Intro to ML - CS 419

Lecture 3: Regression

January 20, 2023

Lecturer: Abir De *Scribe: Groups 5 and 6*

In statistical modeling, **regression analysis** is a set of statistical processes for estimating the relationships between a dependent variable (often called the 'outcome' or 'response' variable, or a 'label' in machine learning parlance) and one or more independent variables (often called 'predictors', 'covariates' or 'features').
The most common form of regression analysis is linear regression, in which one finds the line (or a more complex linear combination) that most closely fits the data according to a specific mathematical criterion.

1 Linear Regression with Illustration

Suppose a company wants to estimate how much it should spend on TV commercials to increase sales to a desired level y*
Given: Previous observations of the form $\{(x_i, y_i)\}$ where x_i is an instance of money spent on advertisements and y_i was the corresponding observed sale figure.

Suppose the observations support the following linear approximation:

$$\text{y} = \beta_o + \beta_1 * \text{x}$$

Then x* = $\frac{y*-\beta_o}{\beta_1}$ can be used to determine the money to be spent.
Now the question is - How to estimate β_o and β_1?

1.1 Math stuff

LOGIC: Minimize a loss function which is indicative of the task.
For example, one can minimize the max error, i.e., $\min_{\beta_o,\beta_1} \max_i [(y_i - \beta_1 x_i - \beta_o)^2]$

Another method is using **Least Square Approximation**, i.e., $\min_{\beta_o,\beta_1} \sum_{i=1}^{N} (y_i - \beta_1 x_i - \beta_o)^2$
How to find β_o, β_1 from this? - **Differentiate with respect to** β_o, β_1
Differentiating with respect to β_1 : $\sum_{i=1}^{N} (y_i - \beta_1 x_i - \beta_o)(-x_i) = 0$

$$\implies -\sum_i y_i x_i + \beta_1 \sum_i x_i^2 + \beta_o \sum_i x_i = 0$$

$$\implies \beta_1 = \frac{\sum_i y_i x_i - \beta_o \sum_i x_i}{\sum_i x_i^2}$$

Differentiating with respect to β_o : -$\sum_{i=1}^{N}(y_i - \beta_1 x_i - \beta_o) = 0$

$\implies \sum_i y_i - \beta_1 \sum_i x_i - N\beta_o = 0$

$\implies \beta_o = \frac{\sum_i y_i - \beta_1 \sum_i x_i}{N}$

Substituting β_1 in terms of β_o from the previous expression,

$$\beta_o = \frac{\sum_i y_i - \frac{\sum_i y_i x_i - \beta_o \sum_i x_i}{\sum_i x_i^2} \sum_i x_i}{N} \implies \beta_o = \frac{\sum_i y_i \sum_i x_i^2 - \sum_i y_i x_i \sum_i x_i + \beta_o (\sum_i x_i)^2}{N \sum_i x_i^2}$$

$$\implies \beta_o = \frac{\sum_i y_i \sum_i x_i^2 - \sum_i y_i x_i \sum_i x_i}{N \sum_i x_i^2 - (\sum_i x_i)^2}$$

Substituting β_o to find β_1:

$$\beta_1 = \frac{\frac{\sum_i x_i}{N} \frac{\sum_i y_i}{N} - \frac{\sum_i x_i y_i}{N}}{(\frac{\sum_i x_i}{N})^2 - \frac{\sum_i x_i^2}{N}}$$

If x_i are varying independently of y_i, the expected value of β_1 is 0. This is also evident from the expression of β_1 (numerator is COV(X,Y) $\implies$ if X,Y are independent, COV(X,Y)=0).

Also, if the standard deviation of X is zero, β_1 blows up. What is the physical characterization of this situation? - we are not getting the data as x_i is no longer a distribution! One cannot fit the data as it doesn't make any sense since we are just copying x_i. Therefore, the effect on mathematical fitting is that β_1 blows up, i.e., NaN, which makes sense!

What if the standard deviation of X is too high? - Given the large heterogenity in data, fitting with two parameters is difficult. This leads to the problem of **under-specification**.
However, if the covariance of X,Y is as high as the standard deviation of X, then we will be able to get some finite value of β_1.

What if X,Y are related by a non-linear relation?
Suppose: $y_i = 2e_i^x + 3sin(x_i) + 3$. What happens if we try to use linear regression to find this relation? - Since we are trying to fit a large range of data varying non-linearly with just 2 parameters, it again leads to the problem of under-specification.
However, if we consider a small variance, i.e., **a small range of x**, then we can still fit a linear curve to it.

2 Regression Using MLE

The objective is to estimate the parameters of the linear regression model $y_i = \beta_1 \text{x} + \beta_o$ where y_i is the dependent variable. The samples are made up of N identically independent observations (x_i, y_i). We saw that least square approximation gives us the true population parameters β_o and β_1. Recall that **Maximum Likelihood Estimation (MLE)** also gives the true population parameters. So we can say that the least square approximation is an MLE problem. But how are the two related? Under what assumptions are the two 'objectives' equivalent?
Suppose y_i need not be given $\beta_1 \text{x} + \beta_o$ but it is sampled from a gaussian distribution with mean lying along a line $\beta_1 \text{x} + \beta_o$.

$$y_i \sim \text{N}(\beta_1 x_i + \beta_o, \sigma^2)$$

$\implies$ Maximum likelihood estimator for a gaussian function with a given variance σ^2 such that mean of y_i is $\beta_1 x_i + \beta_o$
$\implies$ LE (likelihood estimator) =

$$\prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma^2} e^{-\frac{1}{2}\frac{(y_i - \beta_1 x_i - \beta_o)^2}{\sigma^2}}$$

$\implies$ we have to maximize the logarithm of the LE.(taking logarithm for mathematical convenience)

$\implies \max_{\beta_o, \beta_1} \log(\text{LE(gaussian)}) = \min_{\beta_1, \beta_o} \sum_{i=1}^{N} (y_i - \beta_1 x_i - \beta_o)^2$ ....(ignoring the constants)

We have arrived at the least square approximation method implying that the Maximum likelihood estimator and least square estimator are one and the same thing.

This method works because most noise is gaussian in nature. We can also model the noise with some different distribution and minimize the corresponding loss function to arrive at new coefficients of the regression model.

For example let the y_i be an exponential distribution with mean $(\beta_1 x_i + \beta_o)$
$\implies$ $y_i \sim$ exponential($\frac{1}{\beta_1 x_i + \beta_o}$) ...(mean of a exponential distribution)
$\implies$ LE (y_i, k_i) =

$$\prod_{i=1}^{N} \frac{1}{\beta_1 x_i + \beta_o} e^{-\frac{y_i}{\beta_1 x_i + \beta_o}}$$

$\implies$ MLE = $\max_{\beta_1, \beta_o} \log \prod_{i=1}^{N} \frac{1}{\beta_1 x_i + \beta_o} e^{-\frac{y_i}{\beta_1 x_i + \beta_o}}$
The MLE estimates of β_1 and β_o are the solutions of the following system of equations

$$\frac{\partial}{\partial \beta_1} \log \left(\prod_{i=1}^{N} \frac{1}{\beta_1 x_i + \beta_o} e^{-\frac{y_i}{\beta_1 x_i + \beta_o}} \right) = 0$$

$$\frac{\partial}{\partial \beta_0} \log \left(\prod_{i=1}^{N} \frac{1}{\beta_1 x_i + \beta_o} e^{-\frac{y_i}{\beta_1 x_i + \beta_o}} \right) = 0$$

To simplify the above system of equations further, we can take the partial derivatives of the log-likelihood function with respect to β_1 and β_o:

$$\implies \frac{\partial}{\partial \beta_1} \log \left(\prod_{i=1}^{N} \frac{1}{\beta_1 x_i + \beta_o} e^{-\frac{y_i}{\beta_1 x_i + \beta_o}} \right) = \frac{\partial}{\partial \beta_1} \left(\sum_{i=1}^{N} \log \frac{1}{\beta_1 x_i + \beta_o} - \frac{y_i}{\beta_1 x_i + \beta_o} \right)$$

$$= \sum_{i=1}^{N} -\frac{x_i}{\beta_1 x_i + \beta_o} + \frac{y_i x_i}{(\beta_1 x_i + \beta_o)^2}$$

and

$$\frac{\partial}{\partial \beta_0} \log \left(\prod_{i=1}^{N} \frac{1}{\beta_1 x_i + \beta_o} e^{-\frac{y_i}{\beta_1 x_i + \beta_o}} \right) = \frac{\partial}{\partial \beta_0} \left(\sum_{i=1}^{N} \log \frac{1}{\beta_1 x_i + \beta_o} - \frac{y_i}{\beta_1 x_i + \beta_o} \right)$$

$$= \sum_{i=1}^{N} -\frac{1}{\beta_1 x_i + \beta_o} + \frac{y_i}{(\beta_1 x_i + \beta_o)^2}$$

Now we can set the above two equations to zero, and we get:

$$\sum_{i=1}^{N} -\frac{x_i}{\beta_1 x_i + \beta_o} + \frac{y_i x_i}{(\beta_1 x_i + \beta_o)^2} = 0$$

$$\sum_{i=1}^{N} -\frac{1}{\beta_1 x_i + \beta_o} + \frac{y_i}{(\beta_1 x_i + \beta_o)^2} = 0$$

One can solve the above equations to get β_o anf β_1.

3 Modelling non-linear relations

Suppose the relation between X and Y is non-linear. What should be done if we still want to use linear regression tools to find the relation?

We can now model y as $y = w_1\phi_1(x) + w_2\phi_2(x) + ... + w_d\phi_d(x)$, where $\phi_1, \phi_2 ..... \phi_d$ are non-linear features (different basis functions) of x. The learning happens on $w_1, w_2, ..., w_d$ rather than $\phi_1, \phi_2, ..., \phi_d$ (the problem is linear in weights of the model!). For now, we can take $\phi_1, \phi_2 .... \phi_d$ to be pre-defined functions and $w_1, w_2, ..., w_d$ to be unknown. (The number of parameters has now increased)

The model, therefore, is

$$y = w^T \phi(x)$$

where w is the column vector

$$w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}$$

and ϕ is the column vector

$$\phi = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_d \end{bmatrix}$$

Therefore, the problem now translates to: $\min_w \sum_{i=1}^{N} (y_i - w^T \phi(x_i))^2$

For minimization over vector w, we need to use vector differentiation.
The above objective translates to:

$$l(w) = (y - \phi w)^T (y - \phi w)$$

where y is the column vector

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_d \end{bmatrix}$$

and ϕ is the matrix

$$\phi = \begin{bmatrix} \phi_1(x_1) & \dots & \phi_d(x_1) \\ \phi_1(x_2) & \dots & \phi_d(x_2) \\ \vdots & \vdots & \vdots \\ \phi_1(x_N) & \dots & \phi_d(x_N) \end{bmatrix}$$

To minimize $l(w)$ with respect to w, we differentiate as follows:

$$\frac{d}{dw}[(y - \phi w)^T (y - \phi w)]$$
$$= \frac{d}{dw}[y^T y - 2y^T \phi w + w^T \phi^T \phi w] \text{ (as } y^T \phi w = w^T \phi^T y \text{ as dot product is commutative)}$$
$$= -2\phi^T y + 2\phi^T \phi w = 0 \implies w = (\phi^T \phi)^{-1} \phi^T y$$