CS 419M Introduction to Machine Learning **Spring 2021-22**

Lecture 2: Loss Functions in Machine Learning

Lecturer: Abir De *Scribe: Group 1*

2.1 Loss function for Image Classification

2.1.1 Problem Setup

- An Image I is represented by a vector x, x ϵ R^d. 2d array is collapsed into a 1d vector by well developed methods. For now we are considering only black & white images.
- Label of an Image y represents the object present in the image. eg: a car, a tree or a river. If there are two classes, let's say dog and car, we can have y = 0 for dog and y = 1 for car.
- We consider a Dataset D to be set of images that belong to two classes either C1 or C2.

$$\mathbb{D} = \{(x_i, y_i) | i \in \mathbb{I}\}$$

where x_i is the 1d vector representation of the image and $y_i \ \epsilon \ \{0, 1\}$.

2.1.2 What is Classification

- Goal of classification is to find out the labels of test/unseen images.
- **Unseen Images:** The instance/image that was not revealed or *held back* during the development of the ML model/algorithm. Used during validation and testing.
- We need to design a function h($\cdot$) that will be able to give accurate class label i.e.,

$$y = h(x) \qquad \forall\ x \in \text{Test set}$$

using the information from training set.
- **Training set:** The set of examples (tuples (x_i,y_i) (in this case, image representations along with labels) provided to the machine learning model/ algorithm at the development stage.

2.1.3 How to find function h(x)

The idea is find the best possible $h(x) \in \mathbb{H}$, where $\mathbb{H}$ is the space of candidate functions such that the error is minimized. From first principles, one could think of

$$\min_{h(x)\in\mathbb{H}} \sum_{(x_i,y_i)\in\mathbb{D}} |h(x_i) - y_i| \qquad \text{where } y_i \in \{0,1\}\text{and } h(x_i) \in \mathbb{R}$$

but the trouble with this formulation is that $y_i \in \{0,1\}$.

Next, we try out a **penalty system** where whenever $h(x_i)$ differs from y_i, a penalty is incurred. We can summarize this as:

- if $y_i = 0$ and $h(x_i) = 1$, penalty = 1
- if $y_i = 1$ and $h(x_i) = 0$, penalty = 1
- if $y_i = 0$ and $h(x_i) = 0$, penalty = 0
- if $y_i = 1$ and $h(x_i) = 1$, penalty = 0

Keeping in mind that $h(x_i) \in \mathbb{R}$, we apply this penalty scheme as

$$\min_{h(x)\in\mathbb{H}} \sum_{(x,y)\in\mathbb{D}} \mathbb{I}(h(x) \neq y)$$

where I is the indicator function with values

$$\mathbb{I}(X) = \begin{cases} 0 & X = false \\ 1 & X = true \end{cases}$$

The above implementation can be thought of as a *hard* penalty, since we are penalising whenever $h(x) \neq y$. Note that we have dropped "i" in the expression for convenience.

The space of candidate functions is intractably large, so $h(x)$ is not easy to find. To make life easier, we restrict the output of h to $[0,1]$ by applying a known function $f(\cdot)$ on it, such that $f(h(x)) \in [0,1]$. Then, we have

$$\min_{h(x)\in\mathbb{H}} \sum_{(x,y)\in\mathbb{D}} \mathbb{I}(f(h(x)) \neq y)$$

In both cases we are searching over the entire space of functions, but in the second case our work is reduced as $f(\cdot)$ ensures that the indicator function has to only deal with a value 0 or 1 as an input argument.

One possible function that has the above property is $sign(x)$ i.e. the signum function. We construct $f(\cdot)$ as an affine transformation applied on the signum function, to avoid negative outputs. Note that our test-time function is no longer $h(x)$.

$$sign(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases}$$

$$\frac{1 + sign(h(x))}{2} = \begin{cases} 1 & h(x) > 0 \\ 0.5 & h(x) = 0 \\ 0 & h(x) < 0 \end{cases}$$

Our new optimization problem becomes

$$\min_{h(x)\in\mathbb{H}} \sum_{(x,y)\in\mathbb{D}} \mathbb{I}\left(\frac{1+sign(h(x))}{2} \neq y\right)$$

These changes help to some extent but we're still not there. The space of candidate functions $\mathbb{H}$ continues to be intractable and working with $\mathbb{I}$ is cumbersome, so we settle for a linear model of h.

2.1.4 Linear Model for h(x)

Take h to be

$$h(x) = w^T x \qquad \text{for some column vector w}$$

The corresponding optimization problem is

$$\min_{w} \sum_{(x,y)\in\mathbb{D}} \mathbb{I}\left(\frac{1+sign(w^T x)}{2} \neq y\right)$$

To move into a continuous space, we get rid of the indicator function. One possibility is

$$\min_{w} \sum_{(x,y)\in\mathbb{D}} \left|\frac{1+sign(w^T x)}{2} - y\right|^2$$

The issue with this formulation is that it is not differentiable due to $sign(x)$.

Is the below modification then a good idea?

$$\min_{w} \sum_{(x,y)\in\mathbb{D}} \left|\frac{1+w^T x}{2} - y\right|^2$$

The answer is no, because $w^T x$ becoming too large implies that the loss will become large as well, which is undesirable.

We ditch this setup in favour of a differentiable one. One function which can satisfy our requirements is the sigmoid function, defined as below .

$$sigmoid(x) = \sigma(x) = \frac{1}{1+e^{-x}} \in [0,1] \qquad where\, x \in \mathbb{R}$$

Our optimization problem changes to

$$\min_{w} \sum_{(x,y)\in\mathbb{D}} \left(\sigma(w^T x) - y\right)^2$$

We run into the issue that the above optimization problem is not a *convex* one, since the loss function is not convex. Once again, note that our function is no longer $h(x)$.

2.1.5 Convex Functions

The main appeal of convex loss functions is that gradient descent is guaranteed to converge to a global minimum if they are used. So we would like to find a surrogate function for $w^T x$ with the following properties:

1. If $w^T x$ is high, y is 1, then loss is 0.
2. If $w^T x$ is low, y is 0 or -1, then loss is 0.
3. The loss has to be convex with respect to w.

One way to check whether a function is convex or not is to find the double derivative with respect to w and check if it is always positive. In terms of functions with vector arguments, the Hessian matrix with respect to w must be **positive semi-definite** for convexity to hold. One definition of positive semi-definiteness is that the eigenvalues of the matrix must be non-negative.

$$H = \left[\frac{\partial^2 a}{\partial w^2}\right] \qquad \lambda(H) \geq 0$$

The input value of **w** at which the function gives the minimum need not be unique, but the minimum must be.

2.2 Group Details and Individual Contribution

Name	Roll Number	Sections
Garaga V V S Krishna Vamsi	180070020	2.1.1, 2.1.2
Vaibhav Kumar	20d070087	2.1.3, 2.1.4
Abhinav Singh	19D180002	2.1.5
Sristy Kushwaha	180110088	2.1.3
Kavyan Lavti	200100084	2.1.4