AIML - CS 337

Lecture 4: Introduction to Loss Functions

18/08/22

Lecturer: Abir De *Scribe:*

In this lecture we develop methods that evaluate how well machine learning models learn our data set. This is accomplished by means of *loss* functions. We see what entails in the design of such loss functions.

1 Motivation for Loss Functions

We study about loss function in a simple setting of classification which is defined as follows:

1.1 Review of Classification Task

The general classification task can be described as follows :

Classification Task : Given a dataset $\{(x_i, y_i)\}$ where $x_i \in \mathbb{R}^d$ and $y \in \mathcal{Y}$, where d is dimension of the input features and $\mathcal{Y}$ is the target variables, our goal is to devise an algorithm that selects a hypothetsis $h \in H : \mathbb{R}^d \to \mathcal{Y}$, So that given some unseen test case x_j and the corresponding label y_j, $h(x_j) = y_j$. For example, in `MNIST` image classification task, each $x_i \in \mathbb{R}^{28\times 28}$ and $\mathcal{Y} = \{0, 1, \cdots 9\}$ are the class labels.

However, in order to devise a good algorithm that selects the best hypothesis $h^* \in H$, the developer needs to know a more concrete metric which is used by the user to evaluate the algorithm on the test set. This metric which mathematically quantifies how well the algorithm models the data in any dataset is called a **Loss Function**.

1.2 Loss Minimization Task

Now that we are aware of the metric which is used by the user to evaluate the algorithm H we try to model the classification task as a loss minimization task instead. The general classification task can thus be modelled by the following loss minimization task :

$$h^* = \arg\min_{h\in H} \sum_{j=1}^{M} \mathbb{1}(h(x_j) \neq y_j)$$

Here x_j is an image in the test dataset, M is the number of images in the test dataset and $\mathbb{1}(\cdot)$ is the indicator function which is equal to 0 if its argument is false and equals 1 if the argument holds

true. If we assume that (a) we could enumerate all the possible functions $H : \mathbb{R}^d \to Y$ and (b) the function that minimizes the value over training set also minimizes it over the test set we could obtain h^* as follows :

$$h^* = \arg\min_{h \in H} \sum_{i=1}^{D} \mathbb{1}(h(x_i) \neq y_i)$$

where x_i are the images in the training dataset, y_i are the corresponding labels and D is the number of images in the training dataset. However, both our assumptions are clearly unreasonable. Most importantly it is not possible to enumerate all possible functions H particularly when H is an infinite set which is often the case. Therefore, we look for relaxations in the loss function in order to make the loss minimization task solvable. Some suitable relaxations for the loss function have been discussed in the next section.

2 Relaxation of Loss Function

In this section, we will try to look at the art of designing loss functions. We will look at, how we can arrive at a mathematically appealing loss function that models the classification problem in steps.
In all the subsections below we will consider the classification problem where $\forall x_i \in X$, there exists a label $y_i \in \{-1, 1\}$, over the training set.

2.1 Constant Hypothesis

While developing the loss function, one student suggested to use constant hypothesis. In other words, $H(x_i) = c$, where c is a uniformly generated random number in the interval $[-1, 1]$.
The loss function optimization problem can then be written as,

$$c^* = \arg\min_{c} \sum_{i=1}^{M} \mathbb{1}(c \neq y_i)$$

We do know that the point probability of a continuous distribution is 0 whence $P(c = 1)$ and $P(c = -1)$ are both zero. Hence, the loss function value is always M in this case. We cannot do better if we stick to this model with uniform distribution.
What if we choose the constant c among the values $\{-1, 1\}$. Let us denote n_+ to be the number of points in the training data set having labels as +1, and similarly denote n_- to be the number of points in the training data set having labels as -1. The optimization problem is same as above, but constrained to the fact that $c \in \{-1, 1\}$. It can be seen easily that if we take, $c = \max(n_+, n_-)$, the loss say L is $\min(n_+, n_-)$. This is the minimum that we can get. This method is Majority Mode, since we took the hypothesis to be the mode in the training set data.

2.2 Linear Hypothesis with Indicator Cost

Lets see if we can do better by increasing the complexity of our hypothesis class H. We suppose, $h(x_i) = w^T x_i + b$, where w is the vector of parameters of the same dimensions as x_i and b is the bias parameter. Hence, we have the following task in hand,

$$\{w^*, b^*\} = \arg\min_{w,b} \sum_{i=1}^{M} \mathbb{I}(w^T x_i + b \neq y_i)$$

For brevity, let L_1 denote the minimum loss achieved by considering a constant hypothesis and L_2 denote the minimum loss achieved by considering a linear hypothesis, then it is guaranteed that $L_2 \leq L_1$. The reason is obvious as we are trying to search a larger space to fit the data. Hence, our Linear model is reducible to constant hypothesis model by taking $(w, b) = (\mathbf{0}, c)$.
We can establish a generalized statement here. As model complexity increases, performance on the data used to build the model (training data) improves. However, performance on an independent set (validation data) may improve up to a point, then starts to get worse. This is called **overfitting**. Nevertheless, we have definitely done better as compared to constant hypothesis.

2.3 Linear Hypothesis with Absolute Difference Cost

Another Loss function was suggested which takes the cost as the absolute value of the difference between hypothesis and the label value (Assuming that the labels are mapped to some subset of integers). In this case the optimization problem becomes,

$$\{w^*, b^*\} = \arg\min_{w,b} \sum_{i=1}^{M} |w^T x_i + b - y_i|$$

But even this is not a good choice. $w^T x_i + b$ takes values in $\mathbb{R}$ but $y_i \in \{-1, 1\}$. A good loss function should act on predictions that are on the same scale as that of the ground truth targets which we allude to next.

2.4 Linear Hypothesis with Sign and Indicator Cost

Instead of looking at the value of $w^T x_i + b$, what if we look at it's sign. Hence, if $w^T x_i + b > 0$, then the estimated label should be 1 and vice-versa (boundary condition can be included within any one of them). This gives us the following optimization problem,

$$\{w^*, b^*\} = \arg\min_{w,b} \sum_{i=1}^{M} \mathbb{I}(\text{sgn}(w^T x_i + b) \neq y_i)$$

where $\text{sgn}(.)$ denotes the signum function.

2.5 Linear Hypothesis with Sigmoid Mapping

While the above loss function is good, it is not conducive to design efficient search algorithms that find h^* and is difficult to optimize. Nonetheless, the above loss function can still be used to test the performance of our trained models.

To make optimization convenient, we map $w^T x_i + b$ that takes values in $\mathbb{R}$ to the interval $[-1, 1]$. To do this, we can use the sigmoid activation function,

$$f(x_i) = \frac{1}{1 + e^{-(w^T x_i + b)}}$$

There's one downside to this, the sigmoid function doesn't map to $[0, 1]$, instead it does to $(0, 1)$. Hence, using the below optimization problem wouldn't make sense,

$$\{w^*, b^*\} = \underset{w,b}{arg\ min} \sum_{i=1}^{M} \mathbb{I}(f(x_i) \neq \frac{y_i + 1}{2})$$

One may be tempted to use the absolute difference squared cost, $|f(x_i) - \frac{y_i+1}{2}|^2$, which makes sense. The issue which we may incur due to this is a non-convex loss function of parameters. It may be difficult to converge to global minima in such cases.

There's a probabilistic approach to loss functions as well, which will be discussed in the upcoming lectures, but we will just state the result here,

$$\{w^*, b^*\} = \underset{w,b}{\arg\min} \sum_{i=1}^{M} \left[-\left(\frac{y_i + 1}{2}\right) \log(f(x_i)) - \left(1 - \frac{y_i + 1}{2}\right) \log(1 - f(x_i)) \right]$$

We can also define the loss in the following manner. We will incur a loss if $f(x_i) > 0.5$ and $y_i = -1$ OR $f(x_i) \leq 0.5$ and $y_i = 1$. We can hence write the following optimization problem,

$$\{w^*, b^*\} = \underset{w,b}{\arg\min} \sum_{i=1}^{M} \max\left(0, \left(\frac{1}{2} - f(x_i)\right) y_i\right)$$

This kind of loss is inspired from the ReLU function which is defined as

$$\text{ReLU}(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{otherwise} \end{cases} \tag{1}$$