MIT-WPU

॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Dr. Vishwanath Karad

MIT WORLD PEACE UNIVERSITY | PUNE

TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

MACHINE LEARNING

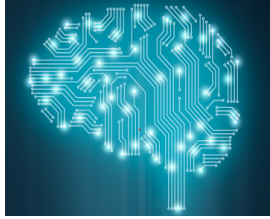
Professional Core(CET3006B)

T. Y. B.Tech AIDS, Sem-V

2024-2025

SoCSE – Dept. of Computer Engineering & Technology

MIT-WPU
॥ विश्वशान्तिर्धुवं ध्रुवा ॥



Course Contents:

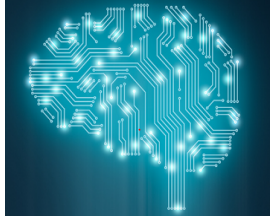
Unit 1. Introduction to ML

Unit 2. Supervised Learning: Classification

Unit 3. Unsupervised Learning: Clustering

Unit 4. Performance Analysis and Model Evaluation

Unit 5. Trends in ML



Course Contents:

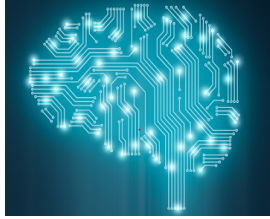
Text Books:

1. E. Alpaydin, Introduction to Machine Learning, PHI, 2004.
2. Peter Flach: Machine Learning: The Art and Science of Algorithms that Make Sense of Data, Cambridge University Press, Edition 2012.
3. T. Mitchell, Machine Learning, McGraw-Hill, 1997.
4. Josh Patterson, Adam Gibson, “Deep Learning: A Practitioners Approach”, O‟REILLY, SPD, ISBN: 978-93-5213-604-9, 2017 Edition 1st

Reference Books:

1. C. M. Bishop: Pattern Recognition and Machine Learning, Springer 1st Edition-2013.
2. Ian H Witten, Eibe Frank, Mark A Hall: Data Mining, Practical Machine Learning Tools and Techniques, Elsevier, 3rd Edition.
3. Shaishalev-shwartz, Shai Ben-David: Understanding Machine Learning from Theory to algorithms, Cambridge University Press, ISBN-978-1-107-51282-5, 2014.

Course Contents:



Supplementary Reading:

1. AurelienGeron, “Hands-on Machine Learning with Scikit-learn and Tensor flow, O’Reilly Media

Web Resources:

1. Popular dataset resource for ML beginners: http://archive.ics.uci.edu/ml/index.php

Web links:

1. https://www.kaggle.com/datasets
2. http://deeplearning.net/datasets/

MOOCs:

1. https://swayam.gov.in/nd1_noc20_cs29/preview
2. https://swayam.gov.in/nd1_noc20_cs44/preview

Course Contents:

Assessment Scheme:

Class Continuous Assessment (CCA): 30 Marks

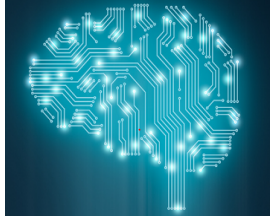
Mid Term	Component 1 (Active Learning)	Component 2
15 Marks	10 Marks	5 Marks

Laboratory Continuous Assessment (LCA): 30 Marks

Practical Performance	Active learning / Mini Project/Additional implementation/ On paper design	End term practical /oral examination
10 Marks	10 Marks	10 Marks

Term End Examination: 40 Marks

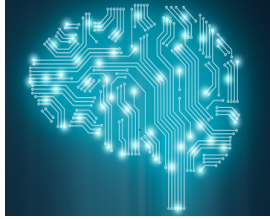
MIT-WPU
॥ विश्वशान्तिर्धुवं ध्रुवा ॥



Syllabus-Unit 4

Regression

- Multivariable regression;
- Model evaluation;
- Least squares regression;
- Regularization; LASSO;
- Applications of regression,
- Overfitting and Underfitting
- Hidden Markov Models (HMM) with forward-backward and Viterbi algorithms;
- Sequence classification using HMM;
- Conditional random fields;
- Applications of sequence classification such as part-of-speech tagging,
- Anomaly and outlier detection methods



Regression

- **“Regression is the process of estimating the relationship between a dependent variable(Y) and independent variables(X)”.**
- **Regression analysis** is a **statistical technique** that attempts to explore and model the relationship between two or more variables.

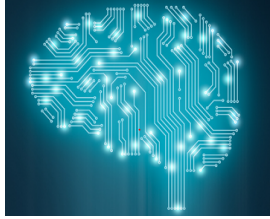
Understanding Regression

- Dependent variable (The value to be predicted)(Y)
- Independent Variable (The predictors)(X)

Assume relationship between independent and dependent variable is **straight line**.

$$y = a + bx$$

- a – intercept on y axis(indicates value of y when x = 0)
- b – slope (how much line rises for each increase in x)



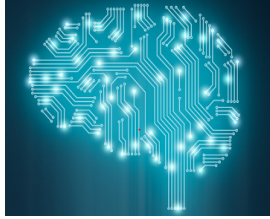
Cntd..

- **Linear Regression** (Straight line model)
  - Simple linear Regression (Single independent variable and dependent variable is continuous)
  - Multiple regression (More than one independent variable and dependent variable is continuous)
- **Logistic Regression** (Binary categorical outcome)
  - Logistic model is used to **model the probability of a certain class or event** existing such as pass/fail, win/lose, alive/dead or healthy/sick.
  - This **can be extended** to model several classes of events such as determining whether an image contains a cat, dog, lion, etc..

For example, the weight, height, and age represent continuous variable. a person's gender, occupation, or marital status are categorical or discrete variables

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Contd..

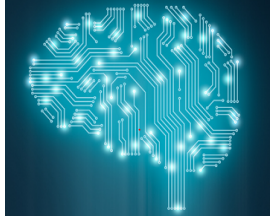


- Regression analysis is one of the most basic tools in the area of machine learning used for Prediction

- Using regression you fit a function on the available data and try to predict the outcome for the future or hold-out data points

- This fitting of function serves two purposes.
  - You can estimate missing data within your data range (Interpolation)
  - You can estimate future data outside your data range (Extrapolation)

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

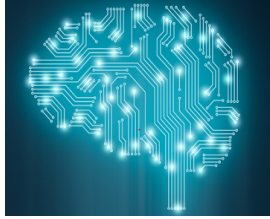


Types of Regression Techniques

- There are many types of regression analysis techniques, and the use of each method depends upon the number of factors
- These factors include the type of target variable, shape of the regression line, and the number of independent variables
- Below are the different regression techniques:
  - -Linear Regression
  - -Logistic Regression
  - -Ridge Regression
  - -Lasso Regression
  - -Polynomial Regression
  - -Bayesian Linear Regression

MIT-WPU
॥ विश्वशान्तिर्धुवं ध्रुवा ॥

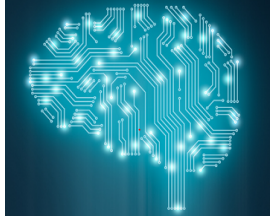
Linear Regression



- “Linear regression predicts the relationship between two variables by assuming a linear connection between the independent and dependent variables
- It seeks the optimal line that minimizes the sum of squared differences between predicted and actual values
- Applied in various domains like economics and finance, this method analyzes and forecasts data trends
- It can extend to multiple linear regression involving several independent variables and logistic regression, suitable for binary classification problems
- There are different types of linear regression. The two major types of linear regression are

  -Simple linear regression

  -Multiple linear regression



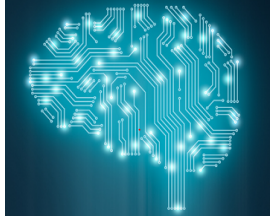
Example: Advertising through TV, Radio, Newspaper to increase sales

- *How accurately can we predict future sales?*
- *Is the relationship linear?*
- *Is there synergy among the advertising media?*

linear regression can be used to answer each of these questions.

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Simple Linear Regression



- In a simple linear regression, there is one independent variable and one dependent variable
- The model estimates the slope and intercept of the line of best fit, which represents the relationship between the variables
- The slope represents the change in the dependent variable for each unit change in the independent variable, while the intercept represents the predicted value of the dependent variable when the independent variable is zero.
- Linear regression is a quiet and the simplest statistical regression method used for predictive analysis in machine learning.

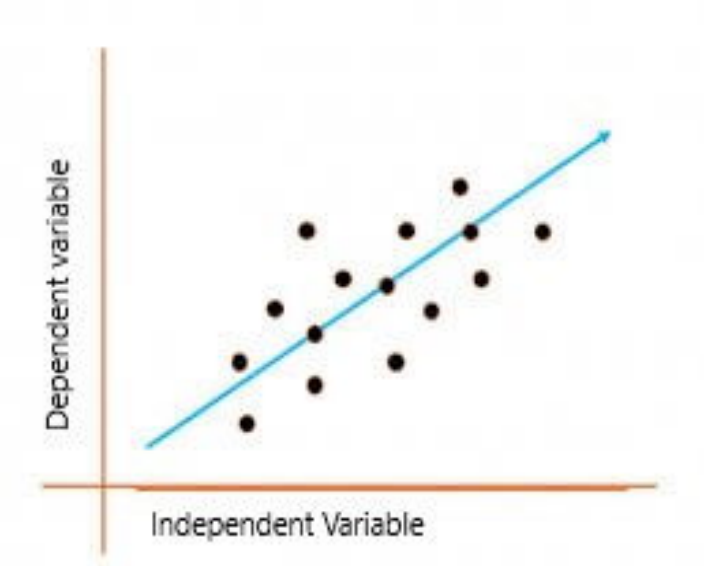
Simple Linear Regression

- Linear regression shows the linear relationship between the independent(predictor) variable i.e. X-axis and the dependent(output) variable i.e. Y-axis, called linear regression

- If there is a single input variable **X**(independent variable), such linear regression is ***simple linear regression***

- The graph presents the linear relationship between the output(y) and predictor(X) variables

- The blue line is referred to as the best-fit straight line. Based on the given data points, we attempt to plot a line that fits the points the best.



Simple Regression Calculation

- To calculate best-fit line linear regression uses a traditional slope-intercept form which is given below,

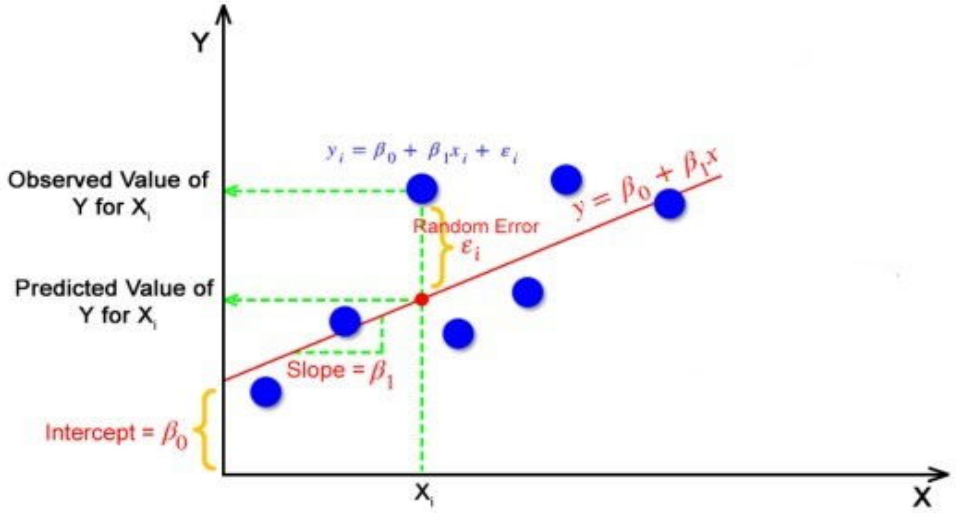
$$Yi = \beta 0 + \beta 1 Xi$$

where Yi = Dependent variable, β0 = constant/Intercept, β1 = Slope/Intercept, Xi = Independent variable.

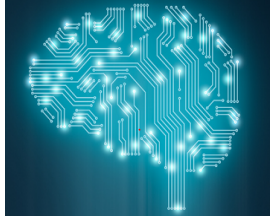
- This graph explains the linear relationship between the dependent(output) variable y and the independent(predictor) variable X using a straight line Y= B0 + B1 X.

- The goal of the linear regression algorithm is to get the best values for B0 and B1 to find the best fit line.

- The best fit line is a line that has the least error which means the error between predicted values and actual values should be minimum.



Least-Squares Method



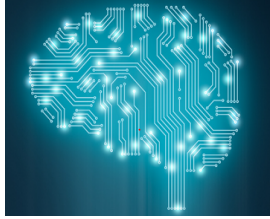
- Finds the line of best fit for a dataset, providing a visual demonstration of the relationship between the data points.
- The differences between the actual and estimated function values on the training examples are called residuals

$$\epsilon_i = f(x_i) - \hat{f}(x_i).$$

- The least-squares method, consists in finding ˆ f such that $\sum_{i=1}^{n} \epsilon_i^2$ is minimised
- Statistical method used to find the equation of the line of be l to the given data.
- This method is called so as it aims at reducing the sum of squares of deviations as much as possible.
- The line obtained from such a method is called a regression line

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Least-Squares Method



Formula of Least Square Method

Step 1: Denote the independent variable values as x_i and the dependent ones as y_i.

Step 2: Calculate the average values of x_i and y_i as X and Y.

Step 3: Presume the equation of the line of best fit as $y = mx + c$, where m is the slope of the line and c represents the intercept of the line on the Y-axis.

Step 4: The slope m can be calculated from the following formula:

$$m = [\Sigma (X - x_i) \times (Y - y_i)] / \Sigma (X - x_i)^2$$

Step 5: The intercept c is calculated from the following formula:

$$c = Y - mX$$

Thus, we obtain the line of best fit as $y = mx + c$, where values of m and c can be calculated from the formulae defined above.

Least-Squares Method

Problem 1: Find the line of best fit for the following data points using the least squares method: (x,y) = (1,3), (2,4), (4,8), (6,10), (8,15).

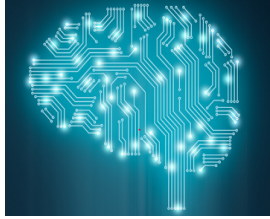
Here, we have x as the independent variable and y as the dependent variable. First, we calculate the means of x and y values denoted by X and Y respectively.

X = (1+2+4+6+8)/5 = 4.2

Y = (3+4+8+10+15)/5 = 8

x_i	y_i	$X - x_i$	$Y - y_i$	$(X-x_i)*(Y-y_i)$	$(X - x_i)^2$
1	3	3.2	5	16	10.24
2	4	2.2	4	8.8	4.84
4	8	0.2	0	0	0.04
6	10	-1.8	-2	3.6	3.24
8	15	-3.8	-7	26.6	14.44
Sum (Σ)		0	0	55	32.8

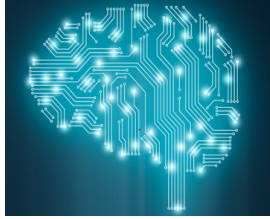
Least-Squares Method



The slope of the line of best fit can be calculated from the formula as follows:
$m = (\Sigma (X - xi)*(Y - yi)) / \Sigma(X - xi)2$
$m = 55/32.8 = 1.68$ (rounded upto 2 decimal places)
Now, the intercept will be calculated from the formula as follows:
$c = Y - mX$
$c = 8 - 1.68*4.2 = 0.94$
Thus, the equation of the line of best fit becomes, $y = 1.68x + 0.94$.

Problem 2: Find the line of best fit for the following data of heights and weights of students of a school using the least squares method:
Height (in centimeters): [160, 162, 164, 166, 168]
Weight (in kilograms): [52, 55, 57, 60, 61]

Multiple Linear Regression/ Multivariable regression



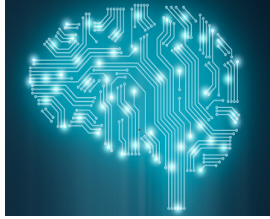
- Multiple linear regression is a technique to understand the relationship between a single dependent variable and multiple independent variables

- The formulation for multiple linear regression is also similar to simple linear regression with the small change that instead of having one beta variable, you will now have betas for all the variables used
- The formula is given as:
  $Y = B0 + B1X1 + B2X2 + \ldots + BpXp + \varepsilon$

Eg.
Prediction of height based on age and gender. Features(X):age, gender class(y): height Regression equation will be:
height = w1*age + w2*gender + b

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Considerations of Multiple Linear Regression



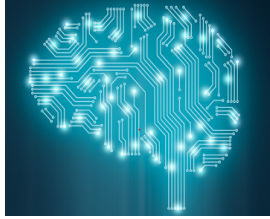
Multiple Linear Regression assumptions

1. Overfitting: When more and more variables are added to a model, the model may become far too complex and usually ends up memorizing all the data points in the training set. This phenomenon is known as the overfitting of a model. This usually leads to high training accuracy and very low test accuracy.

1. Multicollinearity: It is the phenomenon where a model with several independent variables, may have some variables interrelate

1. Feature Selection: With more variables present, selecting the optimal set of predictors from the pool of given features (many of which might be redundant) becomes an important task for building a relevant and better model

Multiple Linear Regression/ Multivariable regression



Simple linear regression :-

$$y = a + bx + e$$

where a = intercept
b = slope
e = residual

Multiple regression :-

$$y = a + b_1x_1 + b_2x_2 + b_3x_3 + \ldots + e$$

where a = intercept
b_1 = slope for x_1
b_2 = slope for x_2
b_3 = slope for x_3
...
e = residual

Example:

Calf birth weight = $a + b_1x$(mother's weight) + b_2x(father's weight) + b_3x(gestation) + e

May be used to :-

a) provide prediction of calf birth weight

b) assess influence of mother's weight (or gestation, etc) after **adjustment** for other factors

Multiple Linear Regression/ Multivariable regression

Regression Equation

$$Y = b_0 + b_1 X_1 + b_2 X_2 + \cdots + b_k X_k + e$$

Regression Coefficients

$$b_1 = \frac{(\sum x_2^2)(\sum x_1 y) - (\sum x_1 x_2)(\sum x_2 y)}{(\sum x_1^2)(\sum x_2^2) - (\sum x_1 x_2)}$$

$$b_2 = \frac{(\sum x_1^2)(\sum x_2 y) - (\sum x_1 x_2)(\sum x_1 y)}{(\sum x_1^2)(\sum x_2^2) - (\sum x_1 x_2)}$$

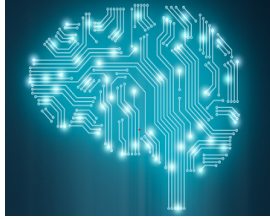
$$a = b_0 = Y - b_1 X_1 - b_2 X_2$$

Sums of Squares

$$ss_{reg} = b_1 \sum x_1 y + b_2 \sum x_2 y$$

$$ss_{res} = \sum y^2 - ss_{reg}$$

Multiple Linear Regression/ Multivariable regression

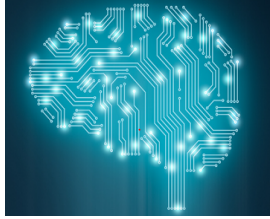


Suppose we have the following dataset with one response variable y and two predictor variables X1 and X2:

y	X_1	X_2
140	60	22
155	62	25
159	67	24
179	70	20
192	71	15
200	72	14
212	75	14
215	78	11

Use the following steps to fit a multiple linear regression model to this dataset.

Multiple Linear Regression/ Multivariable regression



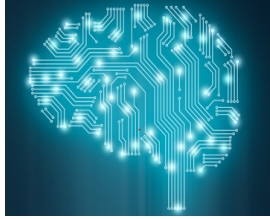
Step 1: Calculate X12, X22, X1y, X2y and X1X2.

	y	X_1	X_2
	140	60	22
	155	62	25
	159	67	24
	179	70	20
	192	71	15
	200	72	14
	212	75	14
	215	78	11
Mean	181.5	69.375	18.125
Sum	1452	555	145

X_1^2	X_2^2	X_1y	X_2y	X_1X_2
3600	484	8400	3080	1320
3844	625	9610	3875	1550
4489	576	10653	3816	1608
4900	400	12530	3580	1400
5041	225	13632	2880	1065
5184	196	14400	2800	1008
5625	196	15900	2968	1050
6084	121	16770	2365	858
38767	2823	101895	25364	9859

Sum

Multiple Linear Regression/ Multivariable regression



Step 2: Calculate Regression Sums.

Next, make the following regression sum calculations:

$\Sigma x_1^2 = \Sigma X_1^2 - (\Sigma X_1)^2 / n = 38{,}767 - (555)^2 / 8 =$ **263.875**

$\Sigma x_2^2 = \Sigma X_2^2 - (\Sigma X_2)^2 / n = 2{,}823 - (145)^2 / 8 =$ **194.875**

$\Sigma x_1 y = \Sigma X_1 y - (\Sigma X_1 \Sigma y) / n = 101{,}895 - (555*1{,}452) / 8 =$ **1,162.5**

$\Sigma x_2 y = \Sigma X_2 y - (\Sigma X_2 \Sigma y) / n = 25{,}364 - (145*1{,}452) / 8 =$ **-953.5**

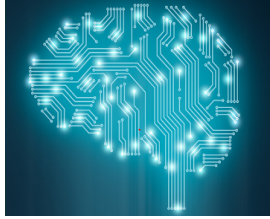
$\Sigma x_1 x_2 = \Sigma X_1 X_2 - (\Sigma X_1 \Sigma X_2) / n = 9{,}859 - (555*145) / 8 =$ **-200.375**

	y	X_1	X_2
	140	60	22
	155	62	25
	159	67	24
	179	70	20
	192	71	15
	200	72	14
	212	75	14
	215	78	11
Mean	181.5	69.375	18.125
Sum	1452	555	145

X_1^2	X_2^2	X_1y	X_2y	X_1X_2
3600	484	8400	3080	1320
3844	625	9610	3875	1550
4489	576	10653	3816	1608
4900	400	12530	3580	1400
5041	225	13632	2880	1065
5184	196	14400	2800	1008
5625	196	15900	2968	1050
6084	121	16770	2365	858
38767	2823	101895	25364	9859

Reg Sums	263.875	194.875	1162.5	-953.5	-200.375

Multiple Linear Regression/ Multivariable regression



Step 3: Calculate b0, b1, and b2.

The formula to calculate b1 is: $[(\Sigma x22)(\Sigma x1y) - (\Sigma x1x2)(\Sigma x2y)] / [(\Sigma x12)(\Sigma x22) - (\Sigma x1x2)2]$

Thus, b1 = $[(194.875)(1162.5) - (-200.375)(-953.5)] / [(263.875)(194.875) - (-200.375)2] = 3.148$

The formula to calculate b_2 is: $[(\Sigma x_1^2)(\Sigma x_2 y) - (\Sigma x_1 x_2)(\Sigma x_1 y)] / [(\Sigma x_1^2)(\Sigma x_2^2) - (\Sigma x_1 x_2)^2]$

Thus, $\mathbf{b_2} = [(263.875)(-953.5) - (-200.375)(1152.5)] / [(263.875)(194.875) - (-200.375)^2] =$ **-1.656**

The formula to calculate b_0 is: $y - b_1X_1 - b_2X_2$

Thus, $\mathbf{b_0} = 181.5 - 3.148(69.375) - (-1.656)(18.125) =$ **-6.867**

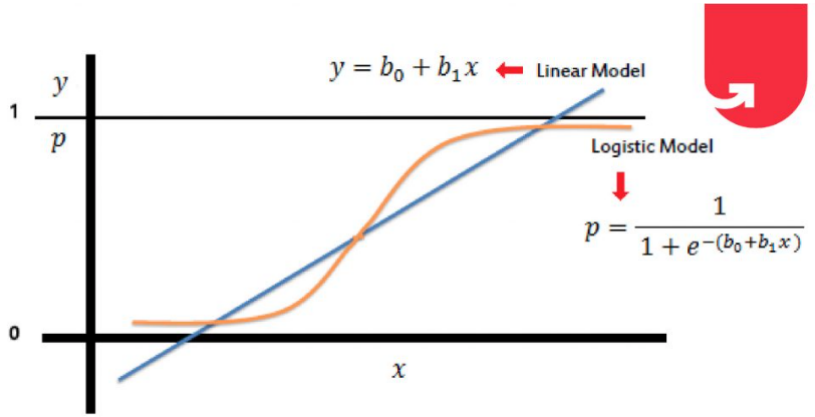
Step 4: Place b0, b1, and b2 in the estimated linear regression equation.

The estimated linear regression equation is: $\hat{y} = b0 + b1*x1 + b2*x2$

In our example, it is $\hat{y} = -6.867 + 3.148x1 - 1.656x2$

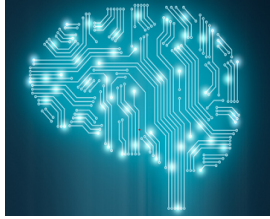
Logistic Regression

- Logistic regression is one of the types of regression analysis technique, which gets used when the dependent variable is discrete. Example: 0 or 1, true or false, etc.
- This means the target variable can have only two values, and a sigmoid curve denotes the relation between the target variable and the independent variable
- Logistic regression uses a logistic function called a sigmoid function to map predictions and their probabilities. The sigmoid function refers to an S-shaped curve that converts any real value to a range between 0 and 1.



MIT-WPU
॥ विश्वशान्तिर्धुवं ध्रुवा ॥

Logistic Regression



- If the output of the sigmoid function (estimated probability) is greater than a predefined threshold on the graph, the model predicts that the instance belongs to that class. If the estimated probability is less than the predefined threshold, the model predicts that the instance does not belong to the class

- For example, if the output of the sigmoid function is above 0.5, the output is considered as 1. On the other hand, if the output is less than 0.5, the output is classified as 0. Also, if the graph goes further to the negative end, the predicted value of y will be 0 and vice versa.

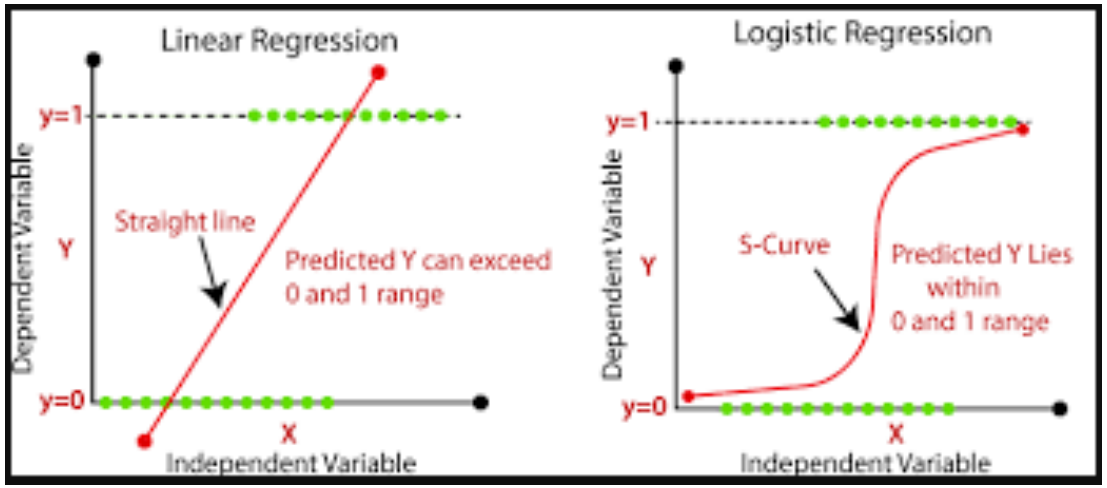
Equation represents logistic regression:

$$y = \frac{e^{(b_0 + b_1 X)}}{1 + e^{(b_0 + b_1 X)}}$$

here,

x = input value
y = predicted output
b0 = bias or intercept term
b1 = coefficient for input (x)

Linear Vs Logistic

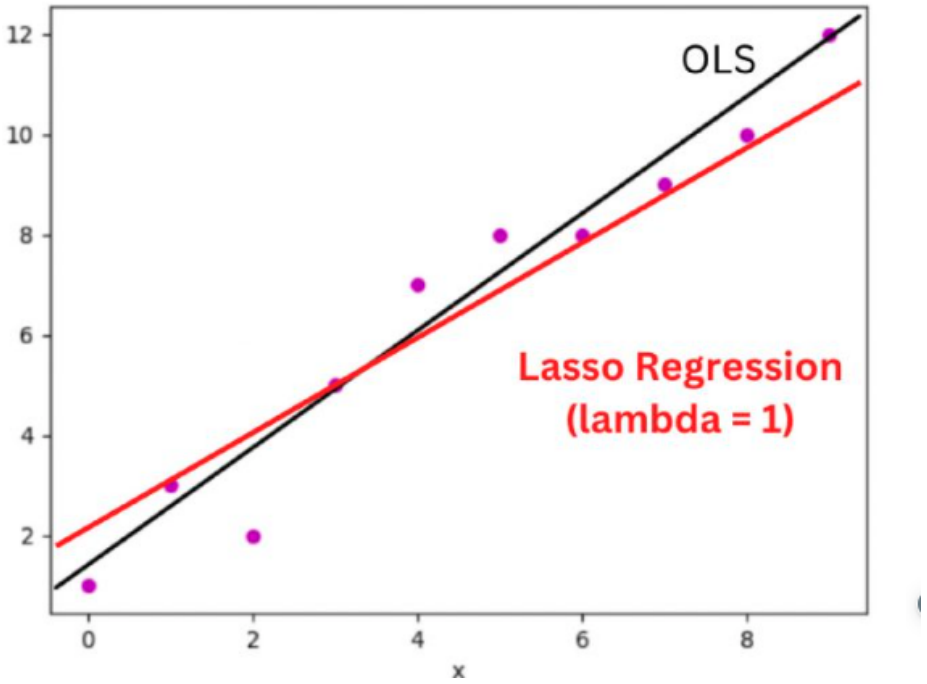


LINEAR REGRESSION	LOGISTIC REGRESSION
A linear approach that models the relationship between a dependent variable and one or more independent variables	A statistical model that predicts the probability of an outcome that can only have two values
Used to solve regression problems	Used to solve classification problems (binary classification)
Estimates the dependent variable when there is a change in the independent variable	Calculates the possibility of an event occurring
Output value is continuous	Output value is discrete
Uses a straight line	Uses an S curve or sigmoid function
Ex: predicting the GDP of a country, predicting product price, predicting the house selling price, score prediction	Ex: predicting whether an email is spam or not, predicting whether the credit card transaction is fraud or not, predicting whether a customer will take a loan or not

Lasso Regression

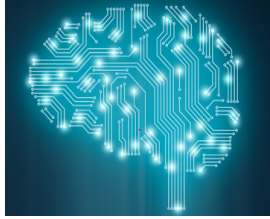
- Lasso Regression is one of the types of regression in machine learning that performs regularization along with feature selection. It prohibits the absolute size of the regression coefficient. As a result, the coefficient value gets nearer to zero, which does not happen in the case of Ridge Regression.

- Due to this, feature selection gets used in Lasso Regression, which allows selecting a set of features from the dataset to build the model.
- In the case of Lasso Regression, only the required features are used, and the other ones are made zero. This helps in avoiding the overfitting in the model. In case the independent variables are highly collinear, then Lasso regression picks only one variable and makes other variables to shrink to zero.



MIT-WPU
॥ विश्वशान्तिर्धुवं ध्रुवा ॥

Lasso Regression

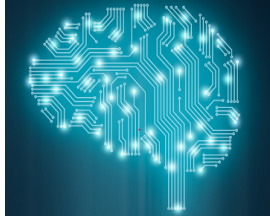


Lasso regression is a type of linear regression that uses shrinkage. Shrinkage is where data values are shrunk towards a central point, like the mean. The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters). This particular type of regression is well-suited for models showing high levels of muticollinearity or when you want to automate certain parts of model selection, like variable selection/parameter elimination.

The acronym “LASSO” stands for Least Absolute Shrinkage and Selection Operator.

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Lasso Regression



L1 Regularization

Lasso regression performs L1 regularization, which adds a penalty equal to the absolute value of the magnitude of coefficients. This type of regularization can result in sparse models with few coefficients; Some coefficients can become zero and eliminated from the model. Larger penalties result in coefficient values closer to zero, which is the ideal for producing simpler models. On the other hand, L2 regularization (e.g. Ridge regression) doesn't result in elimination of coefficients or sparse models. This makes the Lasso far easier to interpret than the Ridge.

Performing the Regression

Lasso solutions are quadratic programming problems, which are best solved with software (like Matlab). The goal of the algorithm is to minimize:

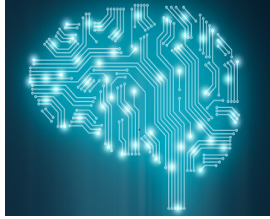
lasso regression

$$\sum_{i=1}^{n}(y_i - \sum_{j} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

Which is the same as minimizing the sum of squares with constraint $\Sigma |B_j \leq s$ (Σ = summation notation). Some of the βs are shrunk to exactly zero, resulting in a regression model that's easier to interpret

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

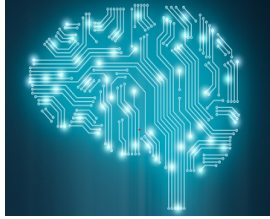
Ridge Regression



- Ridge regression is a statistical regularization technique. It corrects for overfitting on training data in machine learning models.
- Ridge regression—also known as L2 regularization—is one of several types of regularization for linear regression models.
- Regularization is a statistical method to reduce errors caused by overfitting on training data. Ridge regression specifically corrects for multicollinearity in regression analysis.
  - This is because, in the case of multi collinear data, the least square estimates give unbiased values. But, in case the collinearity is very high, there can be some bias value.
  - Therefore, a bias matrix is introduced in the equation of Ridge Regression. This is a powerful regression method where the model is less susceptible to overfitting
- This is useful when developing machine learning models that have a large number of parameters, particularly if those parameters also have high weights.

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Ridge Regression



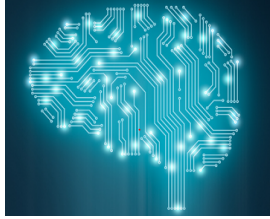
Below is the equation used to denote the Ridge Regression, where the introduction of λ (lambda) solves the problem of multicollinearity:

$$\beta = (X^{T}X + \lambda*I)^{-1}X^{T}y$$

Performs L2 regularization, i.e., adds penalty equivalent to the square of the magnitude of coefficients
Minimization objective = LS Obj + α * (sum of square of coefficients)

MIT-WPU
॥ विश्वशान्तिर्धुवं ध्रुवा ॥

Polynomial Regression



- Polynomial Regression is another one of the types of regression analysis techniques in machine learning, which is the same as Multiple Linear Regression with a little modification. In Polynomial Regression, the relationship between independent and dependent variables, that is X and Y, is denoted by the n-th degree.

- It is a linear model as an estimator. Least Mean Squared Method is used in Polynomial Regression also. The best fit line in Polynomial Regression that passes through all the data points is not a straight line, but a curved line, which depends upon the power of X or value of n.
- Polynomial Regression is a regression algorithm that models the relationship between a dependent(y) and independent variable(x) as nth degree polynomial. The Polynomial Regression equation is given below:

  $y= b0+b1x1+ b2x12+ b2x13+..... bnx1n$

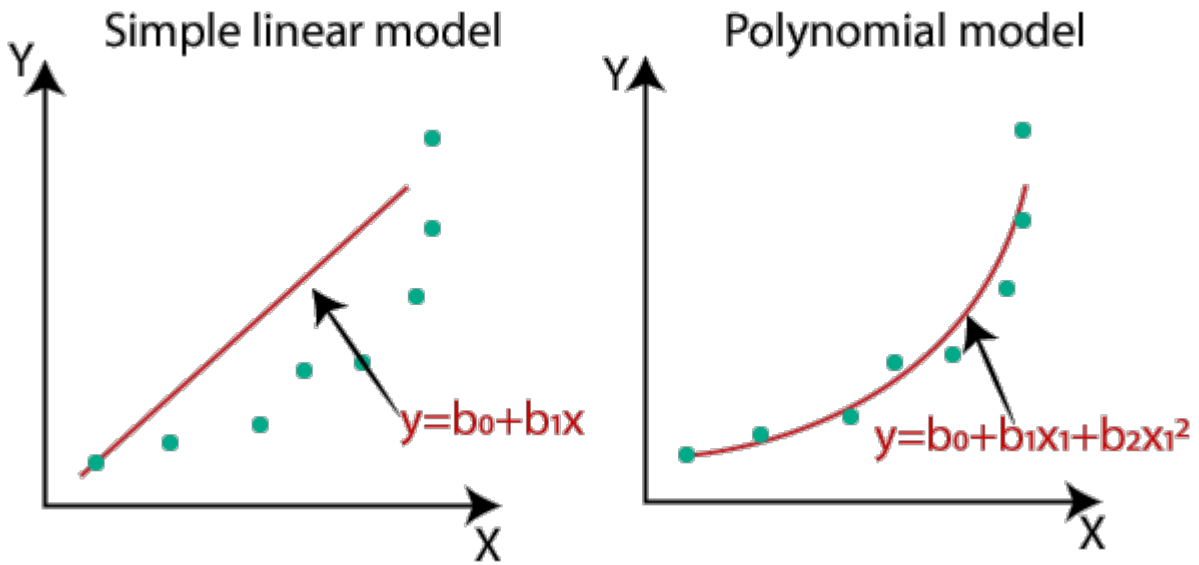
Polynomial Regression

- It is also called the special case of Multiple Linear Regression in ML. Because we add some polynomial terms to the Multiple Linear regression equation to convert it into Polynomial Regression.
- The dataset used in Polynomial regression for training is of non-linear nature.
- It makes use of a linear regression model to fit the complicated and non-linear functions and datasets.
- Hence, "In Polynomial regression, the original features are converted into Polynomial features of required degree (2,3,..,n) and then modeled using a linear model."

Need for Polynomial Regression:

- If we apply a linear model on a **linear dataset**, then it provides us a good result as we have seen in Simple Linear Regression, but if we apply the same model without any modification on a **non-linear dataset**, then it will produce a drastic output. Due to which loss function will increase, the error rate will be high, and accuracy will be decreased.
- **When data points are arranged in a non-linear fashion, we need the Polynomial Regression model**.

Polynomial Regression



In the above image, we have taken a dataset which is arranged non-linearly. So if we try to cover it with a linear model, then we can clearly see that it hardly covers any data point. On the other hand, a curve is suitable to cover most of the data points, which is of the Polynomial model.

Hence, if the datasets are arranged in a non-linear fashion, then we should use the Polynomial Regression model instead of Simple Linear Regression.

Polynomial Regression

Equation of the Polynomial Regression Model:

Simple Linear Regression equation: $y = b0+b1x$ .....(a)

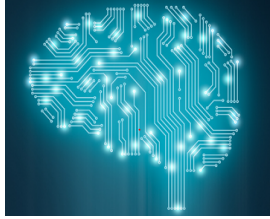
Multiple Linear Regression equation: $y= b0+b1x+ b2x2+ b3x3+....+ bnxn$ .....(b)

Polynomial Regression equation: $y= b0+b1x + b2x2+ b3x3+....+ bnxn$ .....(c)

When we compare the above three equations, we can clearly see that all three equations are Polynomial equations but differ by the degree of variables. The Simple and Multiple Linear equations are also Polynomial equations with a single degree, and the Polynomial regression equation is Linear equation with the nth degree. So if we add a degree to our linear equations, then it will be converted into Polynomial Linear equations.

MIT-WPU
॥ विश्वशान्तिर्धुवं ध्रुवा ॥

Regression Metrics



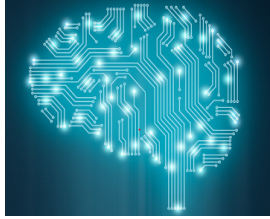
True Values and Predicted Values:
In regression, we've got two units of values to compare: the actual target values (authentic values) and the values expected by our version (anticipated values). The performance of the model is assessed by means of measuring the similarity among these sets.

Evaluation Metrics:
Regression metrics are quantitative measures used to evaluate the nice of a regression model. Scikit-analyze provides several metrics, each with its own strengths and boundaries, to assess how well a model suits the statistics.

Types of Regression Metrics
Some common regression metrics in scikit-learn with examples

- Mean Absolute Error (MAE)
- Mean Squared Error (MSE)
- R-squared (R^2) Score
- Root Mean Squared Error (RMSE)

Regression Metrics



Loss Functions for Regression

Below you will find the loss functions you can use for solving a Regression problem

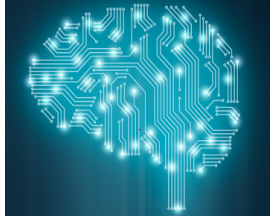
1. Mean Absolute Error (MAE)

This is also known as the L1 loss

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

This loss function is easy to compute and measures the absolute difference between the true and predicted value. It is not sensitive to outliers and it is also not differentiable at zero.

Regression Metrics



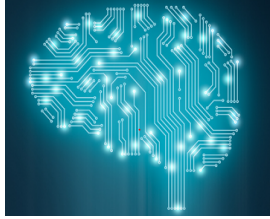
2. Mean Squared Error (MSE)

This is also known as the L2 loss

$$MSE - \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

This loss function handles outliers in an efficient manner as outliers are detected due to the quadratic loss. Convergence is also smooth as the gradient becomes smaller as the loss decreases.

Regression Metrics



3. Mean Bias Error (MBE)

This loss measures the ability of the model to over estimate or under estimate the bias

$$MBE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)$$

The above loss differs from MAE as it doesn't use the absolute value. It helps in identifying the direction of the model bias i.e (+ve or -ve). The issue with this loss is that some times the errors may cancel out between individual samples resulting in a zero loss which may be misleading.

Regression Metrics

Root Mean Squared Error

- RMSE stands for Root Mean Squared Error. It is a usually used metric in regression analysis and machine earning to measure the accuracy or goodness of fit of a predictive model, especially when the predictions are continuous numerical values.
- The RMSE quantifies how well the predicted values from a model align with the actual observed values in the dataset.
- Calculate the Squared Differences: For each data point, subtract the predicted value from the actual (observed) value, square the result, and sum up these squared differences.
- Compute the Mean: Divide the sum of squared differences by the number of data points to get the mean squared error (MSE).
- Take the Square Root: To obtain the RMSE, simply take the square root of the MSE.

Mathematical Formula

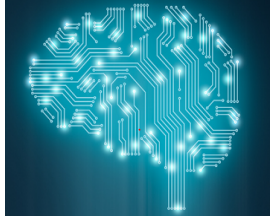
- The formula for RMSE for a data with ‘n’ data points is as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2}$$

Where: RMSE is the Root Mean Squared Error.
xi represents the actual or observed value for the i-th data point. yi represents the predicted value

MIT-WPU
॥ विश्वशान्तिर्धुवं ध्रुवा ॥

Regularization



Overfitting is one of the most serious kinds of problems related to machine learning. It occurs when a model learns the training data too well. The model then learns not only the relationships among data but also the noise in the dataset. Overfitted models tend to have good performance with the data used to fit them (the training data), but they behave poorly with unseen data (or test data, which is data not used to fit the model).

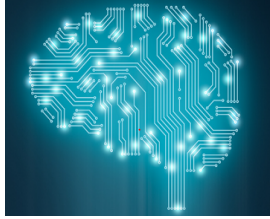
Overfitting usually occurs with complex models. **Regularization** normally tries to reduce or penalize the complexity of the model. Regularization techniques applied with logistic regression mostly tend to penalize large coefficients b_0, b_1, …, b_r:

- **L1 regularization** penalizes the LLF with the scaled sum of the absolute values of the weights: $|b_0| + |b_1| + \cdots + |b_r|$.
- **L2 regularization** penalizes the LLF with the scaled sum of the squares of the weights: $b_0^2 + b_1^2 + \cdots + b_r^2$.
- **Elastic-net regularization** is a linear combination of L1 and L2 regularization.

Regularization can significantly improve model performance on unseen data

MIT-WPU
॥ विश्वशान्तिर्धुवं ध्रुवा ॥

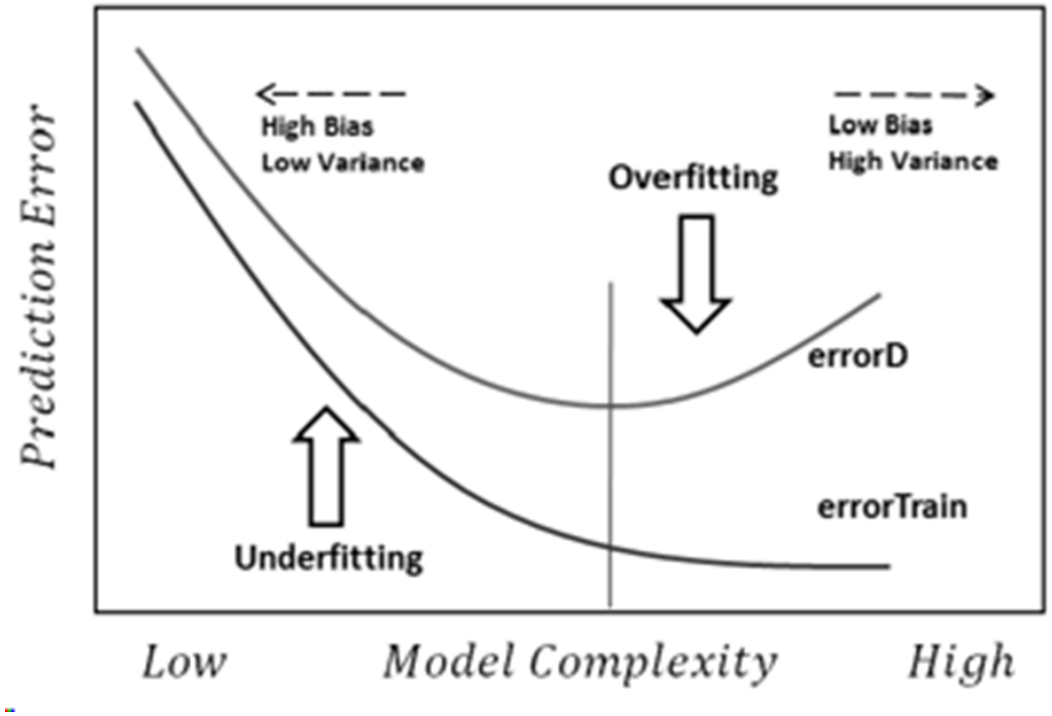
Applications of regression



- Financial forecasting (like house price estimates, or stock prices)
- Sales and promotions forecasting
- Testing automobiles
- Weather analysis and prediction
- Time series forecasting

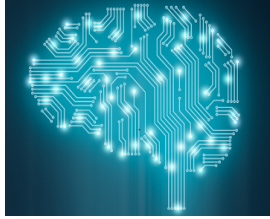
Overfitting

- Natural end of process in DT is 100% purity in each leaf
- This overfits the data, which end up fitting noise in the data
- Overfitting leads to low predictive accuracy of new data
- Past a certain point, the error rate for the validation data starts to increase



MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Overfitting & Underfitting

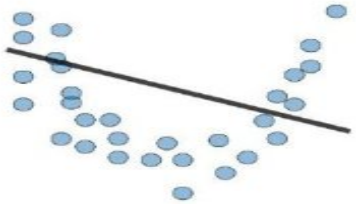
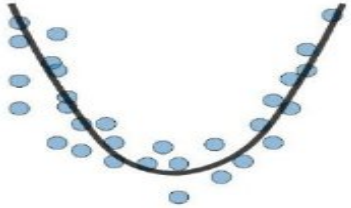
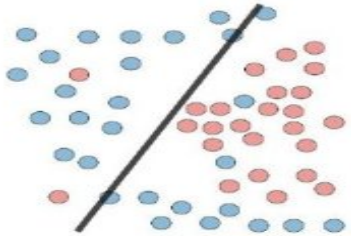
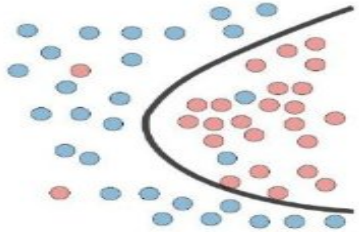
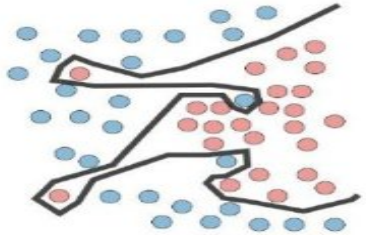
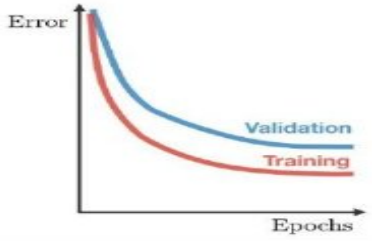
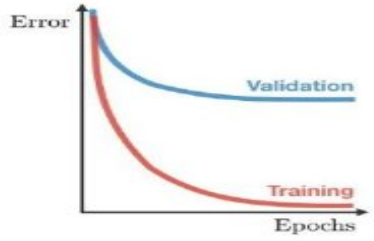


Training a Model:

- Overfit: if the model is performing much better on train set but not performing well on cross-validation set
- Overfit is called as “High Variance” problem
- Underfit: if the model is not performing well on train set itself
- Underfit is considered as “High Bias” problem

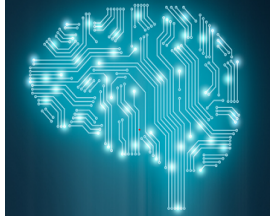
MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Overfitting & Underfitting

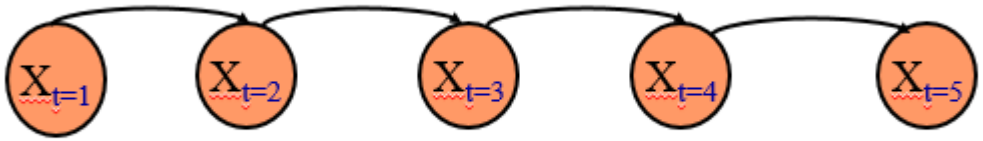
	Underfitting	Just right	Overfitting
Symptoms	- High training error - Training error close to test error - High bias	- Training error slightly lower than test error	- Low training error - Training error much lower than test error - High variance
Regression			
Classification			
Deep learning			
Remedies	- Complexify model - Add more features - Train longer		- Regularize - Get more data

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Markov Models



- A Markov model is a finite state machine with N distint states begins at (Time t = 1) in initial state
- It moves from current state to Next state according to the transition probabilities associated with the Current state
- This kind of system is called Finite or Discrete Markov model.
- Markov Property : The Current state of the system depends only on the previous state of the system
- The State of the system at Time [T+1] depends on the state of the system at time T.



Discrete Markov Model : Example

A Discrete Markov Model with 5 states.

- Each aij represents the probability of moving from state ‘ i’ to state ’j’.
- The probability to start in a given state I is πi.
- The Vector π represents the start probabilities.
- To define Markov model, the following probabilities have to be specified: transition probabilities
  aij = P(Si | Sj) and initial probabilities
  πi = P(Si)

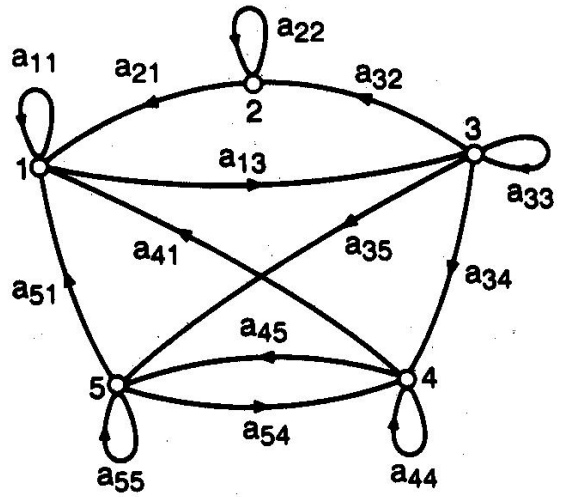
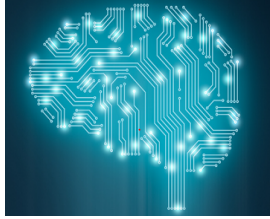
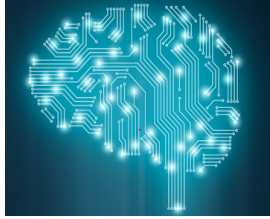


Figure 6.1 A Markov chain with five states (labeled 1 to 5) with selected state transitions.



Hidden Markov Models

- A Hidden Markov Model (HMM) is a statistical model that represents systems with hidden states and observable events. It consists of:
  - **States**: Hidden conditions that are not directly observable.
  - **Observations**: Observable events influenced by the hidden states.
  - **Transition Probabilities**: Probabilities of moving from one state to another.
  - **Emission Probabilities**: Probabilities of observing a particular event given a state.
  - **Initial State Probabilities**: Probabilities of the system starting in a particular state.



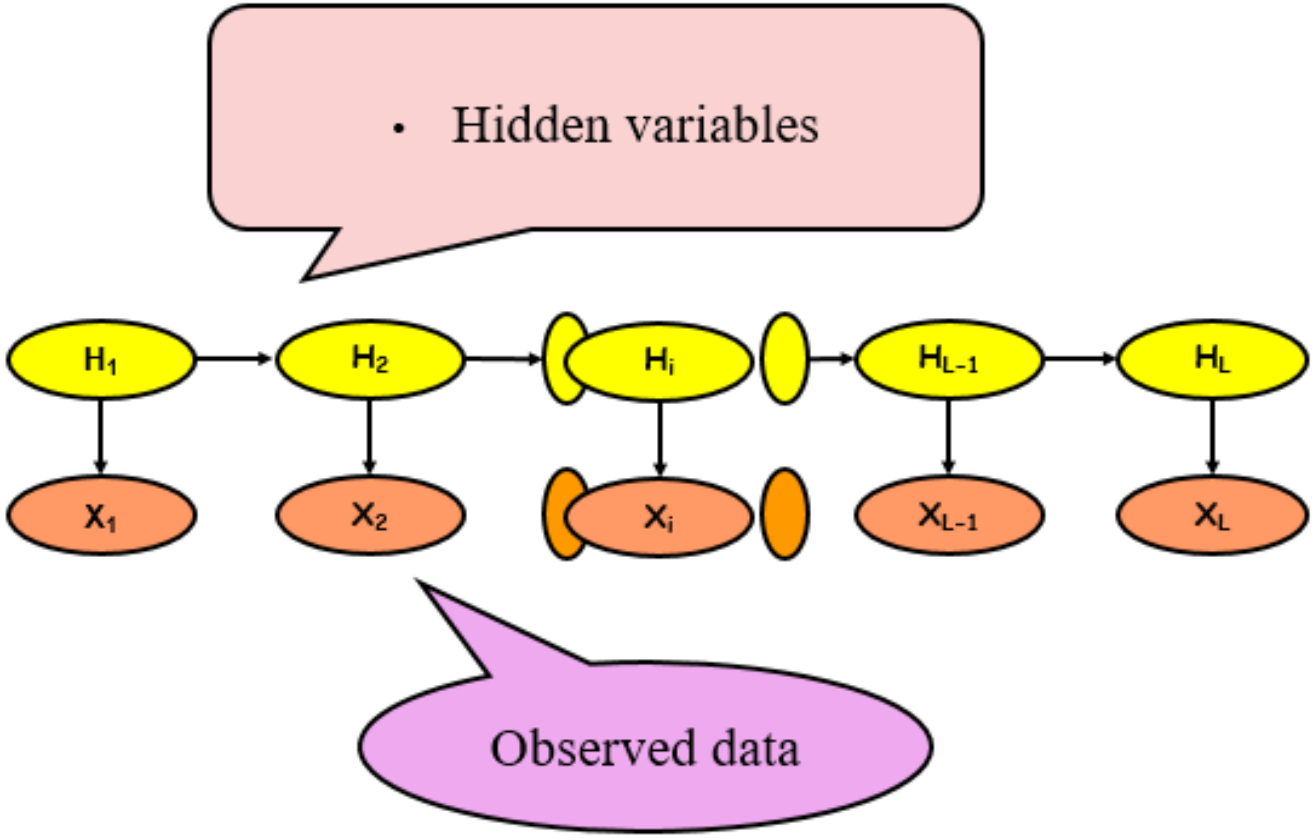
Understanding HMM Parameters

- To fully understand the Viterbi algorithm, it is essential to grasp the parameters of an HMM:

1. **States (S):** The set of hidden states in the model, denoted as $S = s_1, s_2, \ldots, s_N$.
2. **Observations (O):** The sequence of observed events, denoted as $O = o_1, o_2, \ldots, o_T$.
3. **Transition Probabilities (A):** The probability of transitioning from one state to another, denoted as $A = a_{ij}$ where $a_{ij} = P(s_j \mid s_i)$.
4. **Emission Probabilities (B):** The probability of observing a particular event from a state, denoted as $B = b_j(o_t)$ where $b_j(o_t) = P(o_t \mid s_j)$.
5. Initial Probabilities (π): The probability of starting in a particular state, denoted as $\pi = \{\pi_i\}$where $\pi_i = P(s_i \text{ at } t = 1)$.

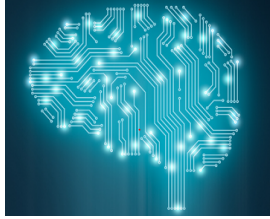
MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Hidden Markov Models



MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Hidden Markov Model in Machine Learning



- **The hidden Markov Model** (HMM) is a statistical model that is used to describe the probabilistic relationship between a sequence of observations and a sequence of hidden states. It is often used in situations where the underlying system or process that generates the observations is unknown or hidden, hence it has the name "Hidden Markov Model."

- It is used to predict future observations or classify sequences, based on the underlying hidden process that generates the data.

- An HMM consists of two types of variables: hidden states and observations.

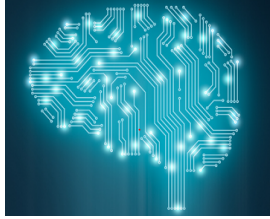
  -The **hidden states** are the underlying variables that generate the observed data, but they are not directly observable.

  -The **observations** are the variables that are measured and observed.

-

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Hidden Markov Model in Machine Learning



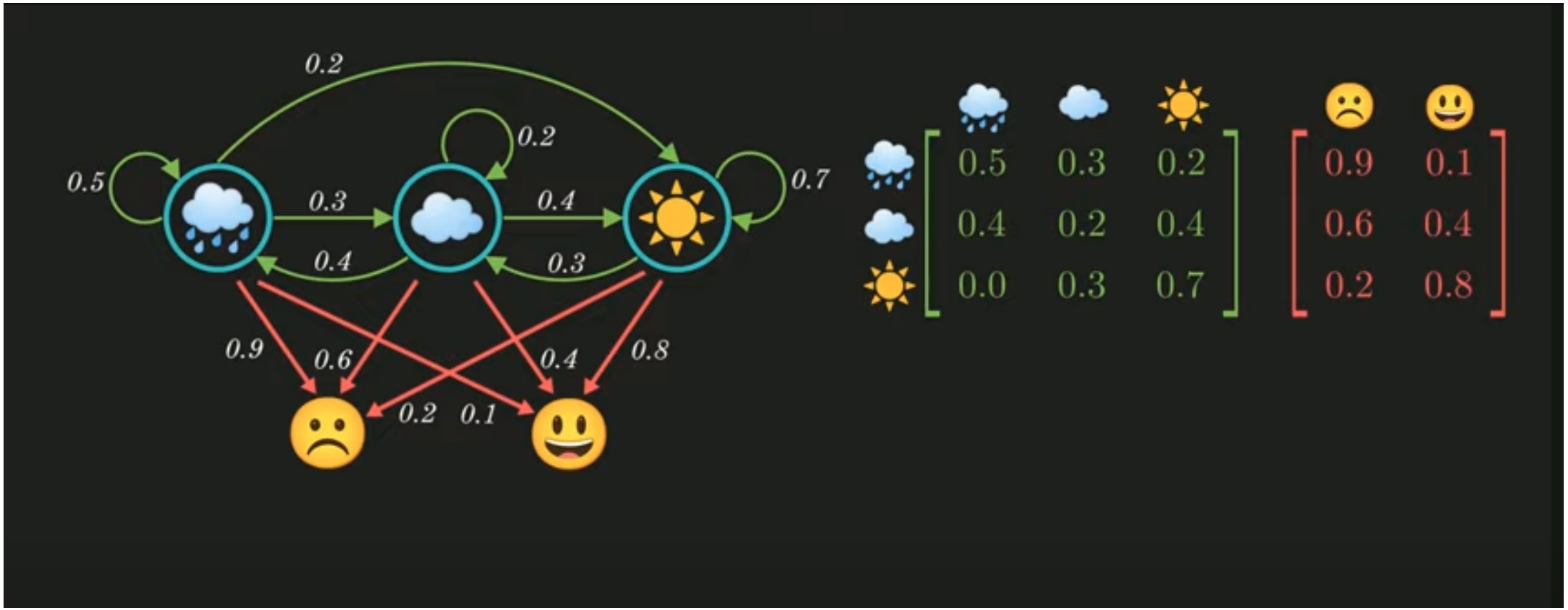
- The relationship between the hidden states and the observations is modeled using a probability distribution. The Hidden Markov Model (HMM) is the relationship between the hidden states and the observations using two sets of probabilities: the transition probabilities and the emission probabilities.

-The **transition probabilities** describe the probability of transitioning from one hidden state to another.

-The **emission probabilities** describe the probability of observing an output given a hidden state.

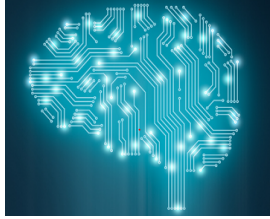
MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Hidden Markov Model in Machine Learning Example



MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Hidden Markov Model Algorithm



The Hidden Markov Model (HMM) algorithm can be implemented using the following steps:

Step 1: Define the state space and observation space

The state space is the set of all possible hidden states, and the observation space is the set of all possible observations.

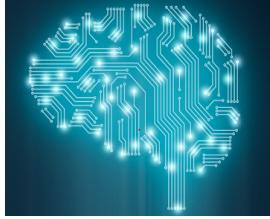
Step 2: Define the initial state distribution

This is the probability distribution over the initial state.

Step 3: Define the state transition probabilities

These are the probabilities of transitioning from one state to another. This forms the transition matrix, which describes the probability of moving from one state to another.

Hidden Markov Model Algorithm



Step 4: Define the observation likelihoods:

These are the probabilities of generating each observation from each state. This forms the emission matrix, which describes the probability of generating each observation from each state.

Step 5: Train the model

The parameters of the state transition probabilities and the observation likelihoods are estimated using the Baum-Welch algorithm, or the forward-backward algorithm. This is done by iteratively updating the parameters until convergence.

Step 6: Decode the most likely sequence of hidden states

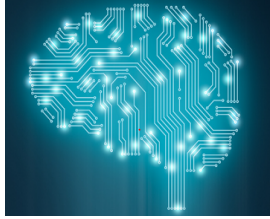
Given the observed data, the Viterbi algorithm is used to compute the most likely sequence of hidden states. This can be used to predict future observations, classify sequences, or detect patterns in sequential data.

Step 7: Evaluate the model

The performance of the HMM can be evaluated using various metrics, such as accuracy, precision, recall, or F1 score

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Three Basic Problems of HMM



Evaluation Problem (Forward-backward Algorithm)

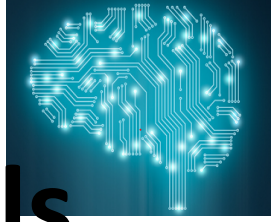
— Given the Hidden Markov Model $\lambda = (A, B, \pi)$ and a sequence of observations O, find the probability of an observation $P(O \mid \lambda)$ known as Evaluation Problem.

Decoding Problem (Viterbi Algorithm)

— Given the Hidden Markov Model $\lambda = (A, B, \pi)$ and an observation sequence O, find the most likely state sequence (s1, s2… sn) known as Decoding Problem.

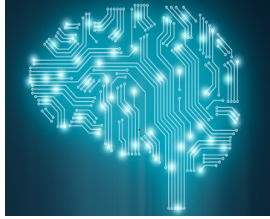
Learning or Training Problem (EM Algorithm)

— Given an observation sequence o_1 o_2 …… oT, update (A, B, π) so that the probability of the evaluation problem is as large as possible known as a Learning or Training Problem.



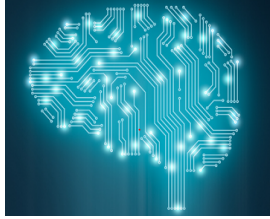
Viterbi Algorithm for Hidden Markov Models

- The Viterbi algorithm is a dynamic programming algorithm for finding the most likely sequence of hidden states in a Hidden Markov Model (HMM).
- The **Viterbi algorithm** is a fundamental dynamic programming technique widely used in the context of **Hidden Markov Models (HMMs)** to uncover the most likely sequence of hidden states given a sequence of observed events.
- It is widely used in various applications such as speech recognition, bioinformatics, and natural language processing.



Contd..

- The Viterbi algorithm operates by iteratively computing the highest probability paths to each state at each time step, storing these probabilities, and backtracking to determine the most probable sequence of hidden states.
- This method ensures an efficient and accurate decoding of hidden state sequences, making it indispensable for applications that require precise sequence analysis and pattern recognition.



Initialization in Viterbi Algorithm

- Initialization is the first step of the Viterbi algorithm. It sets up the initial probabilities for the starting states based on the initial state probabilities and the emission probabilities for the first observation.

Mathematically it can be represented as:

$$V_1(j) = \pi_j . b_j(o_1) \forall j \epsilon \{1, \ldots, N\}$$

$$\text{Path}_j(1) = [j] \forall j \epsilon \{1, \ldots, N\}$$

The Forward Algorithm

- The Forward Algorithm is used to compute the probability of observing a sequence of observations given an HMM. It is a dynamic programming algorithm that recursively calculates the probabilities of partial observation sequences

Step 1: Initialization

$$\alpha_1(j) = \pi_j.b_j(o_1)$$

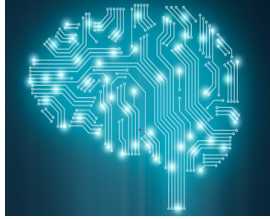
Step 2: Recursion

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i).a_{ij}.b_j(o_t)$$

Step 3: Termination

$$P(O \mid \lambda) = \sum_{j=1}^{N} \alpha_T(j)$$

The Backward Algorithm



- The Backward Algorithm complements the Forward Algorithm by computing the probability of the ending part of the observation sequence, starting from a given state.

Step 1: Initialization

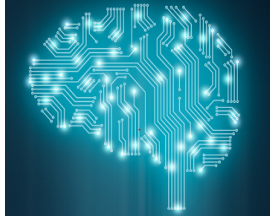
$$\beta_T(j) = 1$$

Step 2: Recursion

$$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j).a_{ij}.b_j(o_{t+1})$$

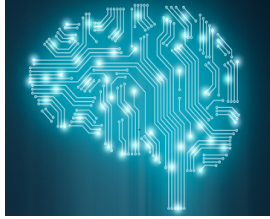
Step 3: Termination

$$P(O|\lambda) = \sum_{j=1}^{N} \pi_j.b_j(o_1).\beta_1(j)$$



Optimizing Viterbi Algorithm

- To optimize the Viterbi algorithm, consider the following:
- **Pruning**: Reducing the state space by eliminating states with very low probabilities.
- **Logarithms**: Using log probabilities to avoid underflow issues.
- **Beam Search**: Keeping track of only the top k most likely paths at each step to reduce computational complexity.



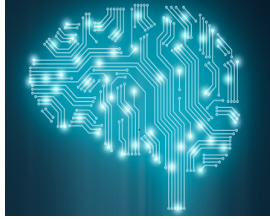
Example: Viterbi Algorithm in Action

- Consider an HMM with two states (Rainy, Sunny) and three observations (Walk, Shop, Clean). The following matrices define the model:
  - **States**: S={Rainy,Sunny}
  - **Observations**: O={Walk,Shop,Clean}
  - **Transition Probabilities (A)**: $A = \begin{pmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{pmatrix}$
  - **Emission Probabilities (B)**: $B = \begin{pmatrix} 0.1 & 0.4 & 0.5 \\ 0.6 & 0.3 & 0.1 \end{pmatrix}$
  - **Initial Probabilities** (π): $\pi = \begin{pmatrix} 0.6 & 0.4 \end{pmatrix}$

Given the observation sequence (Walk, Shop, Clean), the Viterbi algorithm computes the most probable state sequence.

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Part of Speech (POS) with Hidden Markov Model



- We have learned the differences between the various parts of speech tags such as nouns, verbs, adjectives, and adverbs.
- Associating each word in a sentence with a proper POS (part of speech) is known as POS tagging or POS annotation.
- POS tags are also known as word classes, morphological classes, or lexical tags.
- POS tags give a large amount of information about a word and its neighbors. Their applications can be found in various tasks such as information retrieval, parsing, Text to Speech (TTS) applications, information extraction, linguistic research for corpora.
- They are also used as an intermediate step for higher-level NLP tasks such as parsing, semantics analysis, translation, and many more, which makes POS tagging a necessary function for advanced NLP applications.
- HMM (Hidden Markov Model) is a Stochastic technique for POS tagging. Hidden Markov models are known for their applications to reinforcement learning and temporal pattern recognition such as speech, handwriting, gesture recognition, musical score following, partial discharges, and bioinformatics.

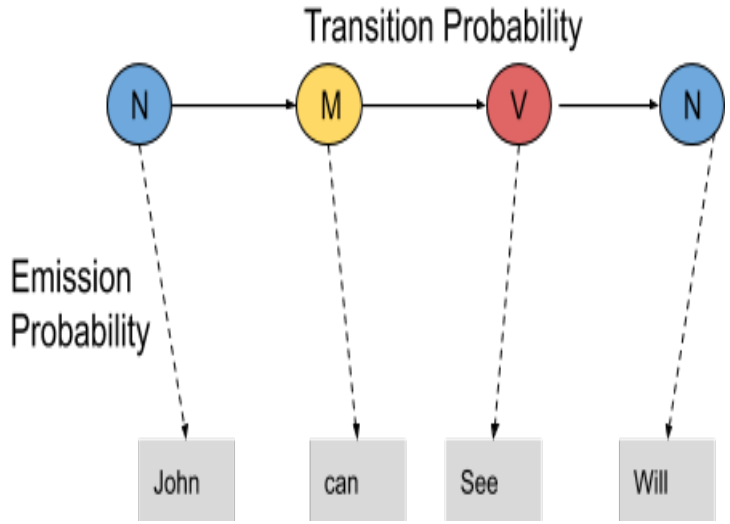
POS tagging with Hidden Markov Model

- Let us consider an example proposed by Dr.Luis Serrano and find out how HMM selects an appropriate tag sequence for a sentence.

  - In this example, we consider only 3 POS tags that are noun, model and verb. Let the sentence “ Ted will spot Will ” be tagged as noun, model, verb and a noun and to calculate the probability associated with this particular sequence of tags we require their Transition probability and Emission probability.

  - The transition probability is the likelihood of a particular sequence for example, how likely is that a noun is followed by a model and a model by a verb and a verb by a noun. This probability is known as Transition probability. It should be high for a particular sequence to be correct.

  - Now, what is the probability that the word Ted is a noun, will is a model, spot is a verb and Will is a noun. These sets of probabilities are Emission probabilities and should be high for our tagging to be likely.



POS tagging with Hidden Markov Model

Let us calculate the above two probabilities for the set of sentences below

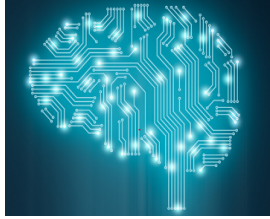
1. Mary Jane can see Will
2. Spot will see Mary
3. Will Jane spot Mary?
4. Mary will pat Spot

Note that Mary Jane, Spot, and Will are all names.

- In the above sentences, the word Mary appears four times as a noun. To calculate the emission probabilities, let us create a counting table in a similar manner.

Words	Noun	Model	Verb
Mary	4	0	0
Jane	2	0	0
Will	1	3	0
Spot	2	0	1
Can	0	1	0
See	0	0	2
pat	0	0	1

POS tagging with Hidden Markov Model



Now let us divide each column by the total number of their appearances for example, 'noun' appears nine times in the above sentences so divide each term by 9 in the noun column. We get the following table after this operation.

From the table, we infer that

The probability that Mary is Noun = 4/9

The probability that Mary is Model = 0

The probability that Will is Noun = 1/9

The probability that Will is Model = 3/4

Words	Noun	Model	Verb
Mary	4/9	0	0
Jane	2/9	0	0
Will	1/9	3/4	0
Spot	2/9	0	1/4
Can	0	1/4	0
See	0	0	2/4
pat	0	0	1

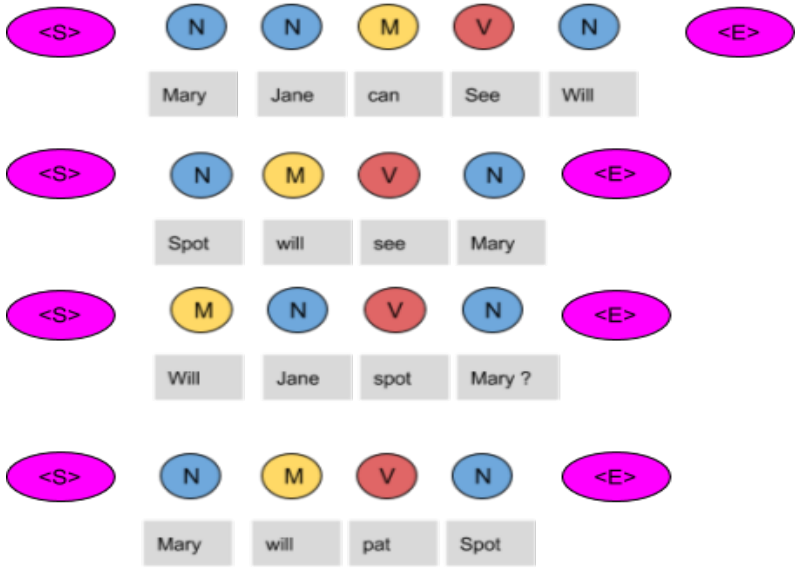
In a similar manner, you can figure out the rest of the probabilities. These are the emission probabilities.

POS tagging with Hidden Markov Model

Next, we have to calculate the transition probabilities, so define two more tags <S> and <E>. <S> is placed at the beginning of each sentence and <E> at the end as shown in the figure below.

Let us again create a table and fill it with the co-occurrence counts of the tags.

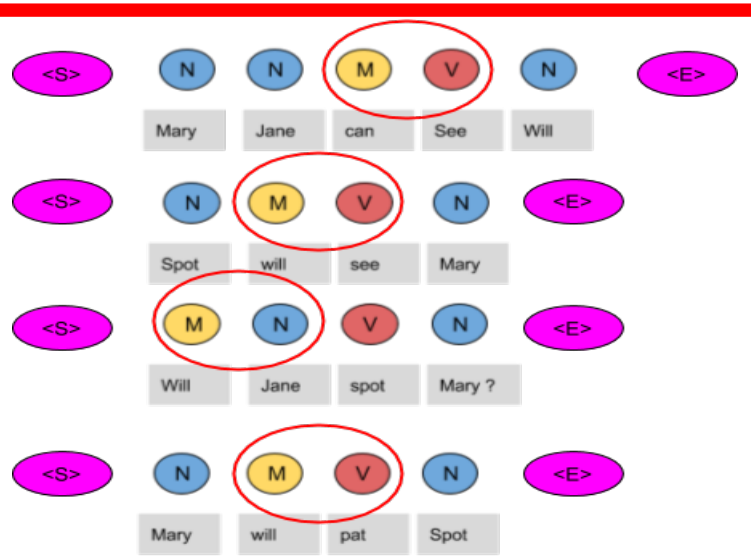
	N	M	V	<E>
<S>	3	1	0	0
N	1	3	1	4
M	1	0	3	0
V	4	0	0	0



In the above figure, we can see that the <S> tag is followed by the N tag three times, thus the first entry is 3.The model tag follows the <S> just once, thus the second entry is 1. In a similar manner, the rest of the table is filled.

POS tagging with Hidden Markov Model

Next, we divide each term in a row of the table by the total number of co-occurrences of the tag in consideration, for example, The Model tag is followed by any other tag four times as shown below, thus we divide each element in the third row by four.

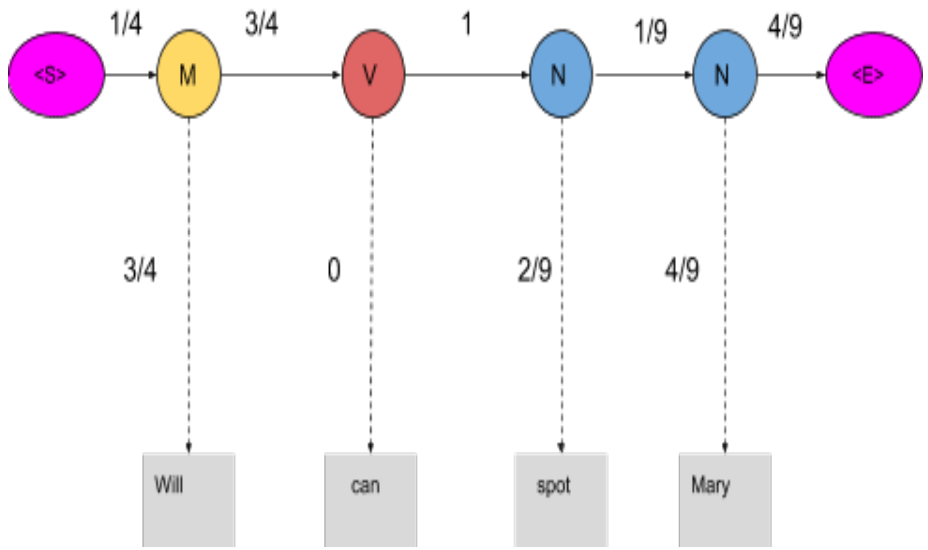


	N	M	V	<E>
<S>	3/4	1/4	0	0
N	1/9	3/9	1/9	4/9
M	1/4	0	3/4	0
V	4/4	0	0	0

These are the respective transition probabilities for the above four sentences. Now how does the HMM determine the appropriate sequence of tags for a particular sentence from the above tables? Let us find it out.

POS tagging with Hidden Markov Model

Take a new sentence and tag them with wrong tags. Let the sentence, ‘ Will can spot Mary’ be tagged as-



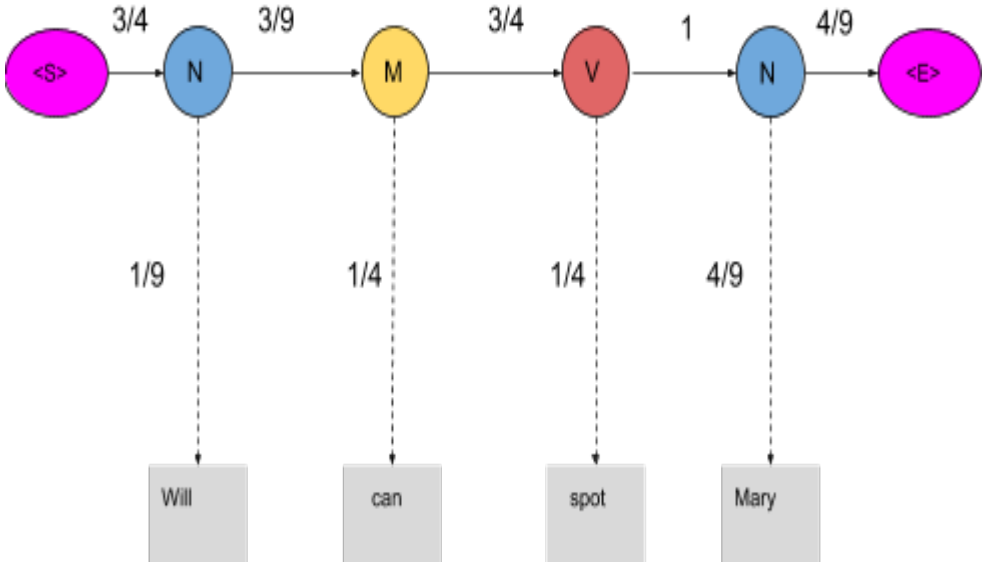
Will as a model
Can as a verb
Spot as a noun
Mary as a noun
Now calculate the probability of this sequence being correct in the following manner.

The probability of the tag Model (M) comes after the tag <S> is ¼ as seen in the table. Also, the probability that the word Will is a Model is 3/4. In the same manner, we calculate each and every probability in the graph. Now the product of these probabilities is the likelihood that this sequence is right. Since the tags are not correct, the product is zero.

1/4*3/4*3/4*0*1*2/9*1/9*4/9*4/9=0

POS tagging with Hidden Markov Model

When these words are correctly tagged, we get a probability greater than zero as shown below



Calculating the product of these terms we get,

3/4*1/9*3/9*1/4*3/4*1/4*1*4/9*4/9=0.00025720164