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

Lecture 5: Introduction to Regression

Lecturer: Abir De *Scribes: Group 1 and Group 2*

5.1 Introduction to Regression

Let us consider, we are given a set of data :

$$\{(x, y)\}$$

Until this lecture we considered $y \in \{-1, +1\}$ but for this lecture now $y \in \mathbb{R}$. Here, our task is to find mapping from x to y.

$$x \mapsto y$$

5.1.1 Applications of Regression

1. Prediction of house price
2. Time series prediction(like prediction of stocks and loans,etc.)
3. Sentiment Detection

Like we take 1st example, in which you have location of house, nearer shops/houses and their prices etc. features are encoded and used for prediction of house price.

5.1.2 Formulation of the Problem

Our task in this is that you are given a set of data i.e. $(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots$ and you have to find value of y when x is given for an unseen sample. Here, unseen means y is not known and x is not present during training.

In this our goal is to come up with some function $h(x)$ so that $h(x) = y$. Now, how can we make this function? We can search the infinite function space $h(x) \in H$, but as we have seen in previous lectures, this is too large a space to search in, so we simplify our problem to a linear problem, and take that: $h(x) = w^T x$ so that

$y_i = w^T x_i$: $y_1 = w^T x_1, y_2 = w^T x_2, \ldots, y_n = w^T x_n$ means we just have to solve the following equation $[y_1 y_2 \ldots y_n] = w^T [x_1 x_2 \ldots x_n]; Y = w^T X$; Here $X = [x_1 x_2 \ldots x_n]$ where $x_1, \ldots, x_n$ are column vector of length d and X is of size dxn. We just have to solve the above equation for w.

5.1.3 What happens when $y \notin \mathcal{R}(X)$

Let us denote the row space of X by $\mathcal{R}(X)$. We know that the equation $y = W^T X$ (or equivalently $y^T = X^T W$) can be solved if $y \in \mathcal{C}(X)$ (or equivalently $y^T \in \mathcal{C}(X^T)$. However, what if $y \notin \mathcal{R}(X)$. Firstly, let us try to figure out how likely is this situation. Let us assume that $X \in \mathcal{R}^{d\times n}$, $y \in \mathcal{R}^{1\times n}$ and $W \in \mathcal{R}^{d\times 1}$
Here, n is the number of data-points and d is the dimension of the feature vector for each data-point. Usually, the number of data-points in the dataset are much larger than the feature vector of every single data-point, i.e. $d << n$.

$$\implies rank(X) = rank(X^T) \leq d \tag{5.1}$$

Now, since y is a n dimensional vector and since X cannot span entire $\mathcal{R}^n$, we can have $y \in \{\mathcal{R}^n \setminus \mathcal{R}(\mathcal{X})\}$ for which no solution (W) exists.

Although $\mathcal{R}$ is infinitesimally smaller than the whole space, implying on a pure probability level it is unlikely that y would be from this space, this is not enough to justify our formulation for regression, as if y is a perfectly linear variable $(y \in \mathcal{R}(x))$, we should be able to find a W. But we have not accounted for any measurement noise in our model. Given dataset $\{(x_i, y_i)\}$ and a linear model $y = W^T X$, we may not get a feasible solution because even if the model is accurate, it is possible that $y \notin \mathcal{R}(X)$ because y can be contaminated with noise. Hence, we should instead consider the following model:

$$y = W^T x + \epsilon \tag{5.2}$$

where ϵ represents noise. Now, to estimate W from the data using this model, we can assume some distribution for ϵ and then find the Maximum Likelihood estimate for W.

5.2 Mathematically formulating Linear Regression

Case 1: Let us assume that ϵ is a zero mean Gaussian random variable, i.e.

$$\epsilon \sim \mathcal{N}(0, \sigma^2) \tag{5.3}$$

Then we can find the MLE (Maximum Likelihood Estimate) $\hat{W}$ for W by solving the following optimisation problem

$$\hat{W} = \max_W \mathcal{P}(x_1, y_1, x_2, y_2, ..., x_n, y_n)$$

where, (x_i, y_i) are the elements of the dataset.

Now, we can assume that the data-points (x_i, y_i) are independent and hence we can factorize the joint distribution as,

$$\hat{W} = \max_W \mathcal{P}(x_1, y_1, x_2, y_2, ..., x_n, y_n) = \max_W \prod_{i=1}^{n} \mathcal{P}(x_i, y_i)$$

Further, using the fact that ϵ is normally distributed as mentioned in equation (5.1), we can find the MLE $\hat{W}$ as,

$$\begin{aligned}
\hat{W} &= \max_W \prod_{i=1}^{n} \exp\left(-\frac{(y_i - W^T x_i)^2}{2\sigma^2}\right) \\
&= \max_W \exp\left(-\sum_{i=1}^{n} \frac{(y_i - W^T x_i)^2}{2\sigma^2}\right) \\
&= \min_W \sum_{i=1}^{n} (y_i - W^T x_i)^2
\end{aligned}$$

Case 2: Let us assume that ϵ follows Laplace distribution, i.e.

$$\epsilon \sim Laplace(0, b) \tag{5.4}$$

Note:

$$Laplace(\mu, b) = \frac{1}{2b} \exp\left(-\frac{|x - \mu|}{b}\right) \qquad \forall x \in \mathbf{R}$$

It can be shown that for this case, the MLE $\hat{W}$ of W is given as,

$$\hat{W} = \min_W \sum_{i \in D} |y_i - W^T x_i|$$

Now, let us try to find the solution for the optimisation problem in case 1.

$$\begin{aligned}
\min_W \sum_{i \in D} (y_i - W^T x_i)^2 &= \min_W \sum_{i \in D} (y_i^2 + (W^T x_i)^2 - 2W^T x_i y_i) \\
&= \min_W \sum_{i \in D} (y_i^2 + x_i^T W W^T x_i - 2W^T x_i y_i)
\end{aligned}$$

Since, this is the case of unconstrained optimisation, we take gradient of the objective function w.r.t W to get the following equality,

$$\begin{aligned}
&\sum_{i \in D} (0 - 2x_i y_i - 2(x_i x_i^T) W^*) = 0 \\
&\implies \sum_{i \in D} (x_i x_i^T) W^* = \sum_{i \in D} (x_i y_i) \\
&\implies \mathbf{W}^* = \Big(\sum_{\mathbf{i} \in \mathbf{D}} (\mathbf{x_i x_i^T})\Big)^{-\mathbf{1}} \sum_{\mathbf{i} \in \mathbf{D}} (\mathbf{x_i y_i})
\end{aligned} \tag{5.5}$$

5.3 Regularization : Overcoming singularity and ill-conditioning

5.3.1 Under what conditions can W* be singular or ill-conditioned?

We have the Maximum likelihood estimate of W* given by,

$$\mathbf{W}^* = (\mathbf{X X^T})^{-\mathbf{1}} (\mathbf{y^T X})$$

Hence, the rank of $\mathbf{XX^T}$ needs to be investigated to check for singularity and the condition of the matrix.

Case 1: $n < d$

The number of features in a datapoint (d) is greater than the number of datapoints (n).

$$rank(XX^T) \leq min(rank(X), rank(X^T)) \tag{5.6}$$

$$rank(X) = rank(X^T) = min(n, d) = n \tag{5.7}$$

$$(5.6), (5.7) \implies rank(XX^T) \leq rank(X) \leq n \tag{5.8}$$

Since, $\mathbf{XX^T}$ is a d*d matrix with a rank less than n, it must be singular.

Case 2: $n \geq d$

$\mathbf{XX^T}$ will be non-singular with high probability. Why? To see this consider the determinant of $\mathbf{XX^T}$,

$$det(\mathbf{XX^T}) = p(x_1, x_2, \ldots, x_n)$$

where p is some multinomial function. Now if $det(\mathbf{XX^T}) = 0$,

$$p(x_1, x_2, \ldots, x_n) = 0$$

For intuition, considering the one variable polynomial case, we see that the probability of randomly picking a root is zero (one point in infinitely many). Similarly, in the multinomial case, the space of roots is infinitesimal compared to the whole space and so the matrix $\mathbf{XX^T}$ is almost surely invertible.

As mentioned above, in this case $\mathbf{XX^T}$ will be non-singular with high probability but maybe be ill-conditioned with some finite probability i.e.

$$eigvals(\mathbf{XX^T}) \in [-\epsilon, \epsilon]$$

In other words, ill-conditioning of $\mathbf{XX^T}$ will blow up its inverse resulting in numerical instability while computing $\mathbf{W}^*$.

5.3.2 What is Regularization?

Regularization is a trick to avoid singularity such as in Case-1 and improve the conditioning for Case-2 by adding some noise along the diagonal of the matrix i.e.

$$\mathbf{W}^* = (\lambda\mathbf{I} + \mathbf{XX^T})^{\mathbf{-1}}(\mathbf{y^TX})$$

Adding a miniscule noise will reduce singularity & with a sufficient regularization we can also get rid of ill-conditioning.

But by changing $\mathbf{W}^*$ in this way, how do we know if it still optimizes our loss function? To see this we note the following:

$$
\begin{aligned}
\mathcal{L}(W(\lambda \to 0)) &= \mathcal{L}(W \to (XX^T)^{-1}(y^T X)) \\
&= (y - ((XX^T)^{-1}(y^T X))^T X)^2 \\
&= y^2(1 - (X^T(X^T)^{-1}X^{-1})X)^2 \\
&= y^2(1 - X^{-1}X)^2 \\
&= 0
\end{aligned}
$$

So, as we can see for the condition $\lambda \to 0$, we get that the loss tends to zero and so it works as good as the original $\mathbf{W}^*$ but reduces singularity and ill conditioning.

5.3.3 How does Regularization ensure that W* is well-conditioned?

Regularization can be visualized as increasing all the eigenvalues by a constant i.e. ,

$$
Av = kv \implies (A + \lambda I)v = (\lambda + k)v \tag{5.9}
$$

Singular matrices have an eigenvalue equal to 0 and increasing it by a small amount would make all the eigenvalues non-zero and the matrix becomes non-singular.

Similarly, for an ill-conditioned matrix we have, $eigvals(\mathbf{XX^T}) \in [-\epsilon, \epsilon]$ so increasing all eigenvalues by some sufficient λ by adding some regularization would make it well conditioned.

5.3.4 Value of the Regularization coefficient (λ)

How large or small should the value of λ be?
For this, let us take an example scenario, where we have only one sample, that is, $|D| = 1$.

$$
\begin{aligned}
&L(w) = \sum_{i \in D}(y_i - W_2^T x_i)^2 \\
&\therefore L(w) = (y_1 - W_2^T x_1)^2 \\
\\
&\text{If } \lambda \to \infty, W_2 \to 0 \\
&\therefore L(w) \to y_1^2 \\
\\
&\text{If } \lambda \to 0, W_2 \to (x_1 x_1^T)^{-1} \\
&\therefore L(w) \to 0
\end{aligned}
$$

However, do note that for a dataset of just a single sample, $x_1 x_1^T$ would clearly not be invertible, because the rank of $x_1 x_1^T$ is 1
Hence, we need to take care of how we set the regularization constant, because if it is high, then the loss function would not return a value of 0, but if it is too small, then the previous problem of

the matrix being non-invertible may creep up.

Now, let us try to find the optimization function for which we obtain W_2 as the solution.

We already know that the initial optimization problem was given by

$$\min_W \sum_{i \in D} (y_i - W^T x_i)^2$$

The solution for the above problem was given by

$$W_1 = (\sum_{i \in D} x_i x_i^T)^{-1} \sum_{i \in D} x_i y_i$$

Now, for the equation obtained after regularization

$$\begin{aligned} W_2 &= (\lambda I + \sum_{i \in D} x_i x_i^T)^{-1} \sum_{i \in D} x_i y_i \\ \therefore 2\lambda I W_2 + 2\sum_{i \in D} x_i x_i^T W &= 2\sum_{i \in D} x_i y_i \\ \therefore 2\lambda I W_2 + 2\sum_{i \in D} x_i x_i^T W - 2\sum_{i \in D} x_i y_i &= 0 \\ \therefore \frac{d}{dW}(W^T(\lambda I)W + \sum_{i \in D}(y_i^2 + W^T x_i x_i^T W - 2W^T x_i y_i)) &= 0 \\ \therefore \frac{d}{dW}(\sum_{i \in D}(y_i^2 + W^T x_i x_i^T W - 2W^T x_i y_i) + \lambda ||W||^2) &= 0 \\ \therefore \frac{d}{dW}(\sum_{i \in D}(y_i - W^T x_i)^2 + \lambda ||W||^2) &= 0 \end{aligned}$$

Hence, we obtain that the optimization problem for the given W_2 obtained after regularization is given by:

$$\min_W \sum_{i \in D} (y_i - W^T x_i)^2 + \lambda ||W||^2$$

Helper code for Understanding effects of Regularization : *Hyperlink to helper code*.

5.4 Group Details

Group	**Name**
1	Modi Jay
1	Vinit Awale
1	Mehul
1	Mithun Balram
1	Vedang
2	Mayank Gupta
2	Malhar Kulkarni
2	N Vishal
2	Pradyumna Atreya
2	Tanisha Khandelwal