Practical Machine Learning with R

Lecture 6 Black Box Methods: Neural Networks and Support Vector Methods

Objectives:

- Intuition behind the paradigm of Black box algorithms and models.
- The representation of how the Neural Network simulates the working of the brain to classify the target feature.
- The Usage of Support Vector Machine to create classification or prediction as per hyperplane separation.

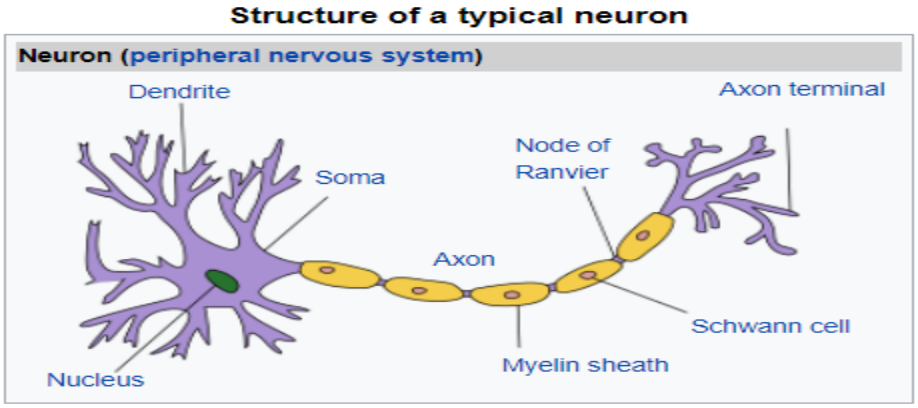
Black Box Methods:

- Black box processes are those for which the algorithm is not explicit to the user due to the complex underlying mathematical framework.
- To be able to understand the inner workings of the algorithm decoding of the algorithm is required which in most cases in a daunting task by its self.
- The functions that are working internally can be recursive, non linear which causes them to be less interpretable.
- One way to offset this disadvantage is achieved by applying large scale simulation experimentation and statistical hypothesis testing to build intuition related to how the various components will work together within different scenarios to provide optimized outcomes. The caveat is that if some crucial path is not explored then mission critical applications can be disastrous.
- The opacity of black box algorithm is worrisome because we do not wish to handle unpredictability in scenarios where the tasks being classified are critical for example scenarios related to disease detection.
- MIT students are building AIM Adaptable Interpretable Machine Learning which focuses on constructing Interpretable Neural Networks which self propogates its mathematical structure. Stanford scholars are transforming the representation of neural networks into tree like structures where every configuration can be explored. Additionally only the productive branches are being evaluated and the unproductive branches as well as leaves are being pruned by a software tool.
- To gain sense regarding the level of complexity, hypothetically if we have 6 layers and 50 nodes per layer then the total number of nodes are 300 and the total number of outcome possibilities are 2^{300}.The Stanford team is using Linear programming a technique from the domain area of Operations Research for this traversal. This tool will alleviate and mitigate risks due to unforeseen network paths.
- These algorithms help accomplish innovative tasks like self driving car, language translations,face recognition,text classification etc
- Neural Networks emulates the human brain processing for modelling purposes.
- Support Vector Machine uses multidimensional surfaces to define the relationship between features and outcomes.
- AI is a broad umbrella which comprises of domains like Robotics, Deep Learning, Expert Systems, Image recognition etc. Neural Networks is under the scope of Deep Learning.

ANN Artificial Neural Networks:

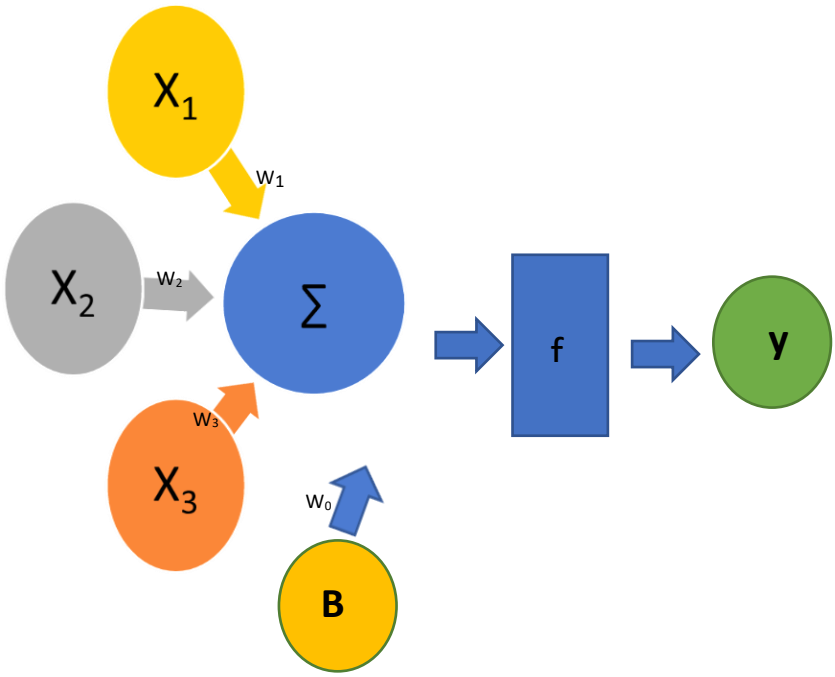
- ANN neural networks simulates how our brain responds to stimuli from sensory inputs by taking input from artificial interconnected neurons (nodes), generating a parallel processor and resulting an outcome consequent of the relationships traversed through the network.

- Human brain is composed of 85 billion neurons therefore compared to a human brain ANN is a very simplistic in structure and complexity but still is proven to heuristically model and generate substantive outcomes. In the contemporary context ANN are not capable of handling the complexity of the inner workings within the neurons of the human brain.
- Alan Turing a pioneering scientist proposed the "Turing test " which ranks a machine as intelligent in its appraisal if a human being cannot distinguish its behavior from a living creature.
- There are plethora of diverse applications that are utilizing neural networks which include though not limited to climatic pattern modelling , social trend mapping ,speech recognition systems.
- ANN are multifunctional and ubiquitous within the scope of machine learning algorithms. These learning algorithms can be utilized for supervised learning tasks like classification, numerical prediction as well as unsupervised learning tasks like clustering.



- For the human biological processes the incoming signals are received by the dendrites via a biochemical process which allows the weightage of these signals to be evaluated as per the importance or frequency of the impulse.
- The dendrite cell aggregates these impulse signals progressively until a threshold is reached consequent to which an output signal is generated via an electrochemical process through an axon.
- The axon further disseminates the signals to neighboring neurons , transmitting it through a gap called Synapse. The exact way of knowledge attainment or brain learning is elusive.

Perceptron: Basic unit of a neuron.A neural network has a input layer atleast one hidden layer and one output layer.Contains the inputs X_1,X_2 etc and there is a weight for these inputs w_1,w_2 etc. B is the bias.



- Bias is like the intercept in the linear equation. It helps in shifting the activation function up and down left and right to help the model to the data. The Bias helps the model fit the data better. The value of bias is 1 and the weight is set by the algorithm.
- A single ANN neuron analogous to the brain neuron can be represented by input signals X_1,X_2 and X_3 attaching the importance of these inputs by the numerical weights w_1,w_2 and w_3. This is analogous to human dendrites. The weight significance is amplified or dampened by the neural network as per the learning process by evaluating the error perceived in the training process.
- The weighted input signals from each node at a level are cumulated, then further processed by the activation function f which transforms them into outputs on the basis of a threshold that is reached. This analogous to the cell body's mechanism of handling input signals. If a threshold reached as per the activation function the signal progresses into further network layers and then finally as a output.
- The resulting outcome can be mathematically represented by the following relationship:

$$y(x) = f\left(\sum_{i=1}^{n} w_i x_i\right)$$

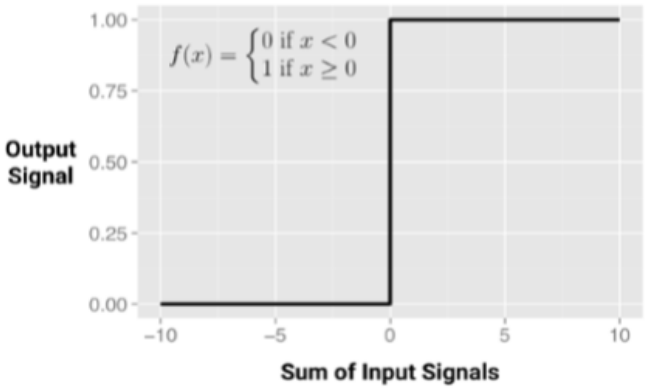
- ANN is composed of a network topology which incorporates the complex architectural frameworks of connected nodes(neurons) distributed across numerous layers which finally produce an output subsequent to the processing actualized by the activation function.
- Training algorithm are responsible for setting up the weights across the spectrum of nodes effectuating the inhibition or activation of neurons upon cumulation and activation function generation
- The initial neural network uses guesses for the weight as it starts as a ignorant network and then learns with subsequent layers as well as forward and back propogations..

Activation Functions

- The paramount purpose of the activation function is ascertaining if the aggregated weighted inputs have achieved a predefined threshold which results in the firing the output outcome signal.
- Activation functions can be quantified mathematically by various functional forms. The most common of these include Unit Step activation functions, Sigmoid activation function, Linear

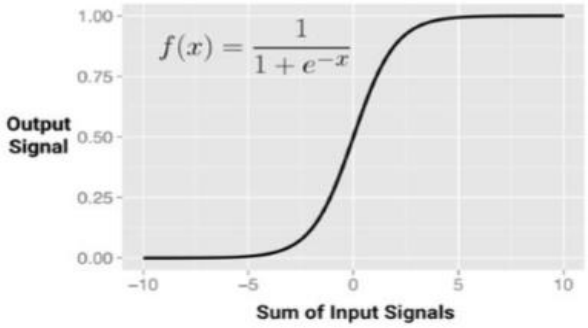
activation function, Saturation Linear activation function, Hyperbolic Tangent activation function and Gaussian activation function.

- **Unit Step Activation Function:**



- This activation function generates the output signal once the sum of the weighted inputs equals more than zero. This threshold activation function does not mathematically represent the relationships between data very closely therefore it is not used very often.

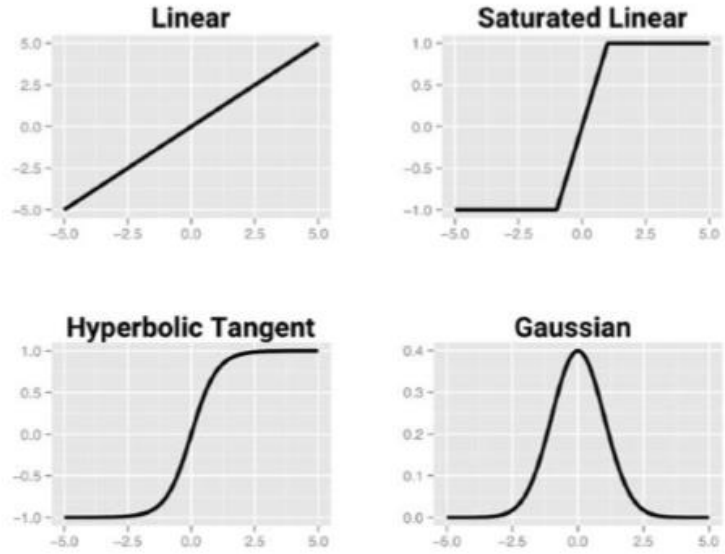
- **Sigmoid Activation Function**:



- The sigmoid function is the most commonly used activation function which is mathematically represented by the logistic sigmoid equation:

$$f(x) = \frac{1}{1 + e^{-x}}$$

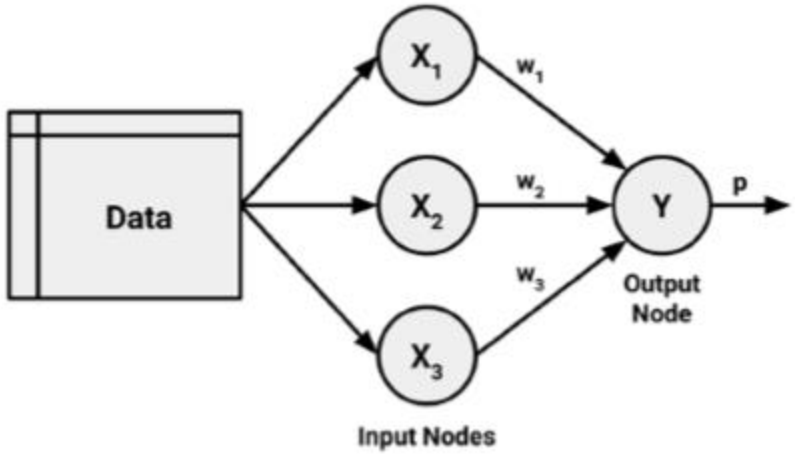
- The output of the sigmoid activation function is no longer binary but in fact it is within the range 0-1 which is a better representation of the data.
- Sigmoid function is differentiable therefore we can calculate the change of the output as per the change of inputs within a range of values. This property of Sigmoid activation function is helpful in optimizing the ANN algorithms. The only disadvantage for this is that it only limited positive output values because of which tanh activation are used.
- Activation Functions can be viewed on the following link: https://en.wikipedia.org/wiki/Activation_function



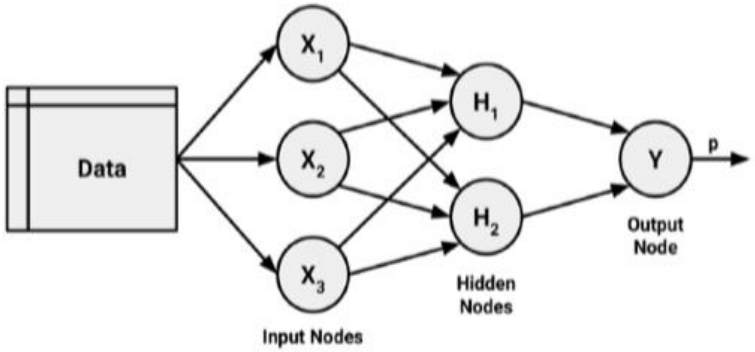
- Each of the activation function are uniquely set up in order to fit only certain types of data for example the linear activation function behaves like the regression model.
- Numerous Activation functions specifically Sigmoid functions have a issue related to the possible output ranges which are narrowly constrained within (0,1). Since input ranges are not constrained therefore corresponding to various input ranges, the output ranges will cluster at the boundaries of the range. To find the solution of the squashing problem the features are standardized or normalized in order to constrain the inputs within a range close to zero.
- The manipulation of the input data by the mechanism of standardization or normalization ensures that the large value features do not dominate the small valued features as well as the model is also trained quickly due to the bounded input values.

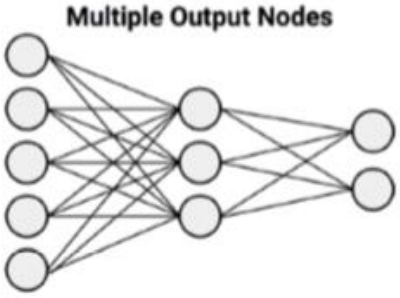
ANN Architectural Framework:

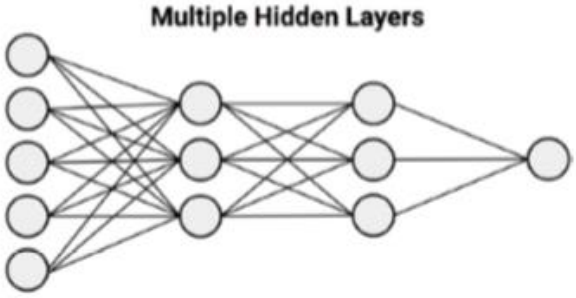
- The learning process of Neural Network in context of efficacy primarily dependent on the number of layers, number of nodes within a layer as well as the ability of back propogation.
- The complexity of the real-life scenarios can be modelled only by complicated neuron networks appropriately configured.
- Single layer network are used if the patterns are linearly separable but realistically the real life scenarios are more complex and require a more intricate multilayer network architecture.
- The intermediary layers within the multilayer neural networks are called hidden layers and in most cases framework is fully connected.



- If the network is unidirectional then these are called feedforward networks and if they are composed of multiple hidden layers then these are called Deep Neural Network. Such networks are called Deep Learning. There can be multiple outputs as well. The activation function is used all the hidden as well as output layer. A fraction of the input is used for each layer to predict and then this prediction is refined through the layers.



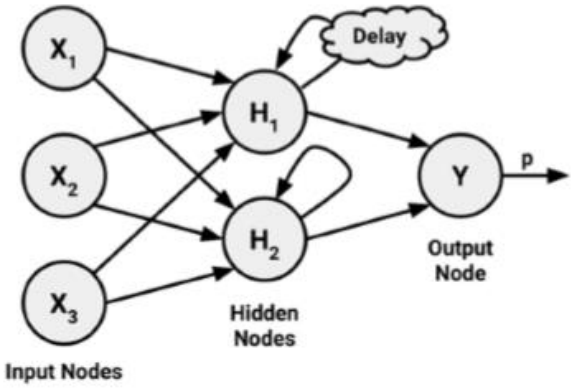




- The networks that allows the propagation of signals in both directions are called recurrent network or feedback networks. These are more powerful that feedforward networks as they depict the real working

of the biological neurons which represent real life scenarios. These help optimize the weights by the knowledge gained.

- Short term memory or delay is also very effective in obtaining the final outcomes over the continuum of time. This helps in learning from a sequence of events over a period of time. Stock market prediction can benefit from this ability.



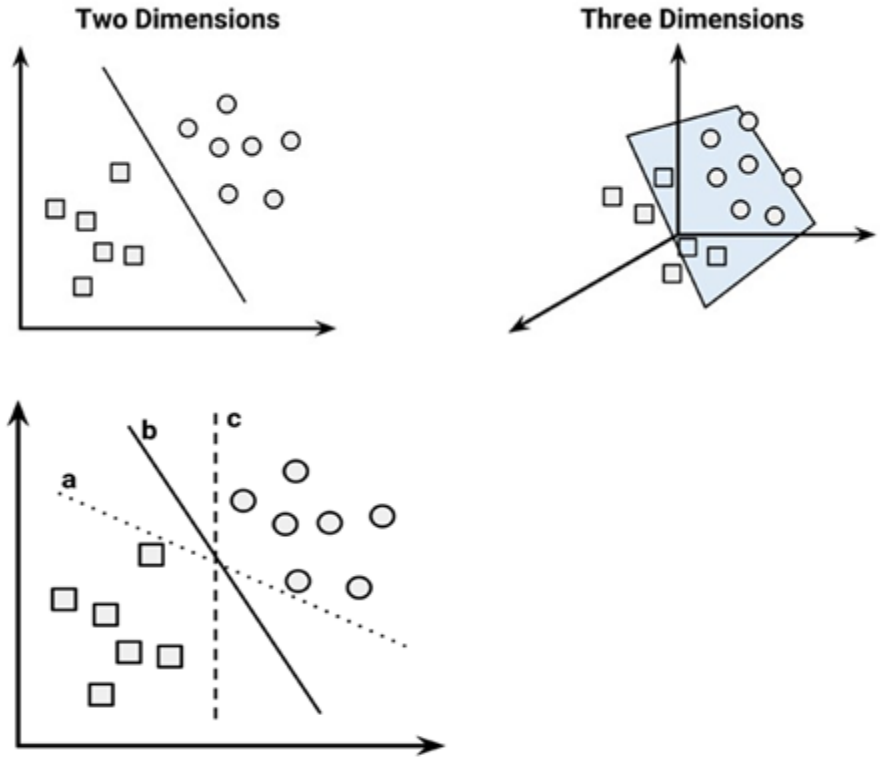
- Back propagation networks are good for numerical classification, are able to model complex scenarios and make very few assumptions regarding the underlying data relationships.
- Back propogation causes the architecture to be computationally complex , might cause overfitting and be difficult to interpret due to the black box paradigm.
- The semantics of back-propogation entails the repeated iterations thorough the network path with each forward and backward traversal defined as an epoch.
- For the first epoch traversal without a priori knowledge the weights are assigned to the inputs at random. Once the forward phase is accomplished the output signal is fed into the backward phase which compares the output to the target value within the training data and the error that is perceived is propagated backwards to recalibrate the weights to improve the fit of the model.
- The technique used to optimize the weight assignment is achieved by the technique called Gradient Descent.
- The back propagation algorithm uses the derivative of the each neuron's activation function in a manner that it identifies the how the magnitude of error will increase as the weights change. The algorithm will try to achieve the maximum reduction of error by an amount quantified by learning rate. The larger the learning rate the faster the algorithm will descent down the gradients.
- Conventionally Multilayer feedforward networks called Multilayer Perceptron are used for various ML implementations.
- The number of input nodes depend on the number of features in the input data and the outcome/outcomes depends on the number of class levels within the outcome.
- The number of hidden layers has to be determined by the user judiciously considering the tradeoff between number of input nodes, amount of training data, noise within the data and complexity of the learning procedure.
- Bias(how well the model fits the training data) and the variance(how well the model predicts the future examples with the training obtained) has to be optimized. Overfitting occurs when there is low bias but high variance and underfitting is when there is high bias and high variance.
- If the network architecture is too complex it might overfit the training data and these might be computationally expensive and slow to train.
- There are many techniques to avoid overfitting. These include retraining neural networks, multiple neural networks in parallel and averaging their outputs, early stopping by before model starts overfitting, regularization which adds a term in the error function to smoothen the outputs. Underfitting can be stopped by adding additional neurons,layers.

Mathematics for neural network:

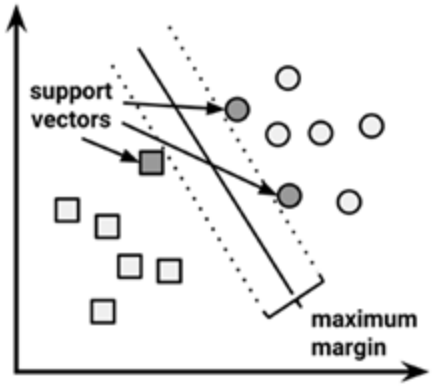
https://juxt.pro/blog/posts/neural-maths.html

Support Vector Machine

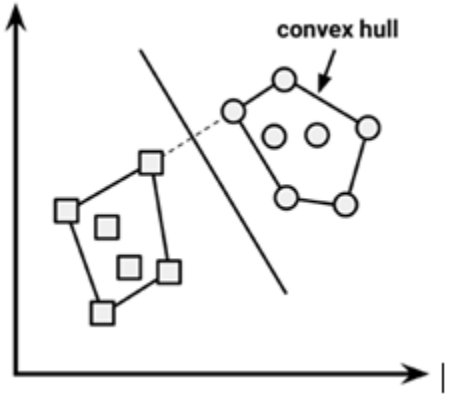
- SVM is a technique of partitioning the multidimensional data using hyperplane referencing the examples and their feature values.
- This mechanism incorporates the instance based learning classification achieved in KNN and numeric prediction achieved Linear Regression.
- SVM can be used for Classification and Numeric Prediction.
- Examples related to SVM applications include identification of cancer by genetic expression patterns, text categorization.
- The hyperplane can partition the examples having the same class values. If the classes are separated perfectly by using the linear hyperplane surface then these classes are called linearly seperable.
- Hyperplane is one dimension less than the dimension used to represent the data. For two dimension the hyperplane is a line versus for a data represented by three dimension the hyperplane is a plane. SVM entails finding the best hyperplane for the division.



- SVM helps decide the best separating hyperplane called the Maximum Margin Hyperplane (MMH) which separates the classes to the maximum extent. All the three lines partition the data into differentiated classes but to ensure that the model generalizes well for testing data and does not get affected by noise in the data the hyperplane chosen should be the best.



- Support vectors are examples at the boundary of MMH Maximum Margin Hyperplane. Each class should have atleast one support vector even though there can be more than one support vectors. The examples with generated arrows are the support vectors.
- The Support Vectors can help determine the MMH and even though there might be many features the classification set up by support vector makes it compact.



- Hypothetically if we wish to find MMH for linearly seperable classes then we have to ensure that the MMH is at the farthest distance from the two groups. The disparate groups were constructed into convex hulls on either sides. The MMH is the perpendicular bisector of the shortest line between the two convex hulls. Quadratic Optimization conducts this on computers.

- The hyperplane can be represented by the equation

$$\boldsymbol{wx} + b = 0$$

  Where w is a set of n vectors representing weights for an n dimensional space. b is the bias. Bias is analogous to the intercept in a slope intercept equation.
- The goal is to find the set of weights that define the hyperplanes as follows:

$$\boldsymbol{wx} + b \geq 1$$
$$\boldsymbol{wx} + b \leq -1$$

  The examples if linearly seperable can be separated on either sides of the aforementioned planes.
  Distance between these two planes by vector geometry is

$$D = \frac{2}{||\boldsymbol{w}||}$$

$||w||$ is the Euclidean Norm which is the distance from the origin to vector $\boldsymbol{w}$

https://en.wikipedia.org/wiki/Euclidean_distance

- To maximises the Distance the Euclidean norm has to be minimized. This can be symbolically represented as follows:

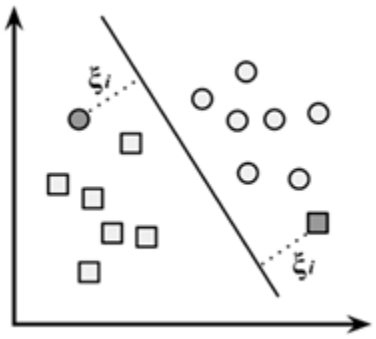
$$\boldsymbol{min}\ \frac{\mathbf{1}||w||^2}{\mathbf{2}}$$

such

$$\boldsymbol{y_i(wx_i - b) \geq 1 \forall x_i}$$

We are minimizing the norm and to make the math easier the we divide by 2 and the second line states that this is subject to the yi point is classified accurately. Y essentially indicated the class and is transformed to 1 or -1. The process is intensive therefore quadratic programming performs the appropriate steps.

Non Linearly Separable Data:



If the data are not linearly separable creating soft boundary margin by the use of a slack variable which allows for some misclassification. A cost value is used to penalize the misclassified points and instead of finding the MMH the total cost is minimized.

$$\boldsymbol{min}\ \frac{\mathbf{1}||w||^2}{\mathbf{2}} + \boldsymbol{C}\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{\varepsilon_i}$$

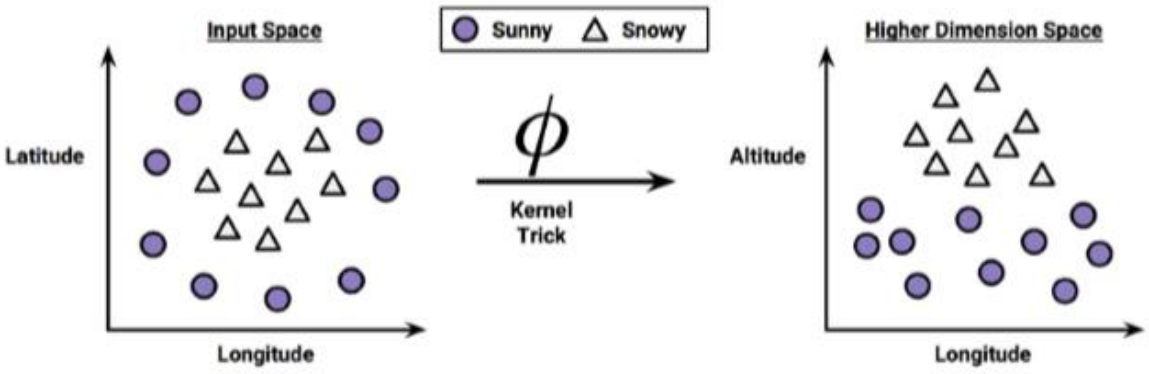
such that

$\boldsymbol{y_i(wx_i - b) \geq 1 - \varepsilon_i \forall x_i \varepsilon_i}$**>=0**

- The higher the C value the harder the optimization algorithm will work to obtain absolute separation. A lower C value gives priority to get a wider margin. The two paradigm have to balanced and the balance determines the creation of a model that generalizes well for the test data.

Using Kernels for non linear spaces:

- SVM can still be classified by introducing a slack variable and some of the examples will be misclassified
- A technique called Kernel trick if applied transforms the space to a higher dimension. This introduction helps to linearlize the space due the change in perspective.



- The first graph is analogous to the birds eye view whereas the second graph is the view from the ground level. The second graph shows linear separation and realistically snow is found on higher altitudes.
- Kernel transformation creates a higher dimension by creating new features using the current ones and applying relevant mathematical manipulations to highlight the new feature as an numerical interlinkage between the two or more features. This way new but qualified features can be obtained and used for the classification and prediction techniques.
- For instance the point closer to the center as per their latitudes and latitudes will have a higher altitude.
- SVM can be used for both classification and numerical prediction though to obtain the best model different kernels and parameters have to be explored.
- SVM do not get influence by noisy data and do not usually overfit the data but the algorithm is slow if there are numerous examples and features.
- SVM are more efficient as compared to Neural Networks since there are several popular algorithms that are currently implemented.
- The accuracy of these is very high though they do become complex and are perceived as black box models.
- The Kernel transformation is denoted by the dot product as follows
  :

$$K(\boldsymbol{x_i}, \boldsymbol{x_j}) = \emptyset(\boldsymbol{x_i}) * \emptyset(\boldsymbol{x_j})$$

- $\emptyset$ is the mapping that converts the space to a higher dimension.
- Linear Kernel transformation therefore it can be depicted as the product of the features.
  $K(\boldsymbol{x_i}, \boldsymbol{x_j}) = (\boldsymbol{x_i}) * (\boldsymbol{x_j})$

  - Polynomial Kernel adds a simple non linear transformation

$K(\boldsymbol{x_i}, \boldsymbol{x_j}) = (\boldsymbol{x_i} * \boldsymbol{x_j} + \mathbf{1})^d$

Sigmoid Kernel

$K(\boldsymbol{x_i}, \boldsymbol{x_j}) = tanh(k\boldsymbol{x_i} * \boldsymbol{x_j} -)^d - \delta$

$Kappa\ k\ and\ delta\ \delta$ are sigmoid parameters and this is equivalent to the activation function of the Neural Networks.

Gaussian RBF Kernel

Conventionally often time Gaussian Kernel is used as a starting point.

$$K(\boldsymbol{x_i}, \boldsymbol{x_j}) = e^{\frac{-||x_i - x_j||}{2\sigma^2}}$$

There is no specific principle that we can follow to implement a specific learning task. This is depends on the task to be learned, the amount of training data as well as relationships between features. Kernels are arbitrary.