LINEAR REGRESSION , LOGISTIC REGRESSION AND GAUSSIAN DISCRIMINANT ANALYSIS

ASHISH KUMAR

$DepartmentOfMathematics, IITRoorkee, Roorkee - 247667$

$a\_kumar1@ma.iitr.ac.in$

Linear Regression, Logistic Regression, Multivariate Normal Distribution, Gaussian Discriminant Analysis Model

Abstract. The article provides an overview of several fundamental statistical modeling techniques, including linear regression, logistic regression, multivariate normal distribution, and Gaussian discriminant analysis model. It explains how these techniques are used to analyze relationships between variables and make predictions based on data. The article covers the underlying assumptions, methods, and applications of each technique, providing readers with a clear understanding of their strengths and limitations. Whether you are a novice or an experienced data analyst, this article should act as an introduction for gaining a deeper understanding of these essential statistical techniques.

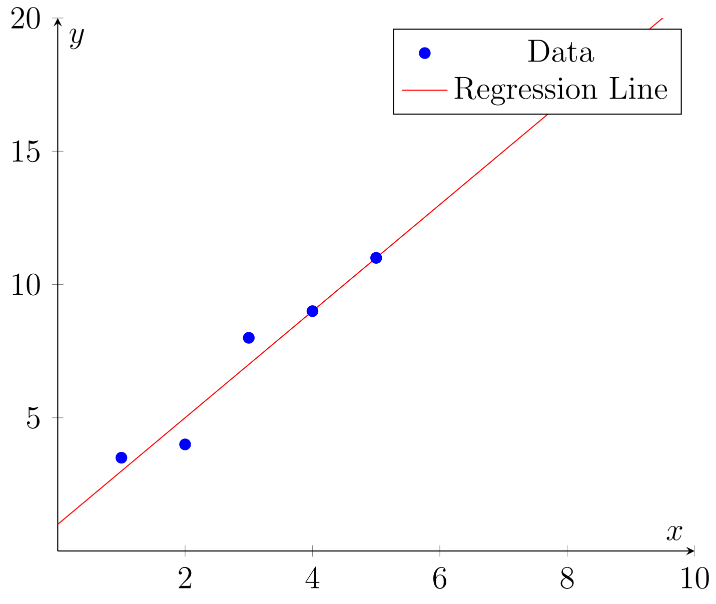
AMS classification: 62J05,62J12,62H10,62H30

Contents

1. Linear Regression

1.1. **Introduction.** Linear Regression is a statistical supervised learning technique to predict the quantitative variable by forming a linear relationship with one or more independent features.

The goal of linear regression is to find the line of best fit that describes the relationship between the two variables. The line of best fit is a straight line that minimizes the distance between the predicted values and the actual values.



An example of Linear Regression

1.2. **Line of best fit.** To relate independent variable with a dependent variable , the simplest linear function we can think of would be

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 ... \qquad (1)$$

where θ_i's are called weights. Here θ_0 would be the intercept term. Hence

$$h(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x, \qquad (2)$$

where $\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \end{bmatrix}$ and $X = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \end{bmatrix}$

1.3. **Cost Function.** We define cost function as

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{n}(h_\theta(x^{(i)}) - y^{(i)}) \tag{3}$$

which is basically how far the $h(x^{(i)})$'s are from the $y^{(i)}$'s. For our line 1 to be the best line , the cost function should be the minimum.

1.4. **LMS.** We need to minimize $J(\theta)$. We can do that by the **gradient descent algorithms**.We will not go deep into the algorithm due to the scope of this article.Applying it to the given sample for minimizing the cost function yields,

$$\theta_j := \theta_j + \alpha(y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)} \tag{4}$$

This is called as **LMS** update rule or **Widrow-Hoff** learning rule.By grouping the updates of the coordinates into an update of the vector θ, we can write the same equation better as

$$\theta := \theta + \alpha\sum_{i=1}^{n}(y^{(i)} - h_\theta(x^{(i)}))x^{(i)} \tag{5}$$

This method looks at every example in the entire training set on every step, and is called **batch gradient descent**. [N.G19]

1.5. **Shortcomings of Linear Regression.** The shortcomings are the assumptions we take:

(1) Linearity Assumption: Linear regression assumes that the relationship between the dependent variable
(2) Outliers: Linear regression is sensitive to outliers in the data.
(3) Over-fitting: Linear regression can be prone to over-fitting, especially when there are many independent variables.
(4) Causality: Linear regression only captures the correlation between the dependent variable and independent variables.

2. Learning Algorithms

2.1. **Discriminative Learning Algorithms :** Consider a given set of data points and you need to classify them based on certain features. Classic algorithms like logistic regression or perceptron algorithm classifies the given set by fitting a straight line (or a decision boundary) in the data model. Then it classifies a new entry based on which part it fell. These "direct" algorithms which maps space of inputs $\mathbf{X}$ to the set $\{0, 1\}$ are called **discriminative learning algorithms**.

2.2. **Generative Learning Algorithms :** Alternate to the previous approach, we can instead create a model based on the given features and then given a new entry, goes ahead to compare it to both models to find the likeliness and decide based on this info.
Now consider the algorithms that instead try to model $p(x|y)$.These algorithms are known as **generative learning algorithms**.
How does this work ? After modelling the **class priors** i.e, $p(y)$ and $p(x|y)$, now given a new entry (i.e, x) we need to find $p(y|x)$ which can be found using the Bayes rule :

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

or

$$p(y|x) = \frac{p(x|y)p(y)}{p(x|y=1)p(y=1) + p(x|y=0)p(y=0)} \tag{6}$$

3. Multivariate Normal Distribution

Also known as Multivariate Gaussian Distribution ($N(\mu, \Sigma)$), the Multivariate Normal Distribution in d-dimensions is given by :

$$p(x; \mu, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right) \tag{7}$$

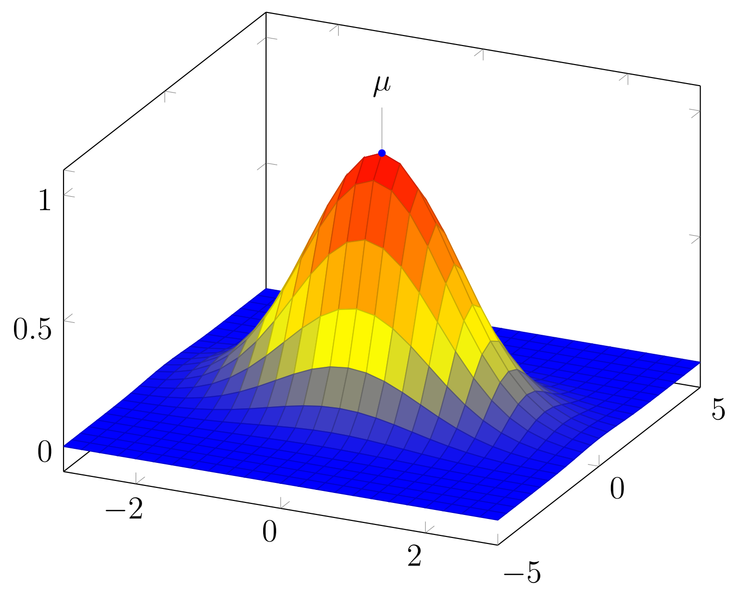
where $|\Sigma|$ denotes the determinant of the covariance matrix,which is symmetrical and positive semi-definite and the **mean vector** $\mu \in \mathbb{R}^d$ and **covariance matrix** $\Sigma \in \mathbb{R}^{dxd}$.
If a random variable X is a vector , then it's mean is given by μ :

$$E(X) = \int_x xp(x; \mu, \Sigma)dx = \mu \tag{8}$$

and the covariance is given by :

$$Cov(Z) = E[(Z - E[Z])(Z - E[Z])^T] \tag{9}$$

Here is an example of what the density of a Gaussain distribution looks like :



An example of a normal distribution with $\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ *and* $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

4. Gaussian discriminant analysis model

A generative learning algorithm where $p(x|y)$ is distributed according to a multivariate normal distribution is Gaussian discriminant analysis (GDA).

Given input features x as continuous random variables,GDA models $p(x|y)$ using the above multivariate normal distribution.

$$y \sim Bernoulli(\phi) \ or \ p(y) = \phi^y 1 - \phi^{1-y}$$

$$x|y = 0 \sim N(\mu_0, \Sigma)$$

$$x|y = 1 \sim N(\mu_1, \Sigma)$$

Finding the log-likelihood of the data :

$$l(\phi, \mu_0, \mu_1, \Sigma) = \log \Pi_{i=1}^n Y p(x^{(i)}, y^{(i)}; \phi, \mu_0, \mu_1, \Sigma)$$

$$= \log \Pi_{i=1}^n Y p(x^{(i)}|y^{(i)}; \mu_0, \mu_1, \Sigma) p(y^{(i)}; \phi)$$

By maximizing l with respect to the parameters, we will find the maximum likelihood estimate of the parameters to be :

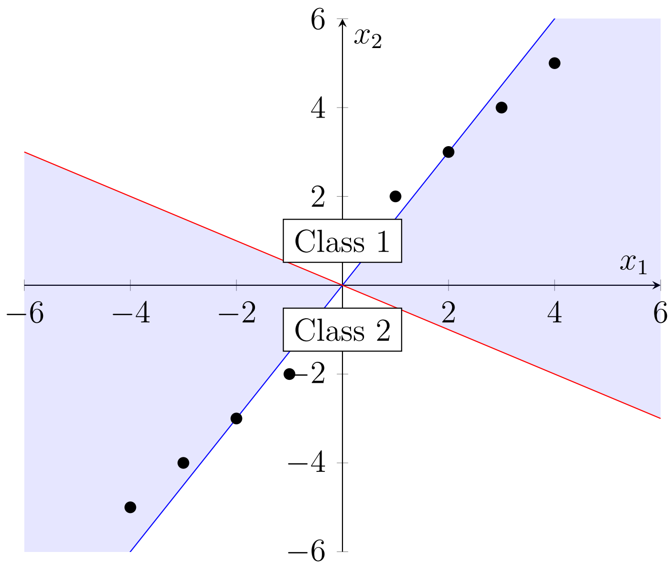
$$\phi = \frac{1}{n} \sum_{i=1}^{n} 1\{y^{(i)} = 1\} \tag{10}$$

$$\mu_0 = \frac{\sum_{i=1}^{n} 1\{y^{(i)} = 0\}x(i)}{\sum_{i=1}^{n} 1\{y^{(i)} = 0\}} \tag{11}$$

$$\mu_1 = \frac{\sum_{i=1}^{n} 1\{y^{(i)} = 1\}x(i)}{\sum_{i=1}^{n} 1\{y^{(i)} = 1\}} \tag{12}$$

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n}(x^{(i)} - \mu_{y(i)})(x^{(i)} - \mu_{y(i)})^T \tag{13}$$

Here is an example of classification done for a data set using GDA.



5. GDA and logistic regression

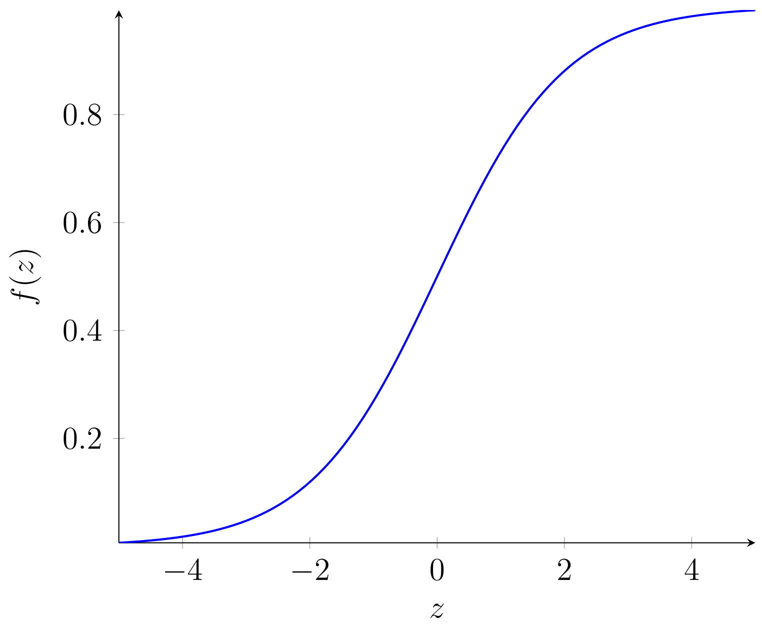
Now that we have understood GDA in brief, we will see how GDA is connected to logistic regression and which is better.

5.1. **Logistic Regression.** Logistic regression is a binary classification strategy. It is closely similar to Linear Regression. The basic idea is to fit a sigmoid function in the model instead of fitting a striaght line.

5.2. **Sigmoid Function.** Logistic or sigmoid function is of form

$$g(z) = \frac{1}{1 + e^{-z}}$$

Upon plotting, it looks like :



Also note that, $g'(z) = g(z)(1 - g(z))$

We change out hypothesis such as

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

.

Given this model, we can write :

$$p(y|x;\theta) = (h_\theta(x))^y (1 - h_\theta(x))^{1-y}$$

for n training examples , (asssuming they were generated independently)

$$L(\theta) = p(\overrightarrow{y}|\mathbb{X};\theta) = \Pi_{i=1}^n p(y_{(i)}|x_{(i)};\theta)$$

Taking the log and maximizing with respect to the parameters using gradiant descent rule :

$$\theta_j = \theta_j + \alpha(y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)} \tag{14}$$

5.3. **Which is better? : GDA or Logistic Regression?** If we view the quantity $p(y = 1|x; \varphi, \mu_0, \mu_1, \Sigma)$ as a function of x, we'll find that it can be expressed in the form

$$p(y = 1|x; \phi, \Sigma, \mu_0, \mu_1) = \frac{1}{1 + e^{-\theta^T x}}$$

which is very similar to the logistic regression form. So, which is better ? Let us compare them in a table.

Differences b/w GDA and Logistic Regression	
Logistic Regression	GDA
Used for binary classification problems	Optimization algorithm used to minimize the cost function
Logistic regression updates the model parameters based on the training data, by minimizing the error between the predicted output and the actual output.	Gradient descent, on the other hand, updates the model parameters by minimizing the cost function, which is a measure of the error between the predicted output and the actual output.
Takes long time to converge to local minima	Faster to converge than LR
Good choice when you have a binary classification problem	Good choice when you need to optimize the parameters of a cost function

Basically GDA makes stronger modeling assumptions about the data than does logistic regression. When these assumptions are correct, GDA forms a better model [N.G18] but if you are uncertain about the assumptions, esp. in large datasets, it would be better to use logistic regression to classify.

References

[N.G18] Andrew N.G. *Machine Learning Yearning*, chapter 4. 1 edition, 2018.

[N.G19] Andrew N.G. Cs229 : Machine learning harvard course, 2019.