每个epoch，反向传播完成后，计算该epoch的$\{\mu'_b, \Sigma'_b\}$后moving avg去更新这两个统计量。