Multiclass exponential loss (Problem 4)

Mohamad Lakkis

May, 2024

This PDF is to explain the *Multiclass exponential loss (Problem 4)*

Part a: The Population minimizer f^* of the exponential loss, using lagrange multipliers

Our Goal:

$$\underset{f(x)}{\text{argmin}} E_{Y|X}[e^{-\frac{1}{K}Y^T f(X)}] = \underset{f(x)}{\text{argmin}} E_{Y|X}[e^{-\frac{1}{K}(Y_1 f_1(X)+...+Y_K f_K(X))}]$$

where $Y = (Y_1, ..., Y_K)^T$, $f = (f_1, ..., f_K)^T$,and $\sum_{k=1}^K f_k(X) = 0$, taking that

$$Y_k = \begin{cases} 1 & \text{if } G = G_k, \\ -\frac{1}{K-1} & \text{otherwise.} \end{cases}$$

In other words,

$$\underset{f(x)}{\text{argmin}} E_{Y|X}[e^{-\frac{1}{K}(Y_1 f_1(X)+...+Y_K f_K(X))}]$$

Subject to $\sum_{k=1}^K f_k(X) = 0$.

Notice how the constraint that $\sum_{k=1}^K f_k = 0$, make sure that when $K = 2$, this becomes the regular 2-class exponential loss.

- Objective Function f:

$$f(f_1, \ldots, f_K) = \sum_{k=1}^K \exp\left(-\frac{f_k}{K-1}\right) \text{Prob}(C = k|X = x)$$

- Constraint Function g:

$$g(f_1, \ldots, f_K) = \sum_{k=1}^K f_k$$

Taking gradients with respect to each f_k and setting them to zero leads to:

$$\nabla f(f_1, \ldots, f_K) = \lambda \nabla g(f_1, \ldots, f_K)$$

The Lagrange function $\mathcal{L}$ is given by:

$$\mathcal{L}(f_1, \ldots, f_K, \lambda) = f(f_1, \ldots, f_K) - \lambda \cdot g(f_1, \ldots, f_K)$$

Expressing this for each component f_k:

$$\frac{\partial f}{\partial f_k} = -\frac{1}{K-1} \exp\left(-\frac{f_k}{K-1}\right) \text{Prob}(C = k|X = x)$$

And so expressing the Lagrange function partial derivatives with respect to each f_k:

$$\frac{\partial \mathcal{L}}{\partial f_k} = \frac{\partial f}{\partial f_k} - \lambda \frac{\partial g}{\partial f_k}$$

Notice that $\cdot \frac{\partial g}{\partial f_k} = 1, \forall k$, and so the latter becomes:

$$\frac{\partial \mathcal{L}}{\partial f_k} = \frac{\partial f}{\partial f_k} - \lambda$$

$$\frac{\partial \mathcal{L}}{\partial f_k} = -\frac{1}{K-1} \exp\left(-\frac{f_k}{K-1}\right) \text{Prob}(C = k|X = x) - \lambda$$

And we set $\forall k, \frac{\partial \mathcal{L}}{\partial f_k} = 0$ to obtain K equations. But notice that we have $K+1$ unknowns, and so we need to add the constraint $\sum_{k=1}^{K} f_k = 0$ to solve the system. (or in other words, we need to take the partial derivative of the Lagrange function with respect to λ and set it to zero).

$$\frac{\partial \mathcal{L}}{\partial \lambda} = \sum_{k=1}^{K} f_k = 0$$

Solving for the minimizer f^*

Step 1: Isolating λ

Isolating λ gives:

$$\lambda = \frac{1}{K-1} \exp\left(-\frac{f_k(x)}{K-1}\right) \text{Prob}(C = k|x)$$

Step 2: Setting All λ Equal

Equating λ expressions for classes i and j:

$$\exp\left(-\frac{f_i(x)}{K-1}\right) \text{Prob}(C = i|x) = \exp\left(-\frac{f_j(x)}{K-1}\right) \text{Prob}(c = j|x)$$

Step 3: Taking the log

$$-\frac{f_i(x)}{K-1} + \log(\text{Prob}(C=k|x)) = -\frac{f_j(x)}{K-1} + \log(\text{Prob}(c=j|x))$$

Step 4: Solving for $f_i(x)$

$$f_i(x) = f_j(x) + (K-1)\left(\log(\text{Prob}(C=j|x)) - \log(\text{Prob}(C=i|x))\right)$$

Step 5: Final Expression for $f_k^*(x)$

We have that

$$\sum_{k=1}^{K} f_k(x) = 0$$

The population minimizer $f_k^*(x)$ is with the use of an assumption over the form of f_k, this process is called ansatz:

$$f_k^*(x) = (K-1)\left(\log(\text{Prob}(C=k|x)) - \frac{1}{K}\sum_{k'=1}^{K}\log(\text{Prob}(C=k'|x))\right)$$

From Minimization of Loss to Maximization of Probability

By maximizing $f_k^*(x)$, you increase the likelihood of correctly classifying an instance into its true class k, because it directly influences the probability $\text{Prob}(c=k|x)$ to be higher for the correct class. This maximization is effectively making $y^T f$ larger, which is crucial because $y_k = 1$ for the correct class and $y_k = -\frac{1}{K-1}$ otherwise. Thus, as $f_k^*(x)$ increases for the correct class k, the product $y^T f$ becomes more positive, maximizing its value, which minimizes the loss due to the negative sign in the exponential function's exponent. So important to notice here that we are now in a problem of maxizing this f_k^*, which in its turn will minimize the loss function as explained above. And so the problem becomes:

$$\underset{k}{\text{argmax}} f_k^*(x)$$

Which is the same as discussed before,

$$\underset{k}{\text{argmax}} Prob(C=k|X=x)$$

And notice that when $f_k^*(x)$ are known (or estimated), we can easily calculate the probability of each class given the input x. Given by,

$$Prob(C=k|X=x) = \frac{e^{\frac{1}{K-1}\hat{f}_k^*(x)}}{\sum_{k'=1}^{K} e^{\frac{1}{K-1}\hat{f}_{k'}^*(x)}}$$

Algorithm 1 ADAboost with exponential loss for multiclass classification

```
1. Initialize the observation weights $w_i = \frac{1}{N}$, $i = 1, 2, \ldots, N$.
2. For $m = 1$ to $M$:
    (a) Fit a classifier(not necessarly a tree, what we did previously is for trees, but can easily be generalized as we will do here)$G_m(x)$ to the training data using weights $w_i$.
    (b) Compute
            $\text{err}_m = \frac{\sum_{i=1}^{N} w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^{N} w_i}$.
    (c) Compute $\alpha_m = \log\left(\frac{1-\text{err}_m}{\text{err}_m}\right) + log(K-1)$.
    (d) Set $w_i \leftarrow w_i \cdot \exp(\alpha_m \cdot I(y_i \neq G_m(x_i)))$, $i = 1, 2, \ldots, N$.
3. Output $G(x) = \underset{k}{argmax} \sum_{m=1}^{M} \alpha_m I(G_m(x) = k)$ (as we have discussed in parta).
```

Part b: Algorithm Used similar to ADABoost

Analysis of the Algorithm

- The algorithm is similar to ADABoost, but with the exponential loss function for multiclass classification.
- we can see that this algorithm puts more weight on the misclassified points
- So we can expect that the test error will keep on decreasing even after the point where the test error starts to increase in the case of ADABoost.(problem 3)
- Notice that when K = 2, this algorithm becomes the regular ADABoost algorithm. Since log(1) = 0, and so the term $log(K-1)$ will be zero.
- In a multi-class setting with K classes, a misclassification can occur in $K-1$ different ways (not just one, as in the binary case). The AdaBoost formula needs to reflect the increased difficulty and the higher stakes involved in choosing correctly among more options.
- Please note that the addition of the term $log(k-1)$ is very important as I have discussed but unfortunately, I couldn't understand mathematically where it came from, I found it using the internet, and I am sorry for that but honestly I wasn't able to know how it is derived. I understood why it is there but not how it is derived mathematically.

- So we can now create our own algorithm for multiclass classification using the exponential loss function. Very similar to ADABoost, but with the addition of the term $log(K-1)$, in step (c).
- Notice that in our version in the python code we decided to use trees. But as we have discussed, we can use any classifier, and so we can easily generalize the algorithm to use any classifier.
- as for the predict method, it is a bit different from the one in problem 3, since here for each class k, we need to calculate $\sum_{m=1}^{M} \alpha_m I(G_m(x) = k)$, and then choose the class with the maximum value.