Homework 1

Pablo Roldan

January 25, 2020

1 Questions

Question 1

Machine Learning is basically mathematical optimization. We want to learn a function $h : X \to Y$. To learn it, we have a training set of data $(x_i, y_i = h(x_i))$.

One type of ML algorithms is called "Empirical Risk Minimization", but should probably be called Empirical Error Minimization. Here, the cost function $L(h)$ ("risk") associated to h is the empirical error commited by h, i.e. the difference between the values predicted by the algorithm and the true values in the training set:

$$L(h) := \frac{1}{n} \sum |h(x_i) - y_i|.$$

"Learning" is thus equivalent to minimizing the above cost function $L(h)$ w.r.t. all functions h in a certain class $\mathcal{H}$ (e.g., linear, polynomial...).

The norm in $L(h)$ can be substituted for various "loss functions", such as

- absolute (L_1) loss: $l(\hat{y}, y) = |\hat{y} - y|$, and then $L(h)$=Mean Absolute Error.
- squared (L_2) loss: $l(\hat{y}, y) = (\hat{y} - y)^2$, and then $L(h)$=MSE.
- L_{0-1} loss: $I(\hat{y} \neq y)$ where I =indicator function, and then $L(h)$=number of hits (discrete topology).

The absolute loss has the disadvantage the it is not differentiable at 0, whereas squared loss has the disadvantage that it is very sensitive to outliers.

Question 2

Give and explain the equation for Stochastic Gradient Descent.

Gradient Descent is used for minimizing a differentiable convex function $f(w)$. Let $f : \mathbf{R}^d \to \mathbf{R}$, and denote the gradient by $\nabla f(w)$. GD is an iterative algorithm. Start with an initial value $w^1 = 0$. At each iteration, take a step in the direction of the (negative) gradient:

$$w^{i+1} = w^i - \eta \nabla f(w^i),$$

where η is the step size (a parameter). After N iterations, output w^N. (Another option is to output the average $\bar{w}$ of w over all iterations.)

It can be shown that, if f is convex and differentiable, then GD converges to the minimum w^*. The number of iterations needed to achieve a certain accuracy ϵ is $O(1/\epsilon^2)$.

In the context of Empirical Risk Minimization, learning amounts to minimizing the cost function $L(h)$ over the hypothesis class $\mathcal{H}$. We use GD to minimize $L(h)$ directly.

For a concrete example, consider the problem of linear regression with cost function MSE. Take $L(h) = \text{MSE} = \frac{1}{n}\sum_i (h(x^i) - y^i)^2$. For linear regression, $h(x) = \theta^T x$, so that the gradient is

$$\nabla_\theta \text{MSE} = \frac{2}{n} X^T (X\theta - y).$$

Here, $X = (x^1, \ldots, x^n)$ are all instances in the training set, and $y = (y^1, \ldots, y^n)$ are the corresponding labels.

In *Stochastic Gradient Descent* we do not require the update direction to be based on $\nabla f(w)$. Instead, we allow the update direction to be a random vector v_i such that its *expected value* equals the gradient direction:

$$\mathbb{E}\,(v_i | w^i) = \nabla f(w^i).$$

Like SD, Stochastic GD can be shown to be convergent, with similar convergence rate.

It turns out that v^i is easy to find for risk functions. Let D be the probability distribution of instances x, so that the risk function is

$$L(w) = \mathbb{E}_D\, l(w, x),$$

where l is the loss function and w is the hypothesis. The random vector v_i can be chosen in the following way:

1. Sample x from the probability distribution D.
2. Take $v_i = \nabla_w l(w^i, x)$.

Then we have (by linearity of the gradient)

$$\mathbb{E}\,(v_i | w^i) = \mathbb{E}\,\nabla l(w^i, x) = \nabla\, \mathbb{E}\, l(w^i, x) = \nabla L(w^i),$$

so v_i will point on average in the direction of gradient.

In the context of ERM, sampling from the distribution D amounts to choosing a vector x^i at random from the training set, with corresponding label y^i. In the case of linear regression with MSE cost function, the random vector v_i then becomes the gradient with respect to θ of the square loss function

$$l(\theta, x^i) = (\theta^T x^i - y^i)^2.$$

The gradient of this function is

$$v_i = 2{x^i}^T (x^i \theta - y^i).$$