Unit-2

Maximum Likelihood Estimation

Estimation is the process of making an educated guess about a population parameter based on a sample from that population. In ML we use a model to represent real-world phenomena. These models are based on parameters that influence our prediction because the reliability of predictions would only be good as the parameters that govern the model.

Maximum Likelihood Estimation (MLE) is a statistical method used to estimate the parameters of a model by finding the values that maximize the likelihood function. The idea is to find the values of the model parameters that maximize the likelihood of observed data such that observed data is appropriate. In simple terms, MLE helps us identify the parameter values that make the observed data most probable, given a particular model. Likelihood is a measure of data observations up to which it can tell us the results or the target variables value for particular data points.

The likelihood function expresses the likelihood of parameter values occurring given the observed data. It assumes that the parameters are unknown.

This implies that in order to implement maximum likelihood estimation we must:

1. Assume a model, also known as a data generating process, for our data.
2. Be able to derive the likelihood function for our data, given our assumed model.

Let $X_1, X_2, X_3, \ldots\ldots X_n$ be a random sample from any distribution with the parameter theta, let $x_1, x_2, x_3, \ldots\ldots, x_n$ are the observed values of $X_1, X_2, X_3, \ldots\ldots X_n$ ($X_1 = x_1, X_2 = x_2, \ldots\ldots X_n = x_n$), then a maximum likelihood estimate of theta is a value that will maximize the likelihood function (L(X; theta)).

MLE parameter theta is a random variable that is given by:

$$\hat{\Theta}_{ML} = \hat{\Theta}_{ML}(X_1, X_2, \ldots\ldots, X_n).$$

$$\text{The value of } \hat{\Theta}_{ML} \text{ is given by } \hat{\theta}_{ML}.$$

Key Concepts:

1. Likelihood Function: This function represents the probability of observing the data as a function of the model parameters. It's denoted as:

   $$L(\theta|X)=P(X|\theta)$$

   where X is the observed data, and θ represents the parameters of the model.

2. Log-Likelihood: Often, the logarithm of the likelihood function is maximized instead of the likelihood itself, because it simplifies calculations. The log-likelihood function is:

$$\ell(\theta|X)=\log L((\theta|X)$$

3. Maximizing the Likelihood: The goal of MLE is to find the parameter θ that maximizes the likelihood function (or equivalently, the log-likelihood function). This is typically done by setting the derivative of the log-likelihood function with respect to the parameters equal to zero and solving for θ.

Steps in MLE:

1. Specify the Model: Identify the probability distribution of the data and its parameters.
2. Construct the Likelihood Function: Write down the likelihood function based on the chosen probability distribution and observed data.
3. Maximize the Likelihood: Differentiate the log-likelihood function with respect to the model parameters and solve the resulting equations for the maximum.
4. Parameter Estimates: The values of the parameters that maximize the likelihood function are considered the MLE of those parameters.

MLE in the Context of Machine Learning

1. **Modeling the Data**: In machine learning, we assume that the data is generated according to some probability distribution or process. For example, in logistic regression, we model the probability of a binary class label, and in a Gaussian mixture model (GMM), we assume the data is generated from a mixture of several Gaussian distributions.
2. **Likelihood in Machine Learning**: The likelihood function in machine learning measures how likely it is to observe the data given the parameters of the model. The objective of MLE is to find the parameters that maximize this likelihood. In supervised learning, we seek to maximize the probability of the observed outputs (labels) given the inputs and the model's parameters.

Least Squares

Least Square method is a fundamental mathematical technique widely used in data analysis, statistics, and regression modeling to identify the best-fitting curve or line for a given set of data points. This method ensures that the overall error is reduced, providing a highly accurate model for predicting future data trends.

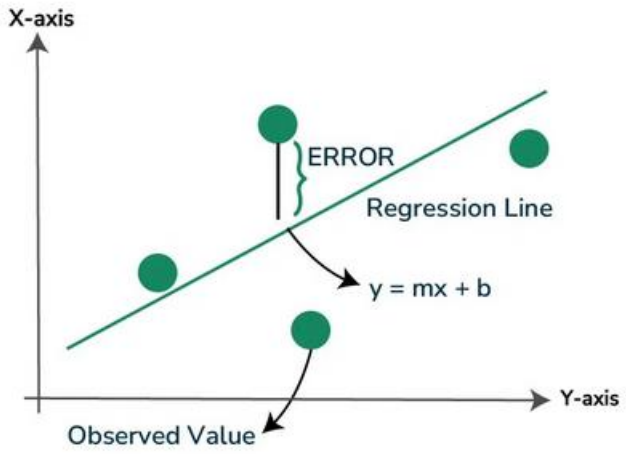
In statistics, when the data can be represented on a cartesian plane by using the independent and dependent variable as the x and y coordinates, it is called scatter data. This data might not be useful in making interpretations or predicting the values of the dependent variable for the independent variable. So, we try to get an equation of a line that fits best to the given data points with the help of the Least Square Method.

Least Square Method is used to derive a generalized linear equation between two variables. when the value of the dependent and independent variable is represented as the x and y coordinates in a 2D cartesian coordinate system. Initially, known values are marked on a plot. The plot obtained at this point is called a scatter plot.

Then, we try to represent all the marked points as a straight line or a linear equation. The equation of such a line is obtained with the help of the Least Square method. This is done to get the value of the dependent variable for an independent variable for which the value was initially unknown. This helps us to make predictions for the value of dependent variable.

Least Squares method is a statistical technique used to find the equation of best-fitting curve or line to a set of data points by minimizing the sum of the squared differences between the observed values and the values predicted by the model.

This method aims at minimizing the sum of squares of deviations as much as possible. The line obtained from such a method is called a regression line or line of best fit.



Least Square Method formula is used to find the best-fitting line through a set of data points. For a simple linear regression, which is a line of the form $y=mx+c$, where y is the dependent variable, x is the independent variable, a is the slope of the line, and b is the y-intercept, the formulas to calculate the slope (m) and intercept (c) of the line are derived from the following equations:

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

$$c = Y - mX$$

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.

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

Solution:

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**

The slope of the line of best fit can be calculated from the formula as follows:

$m = (\Sigma (X - x_i)*(Y - y_i)) / \Sigma(X - x_i)^2$

$m = 55/32.8 = 1.68$ (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, $y = 1.68x + 0.94$.

Limitations of the Least Square Method

The Least Square method assumes that the data is evenly distributed and doesn't contain any outliers for deriving a line of best fit. But, this method doesn't provide accurate results for unevenly distributed data or for data containing outliers.

Robust Linear Expression

Simple linear regression aims to find the best fit line that describes the linear relationship between some input variables (denoted by X) and the target variable (denoted by y). This has some limitations as in real-world problems, there is a high probability that the dataset may have outliers. This results in biased model fitting. To overcome this limitation of the biased fitted model, robust regression was introduced.

A robust linear expression refers to a linear model that is resilient to outliers, noise, or variations in the data. In statistical modeling or machine learning, linear models are often sensitive to these disturbances, which can lead to inaccurate predictions. Robust methods seek to minimize this sensitivity and provide more reliable estimates.

Key techniques for robust linear modeling include:

1. **Robust Regression**: Traditional linear regression minimizes the sum of squared errors (ordinary least squares, OLS). However, OLS is sensitive to outliers because large deviations are squared, amplifying their influence. Robust regression techniques, such as **Huber regression** or **Least Absolute Deviations (LAD)**, use alternative loss functions that reduce the impact of outliers.
2. **Regularization (Ridge/Lasso)**: Adding a penalty to the model's coefficients (L2 for Ridge, L1 for Lasso) can reduce sensitivity to noise by preventing the model from overfitting the data.
3. **Quantile Regression**: Instead of predicting the mean of the dependent variable, quantile regression predicts different quantiles, offering a more complete picture of the potential outcomes.
4. **M-Estimators**: These are generalizations of maximum likelihood estimators that are designed to be less sensitive to outliers. Instead of the quadratic loss used in OLS, they apply other loss functions that down-weight large residuals.

Ridge Regression

Regularization is one of the most important concepts of machine learning. It is a technique to prevent the model from overfitting by adding extra information to it.

Sometimes the machine learning model performs well with the training data but does not perform well with the test data. It means the model is not able to predict the output when deals with unseen data by introducing noise in the output, and hence the model is called overfitted. This problem can be deal with the help of a regularization technique.

This technique can be used in such a way that it will allow to maintain all variables or features in the model by reducing the magnitude of the variables. Hence, it maintains accuracy as well as a generalization of the model.

It mainly regularizes or reduces the coefficient of features toward zero. In simple words, "*In regularization technique, we reduce the magnitude of the features by keeping the same number of features.*"

There are mainly two types of regularization techniques-

- Ridge Regression
- Lasso Regression

Ridge Regression

- Ridge regression is one of the types of linear regression in which a small amount of bias is introduced so that we can get better long-term predictions.
- Multicollinearity occurs when two or more predictor variables in a regression model are highly correlated, leading to unreliable and unstable estimates of regression coefficients. Ridge regression is a procedure for eliminating the bias of coefficients and reducing the mean square error by shrinking the coefficients of a model towards zero to solve problems of overfitting or multicollinearity that are normally associated with ordinary least squares regression.
- It adds an additional term to the OLS loss function that pulls the estimating coefficients toward zero. **This is done by adding a penalty term to the log likelihood, where this penalty term is governed by a parameter denoted as lambda (λ), thus lowering the variance of the model and increasing its stability as well as the robustness of the prediction made by the model.**
- Ridge regression is particularly useful in cases where predictor variables are correlated or if the number of predictor variables is greater than the number of observing factors, as it yields more stable and reliable results.

Cost Function in Ridge Regression

The objective of ridge regression is to minimize the following cost function:

$$J(\beta) = \sum_{i=1}^{n} (y_i - X_i\beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

Where:

- J(β) is the cost function to minimize.
- yiare the actual target values.
- Xiare the input features.
- β are the regression coefficients to be learned.
- λ is the regularization parameter that controls the trade-off between minimizing the error and penalizing large coefficients.
- The first term, ∑i=1n(yi−Xiβ)2 is the residual sum of squares (RSS), which is the same as in ordinary linear regression.
- The second term, λ∑j=1pβj2 is the penalty on the size of the coefficients.

$$Cost\ Function\ = \frac{1}{n}\sum_{i=1}^{n}\left(h_\theta(x)^i - y^i\right)^2 + \lambda \sum_{i=1}^{n} |slope|$$

$$\lambda = Hyperparameter$$

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2.$$

Understanding Multicollinearity:

- **Multicollinearity** occurs when two or more independent variables (features) in a regression model are highly correlated, meaning they convey similar information.
- In the presence of multicollinearity, ordinary least squares (OLS) estimates become **unstable** and **sensitive** to small changes in the data. This can lead to:
    - Large, highly variable coefficient estimates.
    - Inflated standard errors, making it difficult to determine the true relationship between the variables and the outcome.
    - Poor generalization to new data, as the model is overfitting to the specific noise in the training data.

Ridge regression deals with **multicollinearity** by shrinking the regression coefficients through **ℓ2 regularization**, which stabilizes the model and makes it more robust.

Bayesian Linear Regression

Linear regression is a popular regression approach in machine learning. Linear regression is based on the assumption that the underlying data is normally distributed and that all relevant predictor variables have a linear relationship with the outcome. But In the real world, this is not always possible, it will follows these assumptions, Bayesian regression could be the better choice.
Bayesian regression employs prior belief or knowledge about the data to "learn" more about it and create more accurate predictions. It also takes into account the data's uncertainty and leverages prior knowledge to provide more precise estimates of the data. As a result, it is an ideal choice when the data is complex or ambiguous.

Bayesian regression uses a Bayes algorithm to estimate the parameters of a linear regression model from data, including prior knowledge about the parameters. Because of its probabilistic character, it can produce more accurate estimates for regression parameters than ordinary least squares (OLS) linear regression, provide a measure of uncertainty in the estimation, and make stronger conclusions than OLS. Bayesian regression can also be utilized for related regression analysis tasks like model selection and outlier detection.

The main difference between traditional linear regression and Bayesian regression is the underlying assumption regarding the data-generating process. Traditional linear regression assumes that data follows a Gaussian or normal distribution, while Bayesian regression has stronger assumptions about the nature of the data and puts a prior probability distribution on the parameters. Bayesian regression also enables more flexibility as it allows for additional parameters or prior distributions, and can be used to construct an arbitrarily complex model that explicitly expresses prior beliefs about the data. Additionally, Bayesian regression provides more accurate predictive measures from fewer data points and is able to construct estimates for uncertainty around the estimates. On the other hand, traditional linear regressions are easier to implement and generally faster with simpler models and can provide good results when the assumptions about the data are valid.

Bayesian Regression can be very useful when we have insufficient data in the dataset or the data is poorly distributed. The output of a Bayesian Regression model is obtained from a probability distribution, as compared to regular regression techniques where the output is just obtained from a single value of each attribute.

Bayesian regression is a type of linear regression that uses Bayesian statistics to estimate the unknown parameters of a model. It uses Bayes' theorem to estimate the likelihood of a set of parameters given observed data. The goal of Bayesian regression is to find the best estimate of the parameters of a linear model that describes the relationship between the independent and the dependent variables.

Some Dependent Concepts for Bayesian Regression

Bayes's Principle

The Bayes Theorem provides a link between an event's prior chance and its subsequent chance once all available information has been considered.

Estimation of the Maximum Likelihood (MLE)

It looks for the parameter values that provide the observed data with the best chance of fitting the presumptive model. MLE gives point estimates of the parameters and does not take into account any prior knowledge or assumptions about them.

Maximum A Posteriori (MAP) Estimation

A Bayesian method known as MAP estimation uses the likelihood function and prior knowledge to estimate the parameters. In MAP estimating, the parameters are given a prior distribution representing prior assumptions or information about their values.

Need for Bayesian Regression

- The previous opinion of the analysis's parameter assumptions is also used in Bayesian Regression. It makes it practical when there is a need for more data, and prior knowledge is essential. Bayesian Regression offers better-informed and more precise estimations of the regression parameters by fusing prior information with the observed data.
- Bayesian Regression provides a natural way to scale the uncertainty in estimating regression parameters because it generates the posterior distribution, which represents the uncertainty in the parameter values, as opposed to the single component estimate generated by conventional regression techniques. It is possible to calculate reliable or Bayesian confidence intervals using this distribution since it provides a range of acceptable parameter values.
- It makes it possible to model relationships between the predictors and the response variable that are more complex and realistic.
- By computing the posterior probabilities of several models, Bayesian Regression makes it easier to choose and compare models.
- Unlike traditional regression techniques, Bayesian Regression handles outliers and significant findings more effectively.

Key concepts of Bayesian Regression

1. Prior Distribution:

- The prior represents our initial belief about the parameters of the model before observing any data.
- In Bayesian regression, we assign a probability distribution to the model parameters (e.g., regression coefficients) to represent our uncertainty about them.
- Common choices for the prior distribution in linear regression are Gaussian (normal) distributions for the regression coefficients, but other distributions can be used depending on the problem.

2. Likelihood:

- The likelihood represents the probability of the observed data given the parameters.
- In the context of Bayesian regression, the likelihood is typically assumed to be Gaussian (for normally distributed errors).
- The likelihood quantifies how well the model with a given set of parameters explains the observed data.

3. Posterior Distribution:

- The posterior is the probability distribution over the model parameters after observing the data. It combines the prior distribution with the likelihood of the observed data.
- The posterior incorporates both the prior beliefs and the evidence from the data, resulting in an updated belief about the parameters.
- The posterior distribution quantifies our updated uncertainty about the model parameters after seeing the data.

4. Bayes' Theorem:

- Bayes' Theorem is the core principle of Bayesian inference. It updates the prior distribution based on new data (via the likelihood) to compute the posterior distribution.

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

- The evidence (or marginal likelihood) is a normalizing constant to ensure the posterior distribution is a valid probability distribution.

Probabilistic Generative Models

Machine learning algorithms today rely heavily on probabilistic models, which take into consideration the uncertainty inherent in real-world data. These models make predictions based on probability distributions, rather than absolute values, allowing for a more nuanced and accurate understanding of complex systems. Probabilistic models are an essential component of machine learning, which aims to learn patterns from data and make predictions on new,

unseen data. They are statistical models that capture the inherent uncertainty in data and incorporate it into their predictions. Probabilistic models are used in various applications such as image and speech recognition, natural language processing, and recommendation systems.

Generative models aim to model the joint distribution of the input and output variables. These models generate new data based on the probability distribution of the original dataset. Generative models are powerful because they can generate new data that resembles the training data. They can be used for tasks such as image and speech synthesis, language translation, and text generation.

Probabilistic generative models are a type of statistical model that describe the process of generating data. These models are based on probability distributions and allow us to model complex data by capturing its underlying patterns and variability. They are useful for tasks such as density estimation, sampling, and learning latent representations.

Here are key aspects of probabilistic generative models:

1. Generative Process

- These models describe how data is generated in terms of a probabilistic process.
- For each data point, the model assumes that there is some hidden (latent) structure or variable, and the data is generated by sampling from a conditional probability distribution based on that latent variable.

2. Latent Variables

- Generative models often involve latent variables, which are hidden or unobserved variables that influence the observed data.
- The goal is to infer these latent variables to understand the structure of the data.

3. Joint Distribution

- Generative models define a joint probability distribution over the observed data X and latent variables Z, often written as P(X,Z).
- To generate new data, we first sample the latent variable Z from a prior distribution, then generate X from the conditional distribution P(X|Z).

Joint Probability

- A joint probability is a probability that measures the likelihood that two or more events will happen concurrently.
- If there are two independent events A and B, the probability that A and B will occur is found by multiplying the two probabilities. Thus for two events A and B, the special rule of multiplication shown symbolically is:
- P(A and B) = P(A) P(B).

Applications of Generative Models

- Data generation: Creating synthetic data, such as generating new images, text, or audio.
- Missing data imputation: Estimating missing values in datasets.
- Density estimation: Learning the probability distribution of data for anomaly detection or classification.
- Semi-supervised learning: Using generative models to improve learning when labeled data is scarce.

Types of Probabilistic Generative Models:

Gaussian Mixture Models (GMM)

Hidden Markov Models (HMM)

Latent Dirichlet Allocation (LDA)

Variational Autoencoders (VAEs)

Generative Adversarial Networks (GANs)

Probabilistic Discriminative Models

The discriminative model aims to model the conditional distribution of the output variable given the input variable. They learn a decision boundary that separates the different classes of the output variable. Discriminative models are useful when the focus is on making accurate predictions rather than generating new data. They can be used for tasks such as image recognition, speech recognition, and sentiment analysis.

Probabilistic discriminative models are a class of models used in machine learning that directly model the conditional probability of the target variable yyy given the input features x, i.e., P(y|x). Unlike **generative models**, which aim to model the joint probability distribution P(x,y) and are concerned with how the data is generated, discriminative models focus on learning the decision boundary between different classes in a dataset. They are designed to directly optimize the prediction of the target variable based on observed features.

Probabilistic discriminative models are highly effective for supervised learning tasks where the goal is to map input features to output labels, leveraging conditional probability for better accuracy and performance

Key Concepts in Probabilistic Discriminative Models:

1. **Conditional Probability**: Discriminative models are concerned with estimating the conditional probability of a label or class given a set of observed features. This allows them to focus on learning the decision boundary between classes without needing to model how the data itself is generated.
2. **Maximum Likelihood Estimation (MLE)**: Discriminative models often maximize the conditional likelihood of the data to find the parameters that best predict the target variable. This is achieved through optimization techniques, such as gradient descent.

3. **Decision Boundaries**: Since discriminative models focus on distinguishing between classes, they directly model the boundary between classes, resulting in a model that is often more efficient and accurate in classification tasks compared to generative models.

Advantages of Probabilistic Discriminative Models:

1. **Efficiency**: Discriminative models typically perform better for classification tasks because they directly focus on modeling the decision boundary, without needing to model the entire data distribution.
2. **Flexibility**: By focusing on $P(y|x)$, discriminative models can be more flexible when dealing with complex decision boundaries, particularly when using nonlinear models like neural networks.
3. **Higher Accuracy in Classification**: Because they focus on conditional probabilities and decision boundaries, discriminative models often achieve higher classification accuracy in practice compared to generative models.

Applications:

- **Classification**: Most classification tasks, including spam detection, sentiment analysis, and image recognition, rely on discriminative models.
- **Natural Language Processing (NLP)**: CRFs are widely used for tasks like part-of-speech tagging and named entity recognition.
- **Sequence Prediction**: Discriminative models like CRFs and neural networks are used for sequence labeling tasks in time series, speech recognition, and more.

Examples of Discriminative Models

- Logistic regression
- Support vector machines (SVMs)
- Traditional neural networks
- Nearest neighbor
- Conditional Random Fields (CRFs)
- Decision Trees and Random Forest

Bayesian Logistic Regression

Logistic Regression is one of the most popular ML models used for classification. It is a generalized linear model where the probability of success can be expressed as a sigmoid of a linear transformation of the features (for binary classification).

Logistic regression is a probabilistic model. Hence, it automatically enables us to compute the probability of success for a new data point as opposed to a hard 0 or 1 for success or failure. A

probability of 0.9 can probably be classified as Positive, while a probability of 0.1 can be classified as Negative. A probability of 0.5 implies we cannot take a call.

Bayesian inference for logistic analyses follows the usual pattern for all Bayesian analyses:

1. Write down the likelihood function of the data.

2. Form a prior distribution over all unknown parameters.

3. Use Bayes theorem to find the posterior distribution over all parameters.

Advantages of Bayesian Logistic Regression:

1. **Incorporation of Prior Knowledge**: By specifying a prior over the parameters, Bayesian logistic regression allows us to incorporate prior beliefs or domain knowledge into the model.
2. **Uncertainty Quantification**: Bayesian inference provides not just point estimates of the parameters but also a posterior distribution that reflects uncertainty in the parameter estimates. This uncertainty can be propagated into predictions, giving **confidence intervals** or **probabilistic predictions**.
3. **Regularization**: The prior acts as a form of regularization. For instance, a Gaussian prior centered at zero prevents the weights from becoming too large, helping to avoid overfitting. This is similar to L2 regularization in classical logistic regression.
4. **Robustness to Small Datasets**: Bayesian methods tend to be more robust in small-data regimes since the prior helps to prevent overfitting, making them ideal for scenarios with limited data.

Disadvantages of Bayesian Logistic Regression:

1. **Computational Complexity**: Bayesian logistic regression requires solving intractable integrals, and the use of approximation techniques (like MCMC or variational inference) can be computationally expensive, especially for large datasets.
2. **Modeling Complexity**: Bayesian methods require careful selection of priors and approximation methods, adding an additional layer of complexity compared to classical logistic regression.
3. **Approximation Errors**: The posterior distribution must often be approximated, and the quality of this approximation affects the results. Poor approximations may lead to suboptimal inferences.

Applications of Bayesian Logistic Regression:

- **Medical and Biological Data**: Bayesian logistic regression is widely used in fields where it is important to quantify uncertainty, such as in medical diagnostics and clinical trials.
- **Natural Language Processing (NLP)**: For tasks like text classification where there is prior knowledge about the expected importance of features (e.g., certain words), Bayesian logistic regression can provide robust results with uncertainty estimates.

- **Finance**: Bayesian methods are often employed in risk modeling, where quantifying uncertainty is crucial.

Key Concepts in Bayesian Logistic Regression:

Logistic Regression Framework:

Bayesian Perspective:

Posterior Distribution:

Posterior Inference:

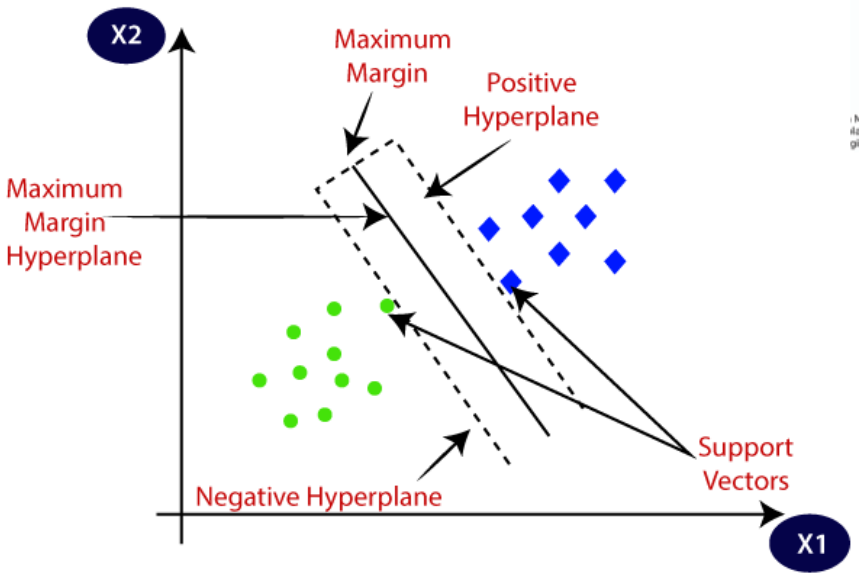
Prediction

Support Vector Machine (SVM)

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:



SVM can be of two types:

- **Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
- **Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

Hyperplane and Support Vectors in the SVM algorithm:

- **Hyperplane:** There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM.
- The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then hyperplane will be a straight line. And if there are 3 features, then hyperplane will be a 2-dimension plane.
- We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.
- The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

Key Concepts in Support Vector Machines:

1. **Linear SVM**:
    - In its simplest form, SVM is a linear classifier. For a binary classification task, it tries to find the optimal hyperplane that separates the two classes. The hyperplane is represented by the equation:

      $$wTx+b=0$$

      where:
        - w is the weight vector,
        - x is the feature vector,
        - b is the bias (or intercept).
    - The goal is to find the parameters w and b that maximize the margin between the two classes, where the margin is defined as the distance between the hyperplane and the nearest data points (support vectors) from each class.
2. **Margin and Support Vectors**:
    - **Margin**: The margin is the distance between the decision boundary (hyperplane) and the closest points from any class. SVM aims to maximize this margin to make the model more robust to new, unseen data.

    - **Support Vectors**: These are the data points that lie closest to the decision boundary. They are the most informative points since they determine the position of the hyperplane.

3. **Hard Margin vs. Soft Margin**:
    - **Hard Margin SVM**: In cases where the data is perfectly linearly separable, a hard margin SVM finds a hyperplane that perfectly separates the classes without any misclassification. However, this is often too restrictive, especially when dealing with noisy or overlapping data.
    - **Soft Margin SVM**: To handle cases where data is not perfectly separable, SVM introduces a **soft margin** by allowing some misclassifications.

Non-linear SVM (Kernel Trick):

- **Linear SVMs** are limited to linearly separable data, but real-world data is often non-linear. To deal with non-linear data, SVM uses a technique called the **kernel trick**.
- The kernel trick transforms the input data into a higher-dimensional space where it becomes linearly separable. This transformation is implicit, and the algorithm computes the dot product in this higher-dimensional space without explicitly transforming the data.

Advantages of SVM:

1. **Effective in High-Dimensional Spaces**: SVM performs well even when the number of dimensions is greater than the number of samples, which is useful in domains like text classification.
2. **Robust to Overfitting**: By maximizing the margin, SVM focuses on the most informative points (support vectors), making it less likely to overfit, especially when using regularization.
3. **Flexibility with Non-linear Data**: Through the use of the kernel trick, SVM can handle highly non-linear decision boundaries.
4. **Support for Small and Large Datasets**: SVM is effective for both small datasets (as it focuses on the margin) and large datasets when appropriately optimized.

Disadvantages of SVM:

1. **Choice of Kernel**: The performance of SVM is highly dependent on the choice of the kernel and its parameters (e.g., CCC and γ\gammaγ for the RBF kernel). It may require cross-validation or grid search to find the optimal settings.
2. **Computational Complexity**: Training an SVM can be computationally intensive, especially with large datasets or when using complex kernels. Solving the quadratic optimization problem becomes more challenging as the dataset grows.
3. **Memory Usage**: Since the algorithm relies on support vectors, SVMs can be memory-intensive, especially for large datasets with many support vectors.

4. **No Direct Probabilistic Output**: While SVMs are excellent classifiers, they do not directly provide probability estimates for class membership. However, **Platt scaling** can be used to transform the output into probabilities.

Applications of SVM:

1. **Text Classification**: SVMs are widely used for document categorization, such as spam detection or sentiment analysis, due to their efficiency in high-dimensional spaces (e.g., word vectors).
2. **Image Classification**: SVMs are commonly used in tasks like object detection and facial recognition due to their ability to handle high-dimensional image features.
3. **Bioinformatics**: SVMs are employed for gene classification, protein function prediction, and other tasks where the feature space is high-dimensional.
4. **Finance**: SVMs can be used for risk assessment, fraud detection, and stock market prediction.

Example: How SVM Works

For a two-class classification problem, assume we have two sets of points in a 2D space, one belonging to Class A and the other to Class B. The goal of SVM is to find the best separating hyperplane (in 2D, this is just a line) that maximizes the margin between the two classes. The points closest to the separating hyperplane from each class are the **support vectors**. The wider the margin, the better the generalization of the model to new data.

In a non-linear case, such as data that can't be linearly separated (e.g., concentric circles), SVM will apply the kernel trick to map the data into a higher-dimensional space where it becomes separable, finding an optimal hyperplane in this transformed space.

Kernel Trick

The **Kernel Trick** is a technique used in machine learning, particularly in Support Vector Machines (SVM) and other algorithms, to efficiently perform computations in high-dimensional feature spaces without explicitly mapping the data to those spaces.

Many machine learning problems, especially classification, require finding a decision boundary that separates the data points belonging to different classes. However, if the data is not linearly separable in the original feature space, a common solution is to transform (or map) the data into a higher-dimensional space where a linear boundary can be found.

For example, consider two classes that are not linearly separable in 2D space. By transforming the data into a 3D space, it might become linearly separable, allowing a hyperplane (linear separator) to effectively divide the classes. However, performing this transformation explicitly can be computationally expensive, especially when dealing with very high-dimensional spaces.

This is where the **kernel trick** comes in. Instead of explicitly transforming the data points to a higher-dimensional space, the kernel trick computes the **inner product** of the data points in the higher-dimensional space **implicitly** by applying a kernel function in the original space. This makes the process computationally feasible.

Advantages of the Kernel Trick:

1. **Computational Efficiency**: Directly mapping data to high-dimensional spaces can be computationally expensive, especially with large datasets. The kernel trick avoids this cost by calculating inner products without performing the transformation explicitly.
2. **Flexibility**: The kernel trick allows SVMs and other algorithms to work effectively in non-linear feature spaces, enabling them to handle more complex decision boundaries.
3. **Powerful Non-Linear Classification**: Using kernel functions like the RBF kernel, SVM can create very flexible decision boundaries that can separate even highly non-linear data.

Types of Kernels

In Support Vector Machines (SVM), kernel functions allow the model to handle non-linear relationships by implicitly mapping the input features into a higher-dimensional space where a linear separation is possible. This is known as the kernel trick. The key idea behind a kernel is that it allows us to compute the dot product of transformed features (in the higher-dimensional space) without explicitly performing the transformation.

Linear Kernel:

- This is the simplest kernel, which is equivalent to using the original features without any transformation. It is used when the data is linearly separable.

$$K(x_i, x_j) = x_i^T x_j$$

Polynomial Kernel:

- The polynomial kernel allows for non-linear relationships by considering polynomial combinations of the input features. It can model more complex decision boundaries by adjusting the degree of the polynomial.

$$K(x_i, x_j) = (x_i^T x_j + c)^d$$

where:

- c is a constant term (sometimes called the bias),
- d is the degree of the polynomial.

Radial Basis Function (RBF) Kernel / Gaussian Kernel:

- The RBF kernel is the most commonly used kernel in SVM. It maps the input features into an infinite-dimensional space, enabling SVM to handle highly non-linear data. The kernel depends on the Euclidean distance between the feature vectors

$$K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right)$$

where:

- γ\gammaγ controls the width of the Gaussian function, determining the influence of each support vector. A small γ\gammaγ value leads to a smooth decision boundary, while a large γ\gammaγ allows more flexible boundaries.

Laplacian Kernel:

- Similar to the RBF kernel, but it uses the **L1 distance** (Manhattan distance) instead of the squared L2 distance (Euclidean distance) in the Gaussian function.

$$K(x_i, x_j) = \exp\left(-\gamma |x_i - x_j|\right)$$

Exponential Kernel:

- This is a variant of the RBF kernel but uses the L1 distance (like the Laplacian kernel). It can be seen as an intermediate between the RBF and Laplacian kernels.

$$K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|\right)$$

Gaussian Mixture Model

A **Gaussian Mixture Model (GMM)** is a probabilistic model used to represent the presence of multiple subpopulations within an overall population, where each subpopulation can be modeled by a Gaussian (or Normal) distribution. It is often used for **unsupervised learning**, particularly for **clustering** tasks, but also finds applications in **density estimation** and **anomaly detection**.

Components-

Number of Components (K)

Mixing Coefficients

Mean Vector

Covariance Matrix

Gaussian (Normal) Distributions

Overall Probability Density Function