CS 217: Artificial Intelligence and Machine Learning **Jan-Apr 2024**

Lecture 3: Regression

Lecturer: Swaprava Nath *Scribe(s): Tejas Sharma*

3.0.1 Example: Air Quality Index

Consider this equation, wherein AQI is expressed as a function of various pollutants like $\mathbf{CO_2}$, $\mathbf{SO_2}$:

$$\mathbf{AQI} \;=\; max\left(f_i(x_i)\right) \quad \forall i \in [1, n]$$

Here, x_i represents the concentration of pollutant i. Based on a few limited observations, how do we obtain the correct AQI? How do we fit the data to obtain the functions correctly so that the AQI is consistent with the next observation?

3.1 Linear Regression: An Overview

Simple, but powerful and highly interpretable tool. It works on transformations of raw data as well.

If we have some concentration of $\mathbf{CO_2}$, we have a way of estimating our AQI and have a way to interpret it.

3.1.1 How to best fit the given data?

We measure the goodness of the fit using an error function, $E(f, D)$. Note that cost function, energy function and lost functioin are just variants of this error function, but in a different context.

What is this D? Well, a collection of original data observed by an efficient reference device. Note that the input data's independent variable x can be a vector or scalar. For now, we assume they are scalars. But the dependent variable or output function, in this case AQI, y is always a scalar.

Alright, how do we define error function? Which is a good error function?

3.2 Possible Error Functions

$\sum_{\mathbf{i=1}}^{\mathbf{n}} \left(\mathbf{f(x_i)} - \mathbf{y_i}\right)$

IS this a good error function? Nope. For one, error is often zero, but what if the standard deviation is not zero? This accounts only for the mean. Maybe a different plot could put both mean and standard deviation to zero. That is a better plot. But the error function would not indiate so.

In short, the sign makes it a bad error function. We should consider the absolute deviation of each point from the function.

$\sum_{\mathbf{i=1}}^{\mathbf{n}} \|\mathbf{f(x_i)} - \mathbf{y_i}\|$

This is much better since it is unsigned and always nonnegative for any point. There fore it to someextent, reflects both the mean deviation and deviation of data overall from mean.

$\sum_{\mathbf{i=1}}^{\mathbf{n}} \left(\mathbf{f(x_i)} - \mathbf{y_i}\right)^{\mathbf{2}}$

This is a very good candidate. In fact, this is precisely the sum of squared deviation of mean and standard deviation. This can be algebraically manipulated relatively easily and hence is even better than the previous. Further, it is continuous and differentiable

$\sum_{\mathbf{i=1}}^{\mathbf{n}} \left(\mathbf{f(x_i)} - \mathbf{y_i}\right)^{\mathbf{3}}$

Not a good error function. Again, signed functions do not serve our purpose.

3.2.1 Okay, so what is the squared error function?

This is a very important function and is precisely the third function discussed so far aka $\sum_{\mathbf{i=1}}^{\mathbf{n}} \left(\mathbf{f(x_i)} - \mathbf{y_i}\right)^{\mathbf{2}}$.

This is continuous and differentiable and can be analyzed and manipulated algebraically. Moreover, it can be visualized in euclidian space. It is the euclidean distance between two n dimentional points, one whose coordinates match to the estimate function f at all x_i-s. The second is the y-coordinates of all points in the given data i.e. y_i-s.

We call the set $\mathbf{D} = (x_i, y_i) \quad \forall i \in [1, n]$ as the **training set** and each point (x_i, y_i) as the i^{th} training data.

Our Approach with vector x_is in data points

For instance, let $\vec{x_i} = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{id} \end{bmatrix}$, $\vec{x_i} \in R^d$, $y_i \in R$. The matrix $\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nd} \end{bmatrix} = \begin{bmatrix} \vec{x_1}^T \\ \vec{x_2}^T \\ \vdots \\ \vec{x_n}^T \end{bmatrix}$

3.3 General Regression

Find a function $f*$ such that $f*(x)$ is the best predictor of $\vec{y}$ w.r.t the given dataset $\mathbf{D}$.

Note that if we consider **all** possible functions to find the optimum estimate, it becomes very difficult to approach let alone solve.

Hence, we look at a limited set of functions, more specifically **parameterized** functions.

For instance, consider $f(x, (\alpha, \lambda)) = \alpha e^{-\lambda^T x}$.

Better still, consider the polynomial function $f(x, w) = w_0 + w_1 x + w_x x^2 + \ldots w_k x^k$.

Our focus is mainly on one specific class of this parameterized regression, referred to as **linear regression**.

3.4 Linear Regression Defined

We refer to the set of regression optimization problems wherein the function is of the form $f(\vec{x}) = \vec{w}^T\mathbf{x}$.

Here, $\vec{w}$ is a vector $\begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w^k \end{bmatrix}$. Note that $\mathbf{X}$ refers to $\begin{bmatrix} \vec{x_1} \\ \vec{x_2} \\ \vdots \\ \vec{x_n} \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1d} \\ 1 & x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nd} \end{bmatrix}$.

Go back to quadratic error function, rather, least square optimization. This is very common.

Consider the function $\sum_{i=1}^{n} \left(\sum_{j=1}^{d} (w_j x_{ij} - y_i) \right)^2$.

Let us get one new point. Let our estimate based on our function f be $\hat{y_i} = \sum_{j=1}^{d} w_j * x_{ij}$. For any i.

Consider a linear case. Here, $E = \sum_{i=1}^{n} (y_i - w_0 - w_1 x_i)^2$. We find (w_0, w_1) so as to minimize this.
To do so, we equate the partial deriavatives to zero.

$$\frac{\partial E}{\partial w_0} = 0 \implies \left(\sum_{i=1}^{n} (y_i - w_0 - w_1 x_i) \right) = 0$$

$$\frac{\partial E}{\partial w_1} = 0 \implies \left(\sum_{i=1}^{n} x_i (y_i - w_0 - w_1 x_i) \right) = 0$$

We have two equations in two unknowns. This can easily be solved.

A tricker problem is to minimize $\sum_{i=1}^{n} \left(y_i \; w^T x_i \right)$ over d dimentions.

We represent the vector $\vec{z} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} = \begin{bmatrix} y_1 - x_1^T \vec{w} \\ y_2 - x_2^T \vec{w} \\ \vdots \\ y_n - x_n^T \vec{w} \end{bmatrix} = \vec{y} - \mathbf{X} \cdot \vec{w}$.

This means we minimize $E = \|z\|^2 \|\vec{y} - \mathbf{x} \cdot \vec{w}\|^2$ by equating ∇E to zero.

$$\begin{aligned} \nabla E &= \nabla \left((\vec{y} - \mathbf{X}\vec{w})^T \cdot (\vec{y} - \mathbf{X}\vec{w}) \right) = \vec{0} \\ & \nabla \left(\vec{w}^T \mathbf{X^T X} \vec{w} - 2(\mathbf{X}\vec{w}) \cdot \vec{y} \right) = \vec{0} \\ & \text{On solving this, we get } \vec{w} = (\mathbf{X^T X})^{-1} \mathbf{X^T} \vec{y} \end{aligned}$$