Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss

LDAM这篇paper因为是比较之前的了一直没看，最近在分析最近的一些模型，感觉都没有很好的解决margin不平衡这问题，回来看这篇margin相关的paper，发现里面竟然用到了Empirical Rademacher Complexity，我目前刚好也是想从这个复杂度入手控制整个model的跨分布问题。

Introduction

这篇paper主要是提出了一个考虑基于类内样本数量的margin的softmax loss，此外也提出了一个训练trick。

因为是19年，当时的工作大多是re-weighting、re-sampling，论文提到这些方法大多是设计一个training loss来从期望上接近balanced test distribution。论文提出的方法是去对尾部类做更强的regularization，在这其中需要一个regularizer去引入各类样本数这一先验信息才行，而不是l2 norm这种与类别无关的。因此，文中考虑到是margin，因为标准的泛化误差上界就是和margin大小成反比的，所以让margin变大可以看作是一种正则化。为了去思考怎么样最小化关于尾部类的泛化性能，论文考虑去获得每个类的最小margin，这样每个类和所有类的test error bound就可以得到了（举了个下图二分类的例子，意思就是说在这种二分类时，可以保持分类性能不变，也就是图中分界线的方向不变，同时$\gamma_1 + \gamma_2$不变，修改每个类各自的margin，而这个margin又是作为一个类似regularizer的存在影响了各个类的error bound，与类内样本数有关，可以被优化）。

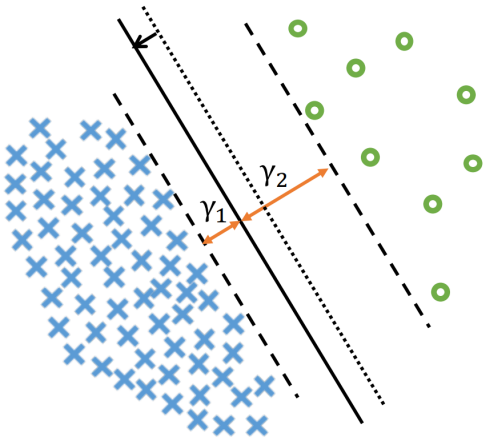

Figure 1: For binary classification with a linearly separable classifier, the margin γ_i of the i-th class is defined to be the the minimum distance of the data in the i-th class to the decision boundary. We show that the test error with the uniform label distribution is bounded by a quantity that scales in $\frac{1}{\gamma_1\sqrt{n_1}} + \frac{1}{\gamma_2\sqrt{n_2}}$. As illustrated here, fixing the direction of the decision boundary leads to a fixed $\gamma_1 + \gamma_2$, but the trade-off between γ_1, γ_2 can be optimized by shifting the decision boundary. As derived in Section 3.1, the optimal trade-off is $\gamma_i \propto n_i^{-1/4}$ where n_i is the sample size of the i-th class.

正是受上面这个情况的启发，作者思考在长尾问题里，因为各类样本数差别较大，最后被优化出来的margin也应该不相同（事实上，根据最后结论，样本数少的类，margin要大一点），作者就是用这个思路去修改了softmax loss，提出了LDAM loss。

文中提到这个方法和re-weighting、re-sampling 是正交的，也就是说论文的方法能和之前的re-weighting、re-sampling一起用不矛盾。事实上，论文是设计了一种新的两阶段的结构把两者结合在一起用，实现了比较好的性能。

总体来说，论文提出的问题是为了让尾部类的泛化性能能提升一点，在使用过程中要和其他方法一起用。

Related work

- 首先re-sampling的问题不赘述了，其它笔记里有。
- 然后就是re-weighting了，这篇paper也属于其中之一。

第一类re-weighting是和样本频率有关的，最常用的做法是样本数越高权重越低，但是这样会对头部类造成比较大的影响，所以比较代表性的一篇paper是考虑说这里的样本数做个修改，提出了一个有效样本数概念，具体我没看，直觉上是把原来固定不动的样本频率设置成可优化的，根据训练情况做优化。

第二类re-weighting是考虑样本难易程度、梯度这些的，比较代表性的是focal loss。

此外，最近又研究指出re-weighting还是要和regularization一起结合着使用，这样才能到达margin最大解，所以论文里也是用了l2 正则化，我看代码应该是同时对最后分类的weight还有feature做了l2 norm。

- 第三块是margin相关的，这其实是挺大一块内容了，论文比较多，还没接触过太多，不过文章中提到有涉及imbalance问题的margin设计，之前一直卡在这里，mark一下，有时间看看。
- 后面两块是域自适应的label shift问题还有元学习，这两块在论文中不是重点，但是作者认为本文的方法可以很好的被用在或者替换前面两类问题方法。关于域自适应，最近看到一篇发在AAAI 2022的paper：AAAI-2022-Cross-Domain Empirical Risk Minimization for... 感觉有一定启发，可以去看看，我也做了相关笔记。

Motivation & LDAM loss

- 先介绍一些符号和定义：

$$\mathcal{P}_j = \mathcal{P}(x \mid y = j)$$

$$L_{\text{bal}}[f] = \Pr_{(x,y)\sim\mathcal{P}_{\text{bal}}}[f(x)_y < \max_{\ell\neq y} f(x)_\ell]$$

$$L_j[f] = \Pr_{(x,y)\sim\mathcal{P}_j}[f(x)_y < \max_{\ell\neq y} f(x)_\ell]$$

这里的L_{bal}、L_j都是平均损失，因为取得是0-1损失，所以后面是概率。下面是margin定义：

$$\gamma(x, y) = f(x)_y - \max_{j\neq y} f(x)_j$$

$$\gamma_j = \min_{i\in S_j} \gamma(x_i, y_i)$$

$$\gamma_{\min} = \min\{\gamma_1, \ldots, \gamma_k\}$$

那么引入margin的0-1 loss也就如下所示：

$$L_{\gamma,j}[f] = \Pr_{x\sim\mathcal{P}_j}[\max_{j'\neq j} f(x)_{j'} > f(x)_j - \gamma]$$

- 接下来是核心的一些定理：

论文出发点是发现尾部类的泛化误差比较大，对于测试集和训练集数据分布相同的情况，测试集上的泛化误差有个上界$\mathrm{C}(\mathcal{F})/\sqrt{n}$，而对于本文讨论的问题，如果训练数据集也是imbalance的，那么就有误差上界：

$$\text{imbalanced test error} \lesssim \frac{1}{\gamma_{\min}}\sqrt{\frac{\mathrm{C}(\mathcal{F})}{n}}$$

但是这个上界并没有考虑进来类别信息，事实上，如果我们能按照每个类的类别数计算出一个error bound，那肯定比上面这个更能贴近我们的数据分布。那么就有了接下来这个定理（可以看成上面那个式子分成各个类别来说）：

$$L_j[f] \lesssim \frac{1}{\gamma_j}\sqrt{\frac{\mathrm{C}(\mathcal{F})}{n_j}} + \frac{\log n}{\sqrt{n_j}}$$

$$L_{\text{bal}}[f] \lesssim \frac{1}{k}\sum_{j=1}^{k}\left(\frac{1}{\gamma_j}\sqrt{\frac{\mathrm{C}(\mathcal{F})}{n_j}} + \frac{\log n}{\sqrt{n_j}}\right)$$

Proof：上面这个式子的证明如下，首先有下面两个式子：

$$\hat{\mathfrak{R}}_j(\mathcal{F}) = \frac{1}{n_j}\mathbb{E}_\sigma\left[\sup_{f\in\mathcal{F}}\sum_{i\in S_j}\sigma_i[f(x_i)_j - \max_{j'\neq j} f(x_i)_{j'}]\right]$$

$$L_{\text{bal}}[f] \leq \frac{1}{k}\left(\sum_{j=1}^{k}\hat{L}_{\gamma_j,j}[f] + \frac{4}{\gamma_j}\hat{\mathfrak{R}}_j(\mathcal{F}) + \epsilon_j(\gamma_j)\right)$$

上面这两个式子实际上就是 Rademacher complexity 最基础的定义稍微改了改，其中第一个就是第j类margin的Rademacher complexity，文中没提，没想错的话应该是因为margin属于0-1（概率相减），所以直接就符合Rademacher complexity定义了。关于第二个式子，证明用了下面这个定理（另外一篇paper里的），下面是对于每个类有一个bound，最后对于balance情况，这些bound应该等weight加权起来。

$$L_j[f] \leq \hat{L}_{\gamma_j,j} + \frac{4}{\gamma_j}\hat{\mathfrak{R}}_j(\mathcal{F}) + \sqrt{\frac{\log\log_2\left(\frac{2\max_{x\in\mathcal{X},f\in\mathcal{F}}|f(x)|}{\gamma_j}\right)}{n_j}} + \sqrt{\frac{\log\frac{2c}{\delta}}{n_j}} \tag{14}$$

对于上面第二个$L_{bal}[f]$的式子，往往会对Rademacher complexity进一步放缩，如下：

$$L_{\text{bal}}[f] \leq \frac{1}{k}\left(\sum_{j=1}^{k}\hat{L}_{\gamma_j,j}[f] + \frac{4}{\gamma_j}\sqrt{\frac{\mathrm{C}(\mathcal{F})}{n_j}} + \epsilon_j(\gamma_j)\right)$$

证明好了，抛开证明，根据上面这个定理，因为最开始目的是最小化尾部类的error，那么按照定理需要加大尾部类的margin，但是这样会损害头部类的margin，那么这就是一个要优化的问题了，如何平衡二者，着显然非常的困难。好在可以在二分类中找到一个答案。

在二分类中，总error bound可以简化成最下滑下面这个式子（也就是省略了最后那个无穷小量以及提出了C(F)这个公因子，因为我们在这里考虑的是两个类别的关系，而C(F)和分类器有关，所以不考虑）：

$$\frac{1}{\gamma_1\sqrt{n_1}}+\frac{1}{\gamma_2\sqrt{n_2}}$$

最前面那张图里有说两者之和要保持一致（也就是那个分解线bound移动一下两个 γ就可以偏移一下了）：

$$\frac{1}{\gamma_1\sqrt{n_1}}+\frac{1}{\gamma_2\sqrt{n_2}}\leq\frac{1}{(\gamma_1-\delta)\sqrt{n_1}}+\frac{1}{(\gamma_2+\delta)\sqrt{n_2}}$$

设$\gamma_1+\gamma_2=\beta$，换元以后，对上式关于γ_1求导：

$$\frac{1}{(\beta-\gamma_1)^2\sqrt{n_2}}-\frac{1}{\gamma_1^2\sqrt{n_1}}=0$$

求解可以得到：

$$\gamma_1^\star=\frac{\beta n_2^{1/4}}{n_1^{1/4}+n_2^{1/4}}\ ,\gamma_2^\star=\frac{\beta n_1^{1/4}}{n_1^{1/4}+n_2^{1/4}}$$

这里有点疑问，按照上面（文中附录）的证明，得到的应该是上面这个解，但论文用的是下面这个，虽然意思差不多，但是还是不等价的，不知道为什么？

$$\gamma_1=\frac{C}{n_1^{1/4}},\ \text{and}\ \gamma_2=\frac{C}{n_2^{1/4}}$$

正是受上面这种二分类是的最优解的形式启发，论文提出在每个类别都加一个这样的margin，论文在使用时候用的是一个soft margin loss, 而非证明里的0-1loss。

除去0-1 loss，其实有很多loss的选择，我记得西瓜书书上就列举了三四个，其中最常用的就是hinge loss了：

$$\mathcal{L}_{\text{LDAM-HG}}((x,y);f)=\max(\max_{j\neq y}\{z_j\}-z_y+\Delta_y,0)$$

$$\text{where}\ \Delta_j=\frac{C}{n_j^{1/4}}\ \text{for}\ j\in\{1,\ldots,k\}$$

更加smooth一点就是结合上交叉熵，也是很常见的一个做法：

$$\mathcal{L}_{\text{LDAM}}((x,y);f) = -\log \frac{e^{z_y-\Delta_y}}{e^{z_y-\Delta_y}+\sum_{j\neq y} e^{z_j}}$$

$$\text{where } \Delta_j = \frac{C}{n_j^{1/4}} \text{ for } j \in \{1,\dots,k\}$$

Deferred Re-balancing Optimization Schedule

论文除了LDAM loss，另一个创新点便是提出了一个训练方法。

论文发现之前的re-weighting、re-balancing这些方法，在实际应用中lr下调之前，性能都不及正常训练的模型。论文也做了消融实验：

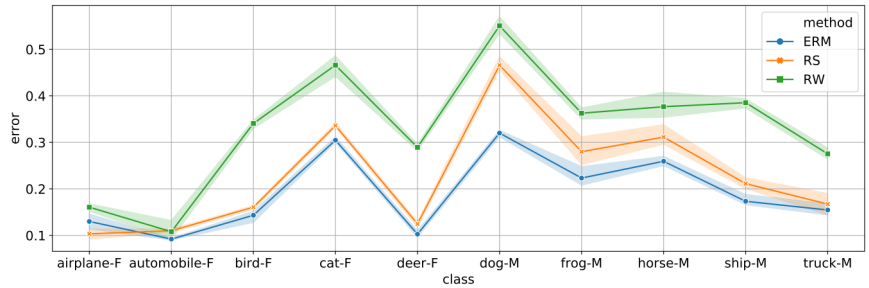

Figure 6: In the setting of training mbalanced CIFAR-10 dataset with step imbalance of $\rho = 100, \mu = 0.5$, to test the quality of the features obtained by the ERM, RW and RS before annealing the learning rate, we use a subset of the *balanced* validation dataset to train linear classifiers on top of the features, and evaluate the per-class validation error on the rest of the validation data. (Little over-fitting in training the linear classifier is observed.) The left-5 classes are frequent and denoted with -F. The features obtained from ERM setting has the strongest performance, confirming our intuition that the second stage of DRW starts from better features. In the second stage, DRW re-weights the example again, adjusting the decision boundary and locally fine-tuning the features.

所以论文就提出分两阶段训练，第一阶段lr比较大时，就用上LADM loss正常训练，以获得一个比较好的基本模型，第二阶段再用上re-weighting loss，这时候因为lr 比较小了，所以整个模型不会变化太大，模型的框架如图：

Algorithm 1 Deferred Re-balancing Optimization with LDAM Loss

Require: Dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$. A parameterized model f_θ
Initialize the model parameters θ randomly
for $t = 1$ to T_0 **do**
  $\mathcal{B} \leftarrow \text{SampleMiniBatch}(\mathcal{D}, m)$ ▷ a mini-batch of m examples
  $\mathcal{L}(f_\theta) \leftarrow \frac{1}{m}\sum_{(x,y)\in\mathcal{B}} \mathcal{L}_{\text{LDAM}}((x,y);f_\theta)$
  $f_\theta \leftarrow f_\theta - \alpha\nabla_\theta\mathcal{L}(f_\theta)$ ▷ one SGD step
  Optional: $\alpha \leftarrow \alpha/\tau$ ▷ anneal learning rate by a factor τ if necessary

for $t = T_0$ to T **do**
  $\mathcal{B} \leftarrow \text{SampleMiniBatch}(\mathcal{D}, m)$ ▷ A mini-batch of m examples
  $\mathcal{L}(f_\theta) \leftarrow \frac{1}{m}\sum_{(x,y)\in\mathcal{B}} n_y^{-1} \cdot \mathcal{L}_{\text{LDAM}}((x,y);f_\theta)$ ▷ standard re-weighting by frequency
  $f_\theta \leftarrow f_\theta - \alpha\frac{1}{\sum_{(x,y)\in\mathcal{B}} n_y^{-1}}\nabla_\theta\mathcal{L}(f_\theta)$ ▷ one SGD step with re-normalized learning rate

Conclusion

总体来说，做了很多次实验，确实涨点比较明显，论文把re-weighting放第二阶段的方法确实能带来性能提升，从第一阶段到第二阶段性能提了5-10%，应该是因为开始注意调整classifier的缘故。

但我个人从结果上看，train和test的error差别还是挺大的，并且我算了算最后各个类别的平均置信度和acc，还是会偏向头部类，感觉这个 margin loss里的 ^(1/4)还是太主观了。