CSE 673
COMPUTATIONAL VISION

VENU GOVINDARAJU
DEEN DAYAL MOHAN

University at Buffalo
Department of Computer Science and Engineering
School of Engineering and Applied Sciences

Covid-19 Guidelines

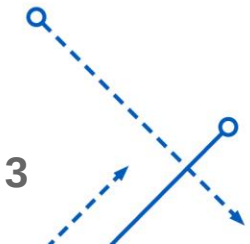
- Effective Aug. 3, the University at Buffalo will require all students, employees and visitors – regardless of their vaccination status – to wear face coverings while inside campus buildings. This includes classrooms, hallways, libraries and other common spaces, as well as UB buses and shuttles.

- Students are expected to wear mask in class during lectures (unless you have a UB approved exception)

- Public Health Behavior Expectations https://www.buffalo.edu/studentlife/who-we-are/departments/conduct/coronavirus-student-compliance-policy.html

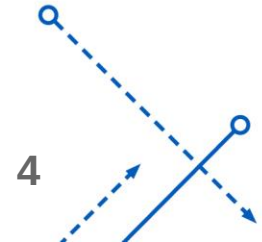
Course Details

- The Grades will be based on a curve with B+ being assigned to the median score.
- The link to the course website is https://cubs-ub.github.io/Computational_Vision/
- The link to piazza is piazza.com/buffalo/fall2021/cse673 . Please sign up
- Office hours and locations will be posted on piazza and course website.



Agenda

- Review Linear Algebra
- Review ML
- Answer question before add drop



LINEAR ALGEBRA REVIEW

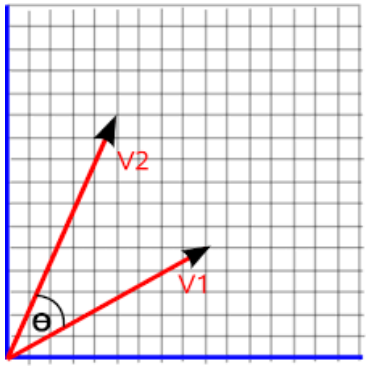
University at Buffalo
Department of Computer Science and Engineering
School of Engineering and Applied Sciences

Basics of Linear Algebra

- Vector - $X \in R^n$ a vector with n entries, where $X_i \in R$ is the ith entry

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n$$

- What does that mean geometrically ?

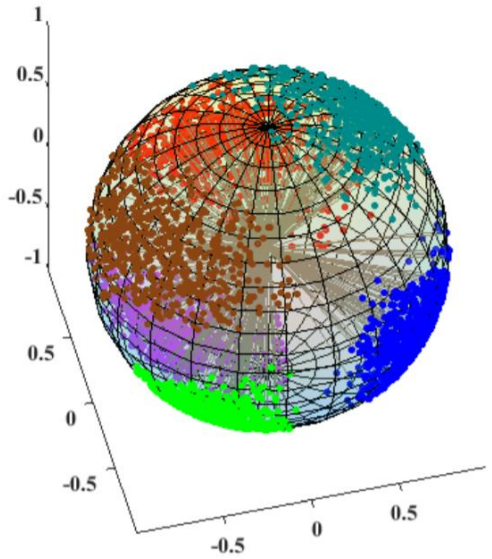


Basics of Linear Algebra

- Length of X, i.e the l2 norm of X ?

$$\|x\| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$$

- What is the space of all vectors which are unit normalized



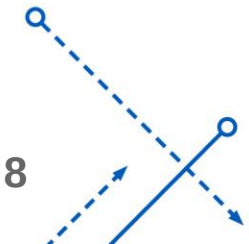
Basics of Linear Algebra

- Matrix — We note $A \in R^{mxn}$ a matrix with m rows and n columns, where $A_{i,j} \in R$ is the entry located in the I^{th} row and j^{th} column

$$A = \begin{pmatrix} A_{1,1} & \cdots & A_{1,n} \\ \vdots & & \vdots \\ A_{m,1} & \cdots & A_{m,n} \end{pmatrix} \in \mathbb{R}^{m \times n}$$

- Identity matrix is a matrix with all the main diagonal elements as 1 and rest as 0

$$I = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix}$$



Basics of Linear Algebra

- Vector Inner Product is given by

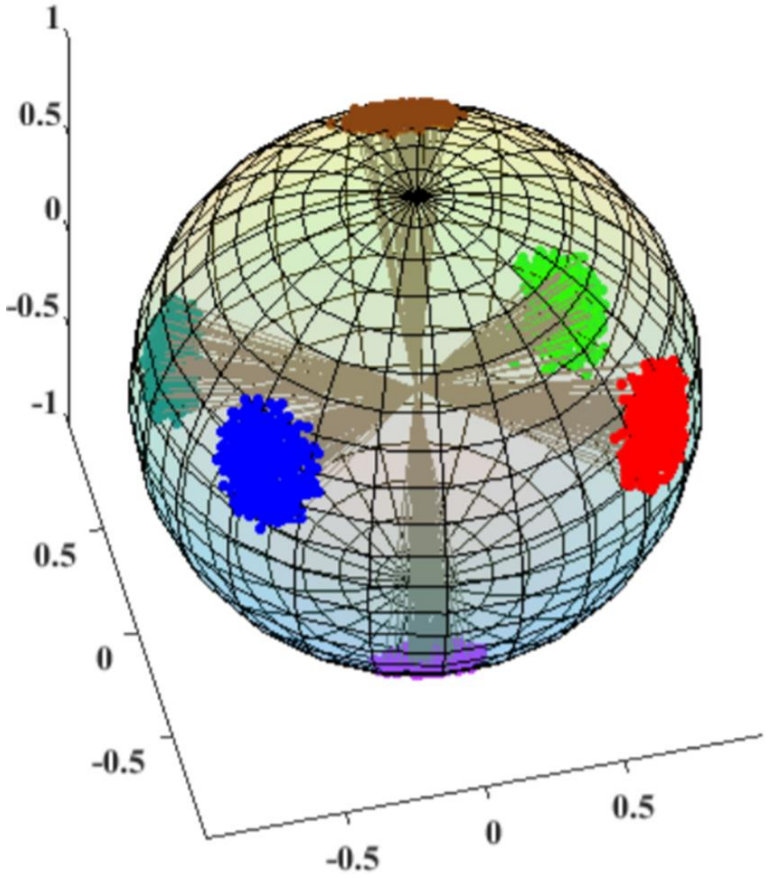
$$x^T y = \sum_{i=1}^{n} x_i y_i \in \mathbb{R}$$

- Vector Outer Product is given by

$$xy^T = \begin{pmatrix} x_1 y_1 & \cdots & x_1 y_n \\ \vdots & & \vdots \\ x_m y_1 & \cdots & x_m y_n \end{pmatrix} \in \mathbb{R}^{m \times n}$$

Basics of Linear Algebra

- If my vectors are both L2 normalized, and I take the dot product what do I get ?
  - Cosine similarity of the two vectors
- What does this mean geometrically ?
  - If cosine similarity is high that means vector lie close to each other in the hypersphere



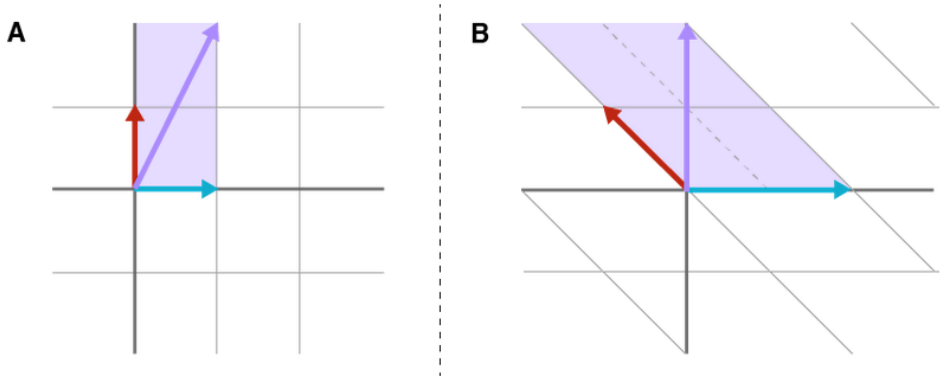
Basics of Linear Algebra

- Transpose of a matrix is given by $\forall i, j, \quad A^T_{i,j} = A_{j,i}$
- Inverse of an invertible matrix is noted as A^{-1} and is only matrix such that $AA^{-1} = A^{-1}A = I$
- Trace of a square matrix tr(A) is the sum of its diagonal entries $\mathrm{tr}(A) = \sum_{i=1}^{n} A_{i,i}$
- A set a vectors are said to be linearly dependent if one vector in the set can be defined as linear combination of other vectors
- If no vector can be written as a linear combination of other vectors, then the set of vectors can be said to be independent

Basics of Linear Algebra

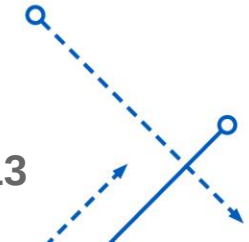
- What is the Geometric Intuition of a Matrix Vector Product ?

$$\overbrace{\begin{bmatrix} 2 & -1 \\ 0 & 1 \end{bmatrix}}^{A} \overbrace{\begin{bmatrix} 1 \\ 2 \end{bmatrix}}^{\mathbf{x}} = \overbrace{\begin{bmatrix} 0 \\ 2 \end{bmatrix}}^{f(\mathbf{x})}$$



Basics of Linear Algebra

- What is the intuition of a determinant ?
  - Determinant of the transformation matrix determine the factor by which, the area original space is scaled
- What does a zero determinant mean ?
  - It means that the transformation squishes the area on the original space to 0
- What is the relationship of determinant of matrix to the eigen values ?



Basics of Linear Algebra

- Matrix A is Positive semi-definite if

$$\boxed{A = A^T} \quad \text{and} \quad \boxed{\forall x \in \mathbb{R}^n, \quad x^T A x \geqslant 0}$$

- Given a matrix A, λ is said to be an eigenvalue of A if there exists a vector $z \in R^n \setminus \{0\}$, called eigenvector, such that we have

$$\boxed{Az = \lambda z}$$

- If A is symmetric matrix, then A is diagonalizable by a real orthogonal matrix U.

$$\boxed{\exists \Lambda \text{ diagonal}, \quad A = U \Lambda U^T}$$

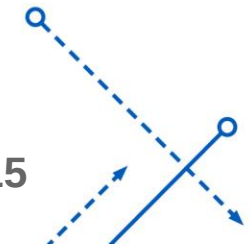
Machine Learning

- A computer program is said to learn from experience(E) with some class of tasks (T) and a performance measure(P) if its performance at tasks in T as measured by P improves with E” - Tom Mitchell

- Field of study the gives computers the ability to learn without being explicitly programmed

- The data that is used by the computers to learn is called training data.

- Machine learning task can be classified into several broad categories
  - Supervised Learning
  - Semi Supervised Learning
  - Unsupervised Learning

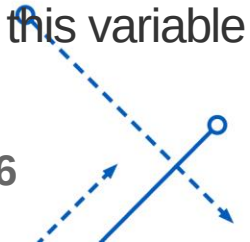


Supervised Learning

- Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs.
- The training data has values of depended variable corresponding to the independent variables.
- The goal of supervised learning algorithm is to learn a function that maps independent variables to the dependent variable (features to label)

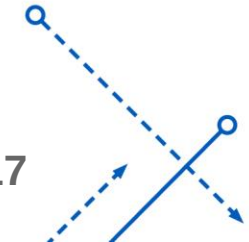
$$Y = F(X)$$

- The label or the dependent variable can be categorical or continuous. Depending upon the data type of this variable the task changes.



Regression vs Classification

- The learning task becomes regression if the dependent variable is a continuous value.
- The learning task is called classification if the dependent variable is a categorical.
- Linear Regression, Regression Trees etc. are algorithms which tries to learn a function which map independent variable to a continuous value output. Logistic Regression, Classification Trees etc. are algorithms used for classification task
- Generally for Regression task, we measure the performance of an algorithm based on the some standard error function like ‘Mean Square Error’.
- Classification tasks use measures like ‘Accuracy’ to evaluate the performance of the algorithm



Linear Regression

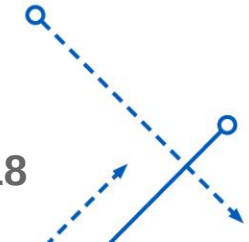
- Linear Regression attempts to model the relationship between independent and dependent variables as a linear function

- The dependent variable is expressed as a linear combination of the independent variables

- If we consider $X_1, X_2, X_3 \ldots X_n$ as the independent variables, then dependent variable **Y** is written as

$$Y = W_0 + W_1X_1 + W_2X_2 + \ldots\ldots W_nX_n$$

- $W_0, W_1, \ldots\ldots\ldots W_n$ are called the parameters of the model.



Linear Regression

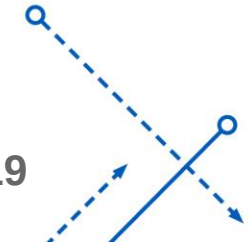
- The equation can be written in matrix form as

$$Y = W^T X$$

- W is a Vector of $W_0, W_1, \ldots\ldots W_n$ and **X** is matrix of the independent variables values

- The objective of the linear regression is to obtain vector **W,** which minimizes the predicted value of the dependent variable to actual value.
- The objective function can be written as :

$$J = \sum_{1:=1}^{n} \left(y_i - w^T x_i\right)^2$$



Closed Form Solution

- Using closed form solution of the equation, we get

$$\hat{w} = \left(x^T x\right)^{-1} x^T y$$

- $\hat{w}$ is the set of values of W which minimizes the error between actual value of the dependent and the predicted value

- The closed form solution for obtaining the optimal values of $\hat{w}$ is not always used. Why ?

Closed Form Solution

- Using closed form solution of the equation, we get

$$\hat{w} = \left(x^T x\right)^{-1} x^T y$$

- $\hat{w}$ is the set of values of W which minimizes the error between actual value of the dependent and the predicted value

- The closed form solution for obtaining the optimal values of $\hat{w}$ is not always used. Why ?
  - Computational Complexity

- In order to overcome this bottleneck, the minimization of the objective function is treated as a optimization problem.

Gradient Descent

- Gradient descent is a first-order iterative optimization algorithm for finding the local minimum of a differentiable function

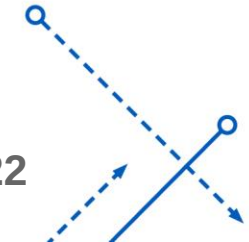
- To find a local minimum of a function using gradient descent, one takes steps proportional to the negative of the gradient (or approximate gradient) of the function at the current point.

- The parameters of the model are updated using gradient descent

$$\boldsymbol{w = w - \eta \nabla J(w)}$$

- Here $\nabla J(w)$ represent the gradient of function $J(w)$

- The value of η represents learning rate, which decides the size of the step to take in the direction of the gradient.

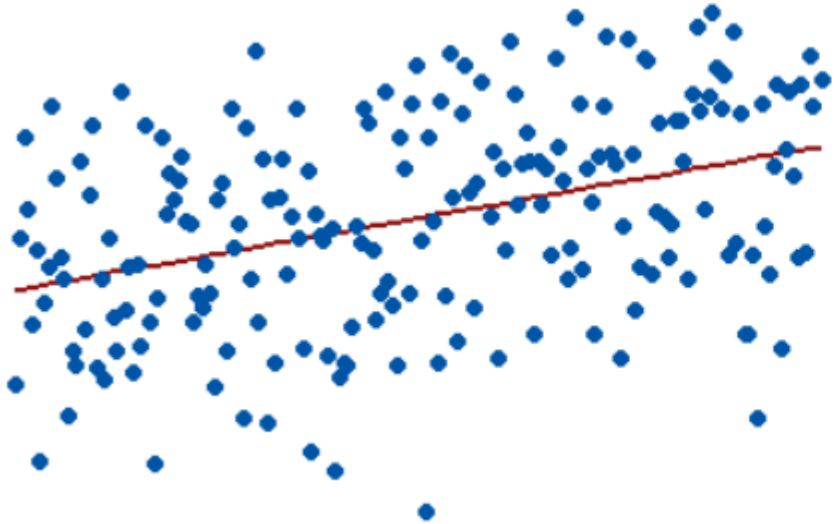


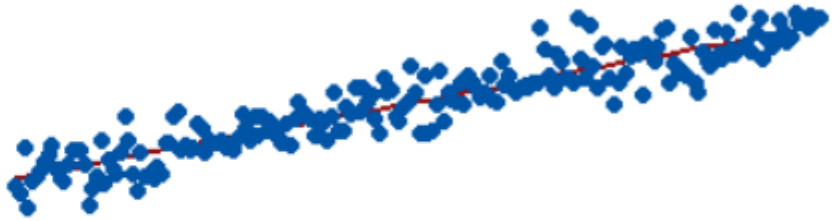
R^2 Estimate

- R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression

- R^2 value = Explained variation / Total variation

- An R^2 of 1 indicates that the regression predictions perfectly fit the data.



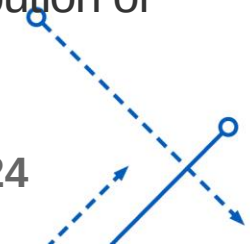


Classification

- The goal of supervised learning algorithm is to learn a function that maps independent variables to the dependent variable (features to label)

$$Y = F(X)$$

- The learning task is called classification if the dependent variable is a categorical
- Broadly classification methods can be divided into discriminative classifiers and generative classifiers.
- Discriminative models learn the boundary between classes whereas generative models learn the distribution of individual classes



Classification

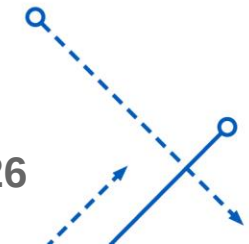
- Classification can be a binary classification or a multi class classification
- Binary Classification has two classes i.e. class 1 or 0 , yes or no;
- Classification of an email into SPAM or not SPAM is an example of Binary Classification
- When the dependent variable may belong to one of ‘K’ categories, then that is a multi class classification problem.
- Classification of an image to cats, dog or humans is an example for multiclass classification

Logistic Regression

- Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable.
- Logistic Regression is a binary classifier, as it can only predict binary outputs.
- Sigmoid function which is a special case of logistic function is used in logistic regression
- Similar to linear regression the interaction between the independent variables is captured using

$$F(X) = W^T X$$

- Where F(X) is a continuous value (as in linear regression). We use Sigmoid function to map it to a binary value



Logistic Regression

- The dependent variable y is given by

$$y = \frac{1}{1 + e^{-(w^T x)}}$$

- What will be the range y ?

Logistic Regression

- The dependent variable y is given by

$$y = \frac{1}{1 + e^{-(w^T x)}}$$

- What is the value of range of value of y ?
  - Yes it is between 0 to 1.
  - It is the confidence (or probability p) of that data sample belonging to a particular class
  - 1-p would be the probability of the data sample belonging to the other class

Logistic Regression

- Similar to what we did for linear regression, we have to create loss function(objective function) and we try to minimize it.

$$p(y=1|\mathbf{x};\mathbf{w}) = \sigma(\mathbf{w}^\top \mathbf{x})$$
$$p(y=0|\mathbf{x};\mathbf{w}) = 1-\sigma(\mathbf{w}^\top \mathbf{x})$$

- This can be written as

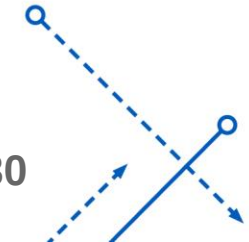
$$p(y|\mathbf{x};\mathbf{w}) = \left(\sigma(\mathbf{w}^\top \mathbf{x})\right)^{y}\left(1-\sigma(\mathbf{w}^\top \mathbf{x})\right)^{(1-y)}$$

- If there are N points in the dataset

$$\prod_{i}^{N}\left(\sigma(\mathbf{w}^\top \mathbf{x}_i)\right)^{y_i}\left(1-\sigma(\mathbf{w}^\top \mathbf{x}_i)\right)^{(1-y_i)}$$

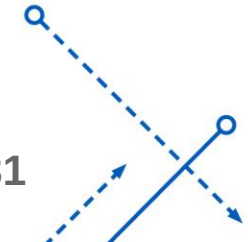
K-Nearest Neighbor (K-NN)

- K-Nearest Neighbor algorithm is a non parametric method
- In k-NN classification, the output is a class membership.
- An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors
- This is a supervised learning algorithm and should not be confused with K-means clustering which is an unsupervised learning algorithm
- A peculiarity of the k-NN algorithm is that it is sensitive to the local structure of the data



K-Nearest Neighbor (K-NN)

- In order to perform K-Nearest Neighbor algorithm, a distance metric has to be selected.
- The distance metric can be Euclidean distance, Cosine similarity etc.
- The training data consist of features and labels associated with the features.
- The goal is to predict the labels for the features in the test data set.
- K in KNN stands for the number of neighbors considered in the algorithm

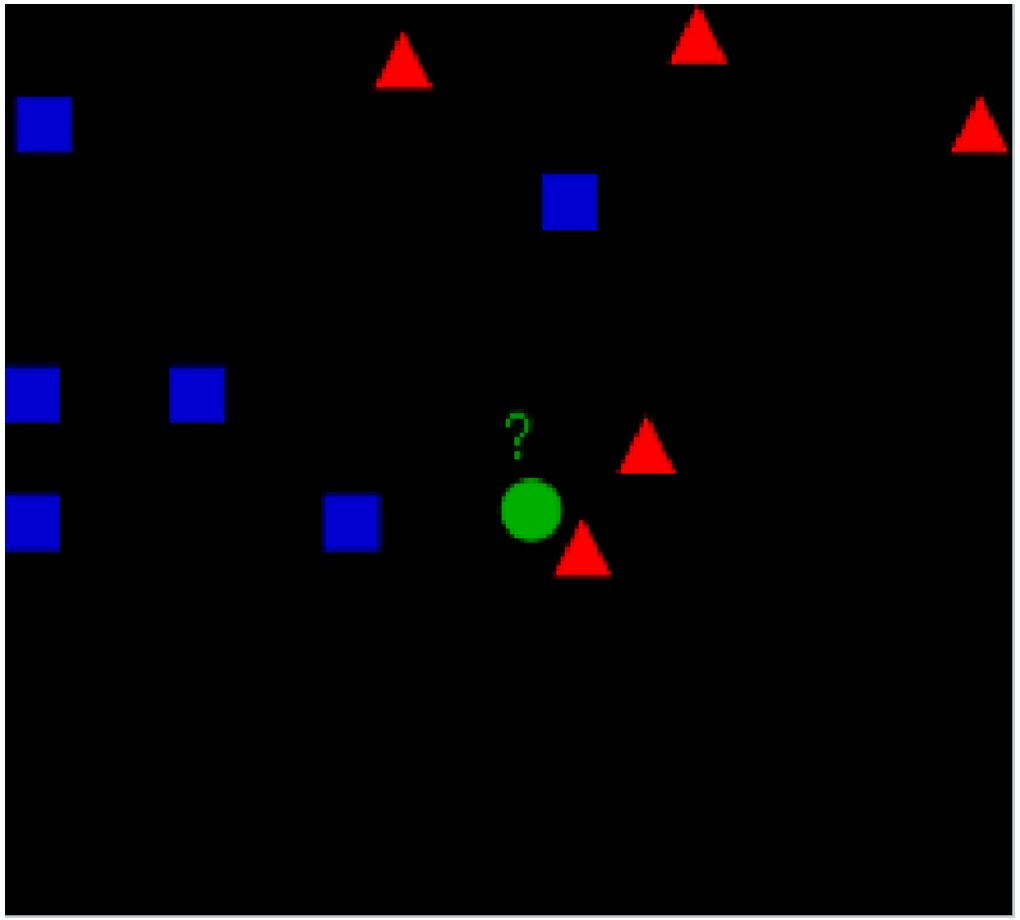


K-Nearest Neighbor (K-NN)

- Distance of each feature from the test set to all the feature vectors of the train set is calculated
- Top ‘K’ closest neighbors are found for each test set sample.
- The test sample is classified by assigning the label which is most frequent among the k training samples nearest to that test sample point.
- The optimal value of ‘K’ is a hyper parameter, which is found by trail and error
- K-NN algorithm will require to search entire train dataset to find the ‘K’ nearest neighbors of a query sample.

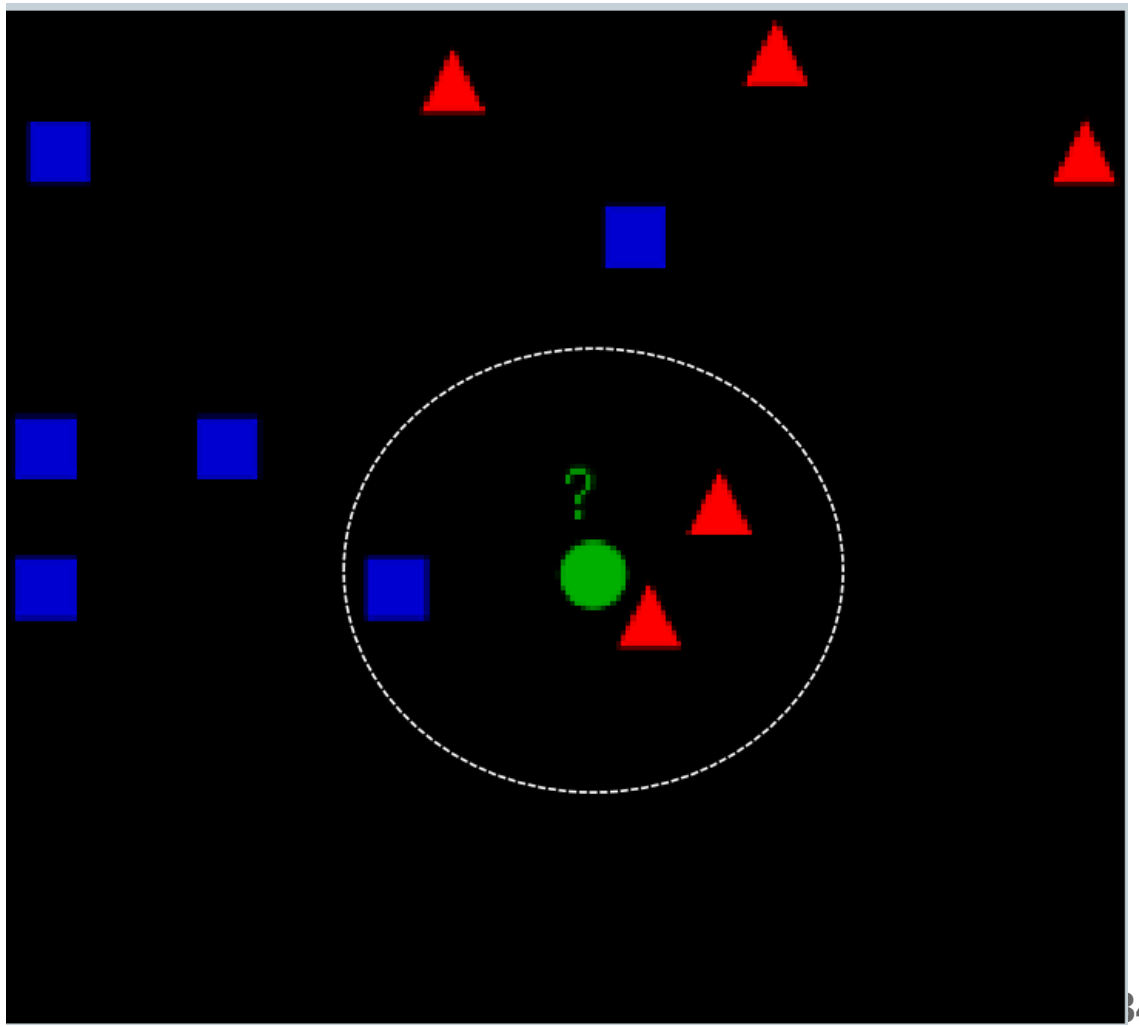
K-Nearest Neighbor

- Let triangle and square represent two classes in the training set
- Let the circle be a test (query) sample.
- What would be the predicted class of the query if K = 3 ?



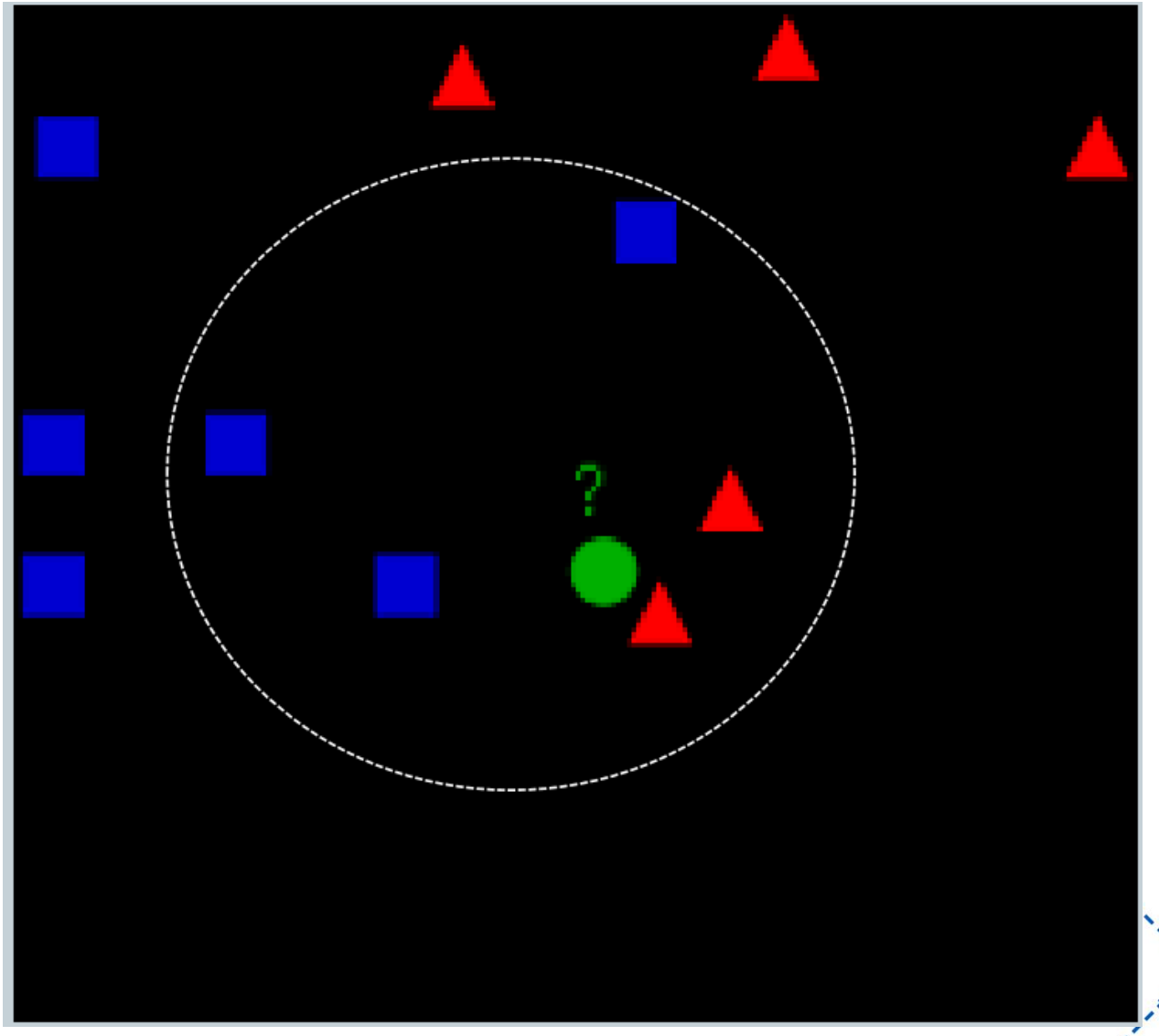
K-Nearest Neighbor

- Let triangle and square represent two classes in the training set
- Let the circle be a test (query) sample.
- What would be the predicted class of the query if K = 3 ?
  - ➢ Label is triangle



K-Nearest Neighbor

- Let triangle and square represent two classes in the training set
- Let the circle be a test (query) sample.
- What would be the predicted class of the query if K = 3 ?
  - ➢ Label is triangle
- What would be the predicted class if K= 5 ?
  - ➢ Label is square

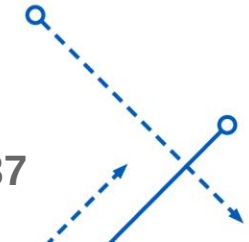


K-Nearest Neighbor

- The main disadvantage of the KNN algorithm is that it is a lazy learner, i.e. it does not learn anything from the training data and simply uses the training data itself for classification
- A drawback of the basic "majority voting" classification occurs when the class distribution is skewed.
- Examples of a more frequent class tend to dominate the prediction of the new example, because they tend to be common among the k nearest neighbors due to their large number
- Another disadvantage of this approach is that algorithm does not generalize well and also not being robust to noisy data

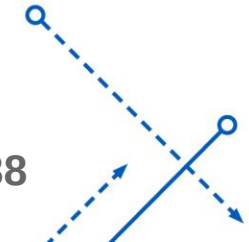
Support Vector Machine

- The parametric classifier tries to estimate values of optimal **W** from the training data. These parameters W is used to classify test
- From a geometric perspective discriminative classifier finds a decision boundary between points such that given a new test point it is able to identify the class correctly
- Logistic Regression learns the decision boundary by minimizing the objective function which is the cross entropy loss
- How do we know which is the best decision boundary, if there are multiple such decision boundaries possible?



Support Vector Machine

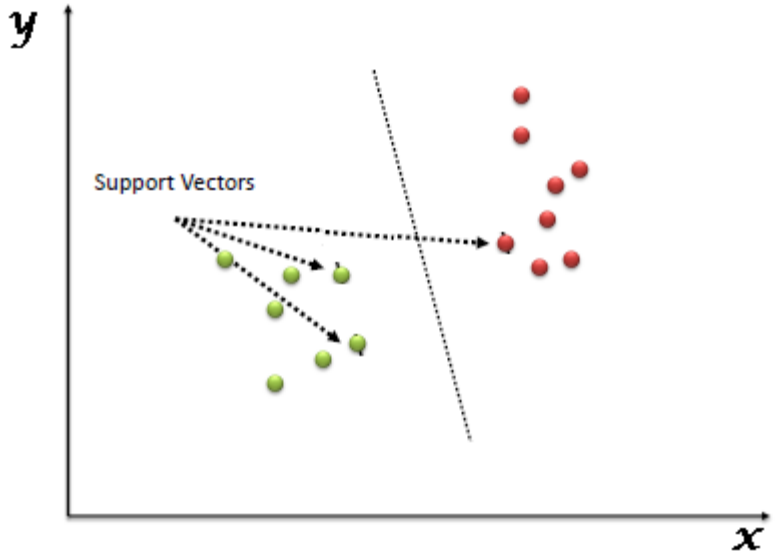
- Of all the points that you try to classify, which are that are most difficult ?



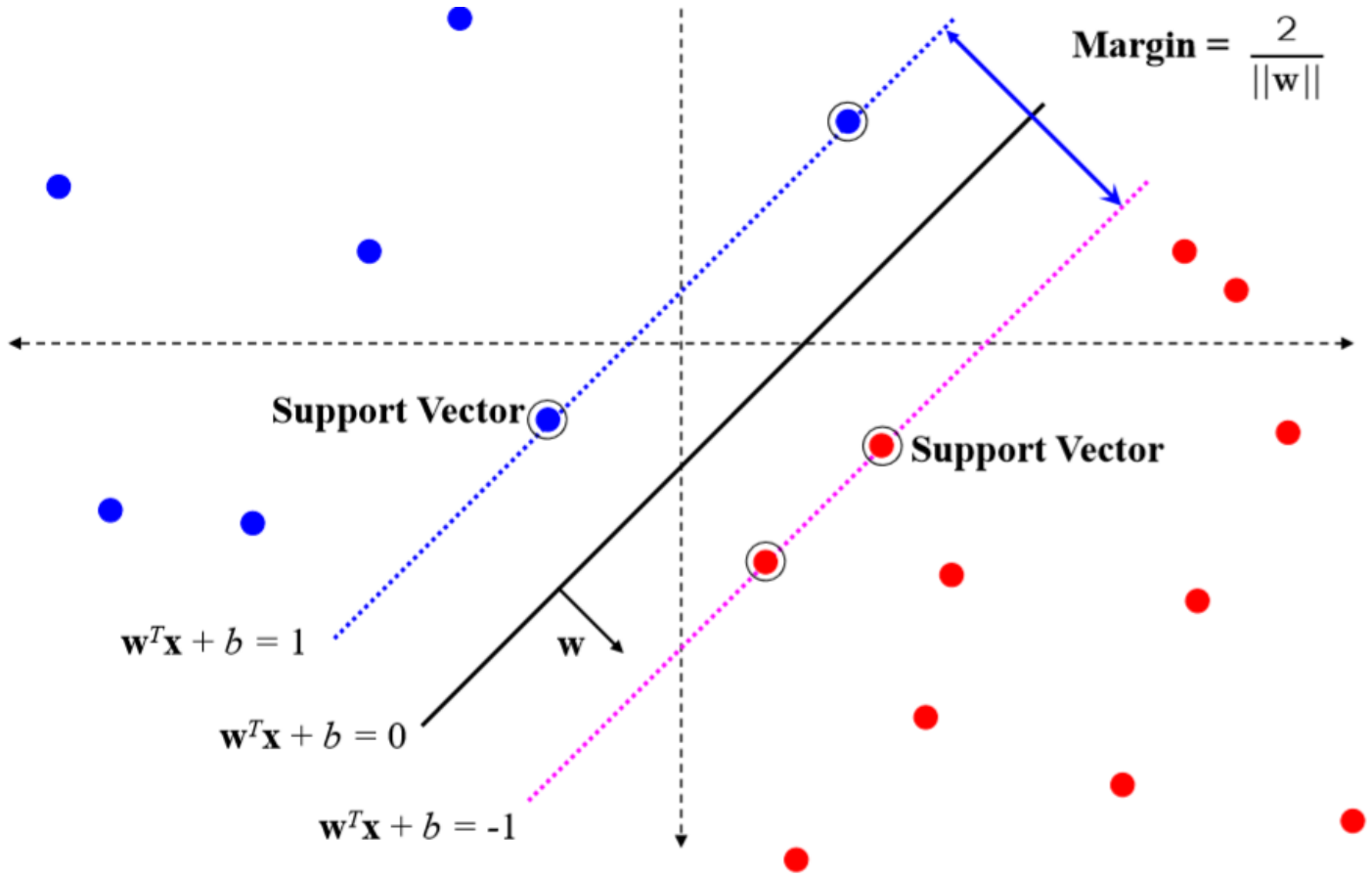
Support Vector Machine

- Of all the points that you try to classify, which are that are most difficult ?
  - The points closest to the decision boundary

- These Points are called support vectors. This have a direct influence on where the decision boundary will be located



Support Vector Machine



Support Vector Machine

- The final optimization of SVM becomes

$$\max_{\mathbf{w}} \frac{2}{||\mathbf{w}||} \text{ subject to } \begin{matrix} \mathbf{w}^\top \mathbf{x}_i + b \geq 1 & \text{if } y_i = +1 \\ \leq -1 & \text{if } y_i = -1 \end{matrix} \text{ for } i = 1 \ldots N$$

- This can be written more compactly as

$$\min_{\mathbf{w}} ||\mathbf{w}||^2 \text{ subject to } y_i \left(\mathbf{w}^\top \mathbf{x}_i + b \right) \geq 1 \text{ for } i = 1 \ldots N$$

Support Vector Machine

- The final loss function would be

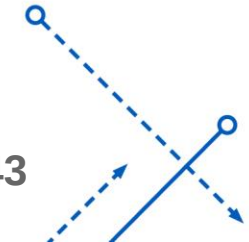
$$\min_{\mathbf{w}\in\mathbb{R}^d} \underbrace{||\mathbf{w}||^2}_{\text{regularization}} + C\sum_i^N \underbrace{\max(0, 1 - y_i f(\mathbf{x}_i))}_{\text{loss function}}$$

- Where $f(x_i)$ is $\mathbf{W^T X}$. The formulation is the reason why SVM losses are also called hinge loss

Support Vector Machine

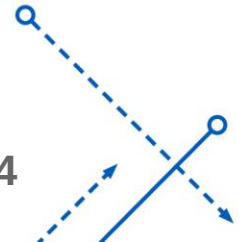
- If data is linearly separable them SVM a unique global minimum value

- What if the data is not linearly separable ?



Support Vector Machine

- If data is linearly separable them SVM a unique global minimum value
- What if the data is not linearly separable ?
  - Non linearity is introduced function ϕ
- Use function ϕ to map the existing data points into a higher dimensional space and try to find a linear separation in that dimension
- But calculating $\phi(X)$ for all the points in the dataset would be computationally inefficient.

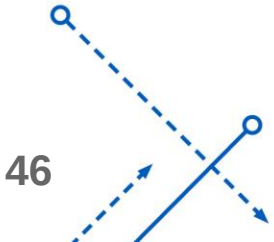


Support Vector Machine

- Here we use the ‘Kernel trick’.
- Kernels are similarity functions that returns the inner product between the images of the data point
- Kernels can often be computed efficiently even for very high dimensional spaces
- No need to explicitly map the data to the feature space
- Most commonly used kernels are Polynomial and RBF

Decision Trees

- The Decision Tree classifier is a widely used classification technique.
- A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails)
- Each branch represents the outcome of the test, and each leaf node represents a class label
- The paths from root to leaf represent classification rules
- Tree models where the target variable can take a discrete set of values are called classification trees and Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees.



Decision Trees

- Consider a simple classification task of vertebrate classification problem. Let us consider only two classes, mammals and non mammals

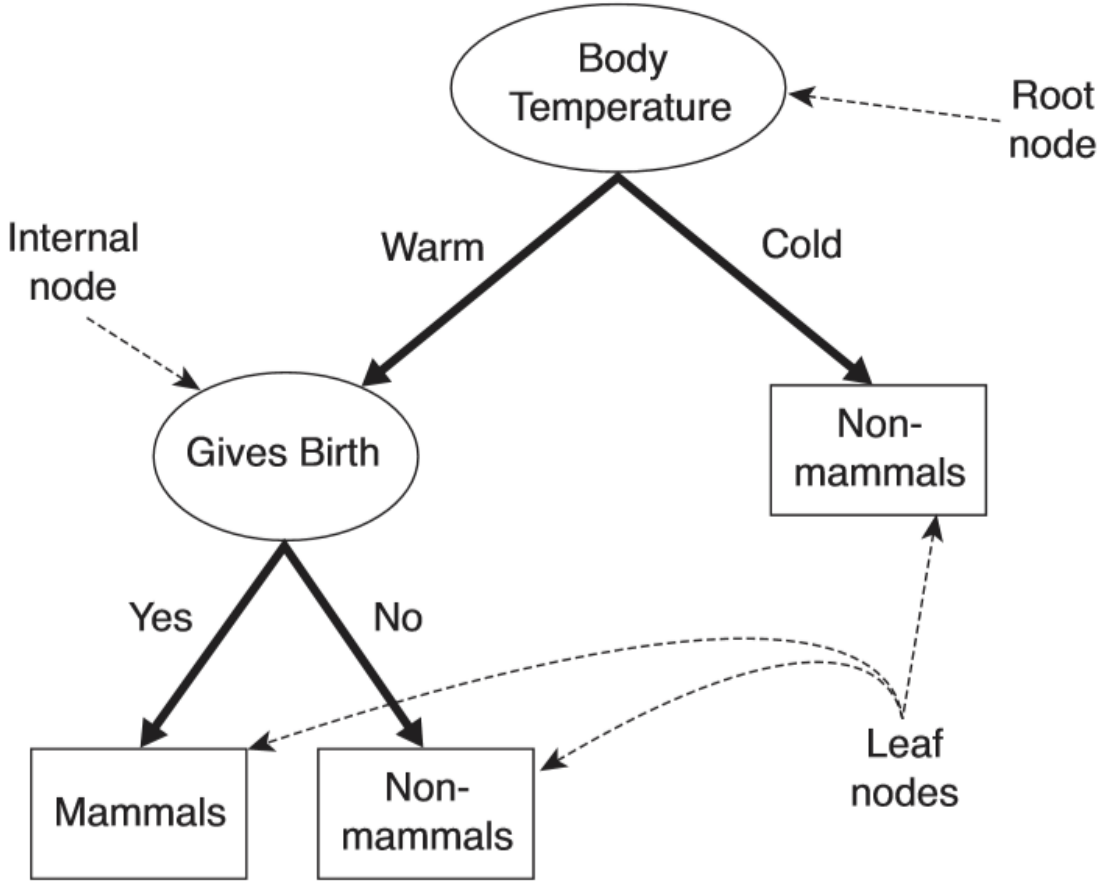
- One way of doing this is to pose a series of questions about the characteristic of the species

- The questions can be:
  - Whether the species are warm or cold blooded ?
  - Do females of that species give birth ?

- Newer questions are asked as a follow up to answers to the previous question until we reach a conclusion to the class label.

Decision Trees

- A root node has no incoming edges and zero or more outgoing edges
- Internal nodes have one incoming node and multiple out going edges
- Leaf nodes have one incoming nodes and no outgoing edges
- Once the tree is constructed, given a species of vertebrate, it can be classified as mammals or non mammals



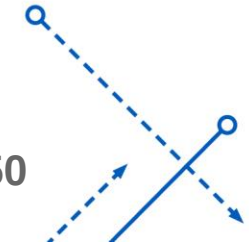
Decision Trees

- So given set of attributes how many such decision trees can be created ?

Decision Trees

- So given set of attributes how many such decision trees can be created ?
  - Many

- If many such trees are possible, are all the trees equally accurate ?



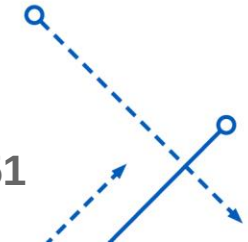
Decision Trees

- So given set of attributes how many such decision trees can be created ?
  - Many

- If many such trees are possible, are all the trees equally accurate ?
  - No, some of the trees are more accurate than others

- Finding the optimal tree is computationally because of the exponential search space

- Efficient algorithms have been developed to create a reasonably accurate decision tree in a sub optimal time

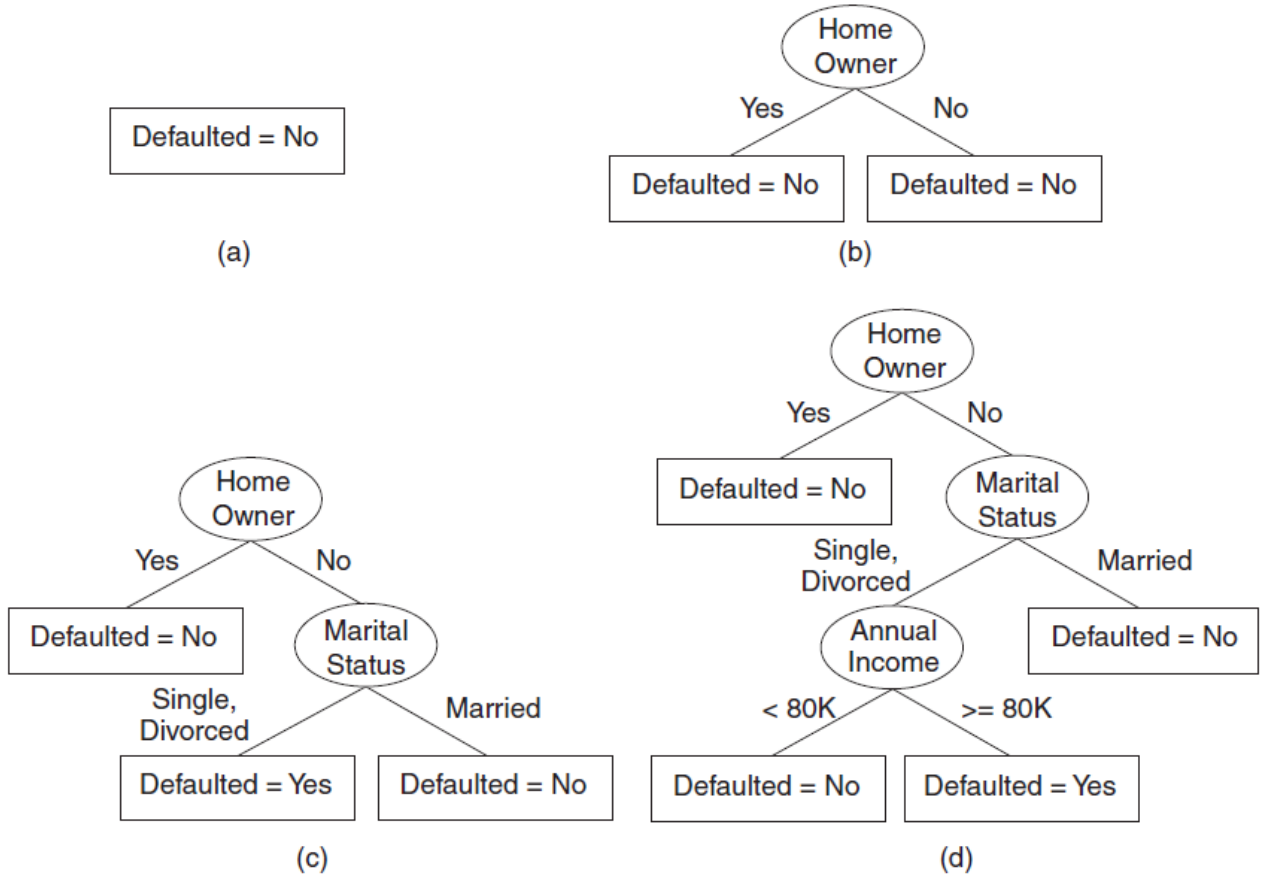


Decision Trees

- Hunt's algorithm is used in many existing decision tree implementation
- A decision tree is grown in a recursive fashion by portioning the training data into successively purer subsets
- Let D_t be the set of training record that are associated with a node t and y = { y_1, y_2, …. y_c} be the class labels
- The recursive definition of Hunt's algorithm is:
  - If all the records in Dt belongs to the same class y_t then t is a leaf node labelled as y_t
  - If D_t has more than one class, then a test attribute condition is selected and the records are portioned into smaller subsets. A child node is created for each outcome of the test condition and records are distributed based on the outcomes. The algorithm is applied recursively to each of the child node.

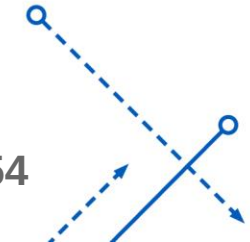
Decision Trees

Tid	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



Measures for Selecting Best Split

- Let P(i|t) be the fraction of records belonging to class i at given node t.
- In a two class problem, the class distribution of any class can be written as (P_0,P_1). Where P_1=1-P_0
- The measure of selecting the best split is based on measuring the impurity. Smaller the degree of impurity more skewed the distribution
- For example if the node has distribution of [0, 1] then the impurity is zero, whereas a node with uniform class distribution (0.5,0.5) has highest impurity.
- Mainly two measures of impurity are used
  - Gini Impurity
  - Entropy



Measures for Selecting Best Split

- Gini impurity is given by

$$\text{Gini}(t) = 1 - \sum_{i=0}^{c-1} [p(i|t)]^2,$$

- Entropy based impurity is given by

$$\text{Entropy}(t) = -\sum_{i=0}^{c-1} p(i|t) \log_2 p(i|t),$$

- C represents the number of different classes and $0\log_2 0 = 0$ for entropy calculations

Measures for Selecting Best Split

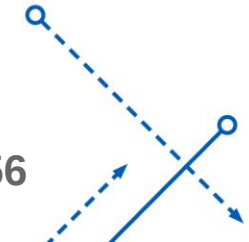
- To determine the best split condition we need to compare the degree of impurity of the parent node (before splitting) to the degree of impurity of the child nodes (after splitting)

- The larger the difference, better the condition. The gain Δ is given by:

$$\Delta = I(\text{parent}) - \sum_{j=1}^{k} \frac{N(v_j)}{N} I(v_j),$$

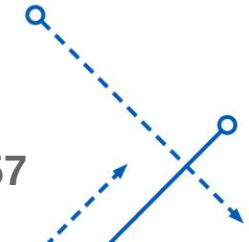
- $N(v_j)$ is the number of records associated with child. $I(v_j)$ impurity associated with child.

- Since impurity of the parent is the same, it is like minimizing the weighted average impurity measure across child nodes



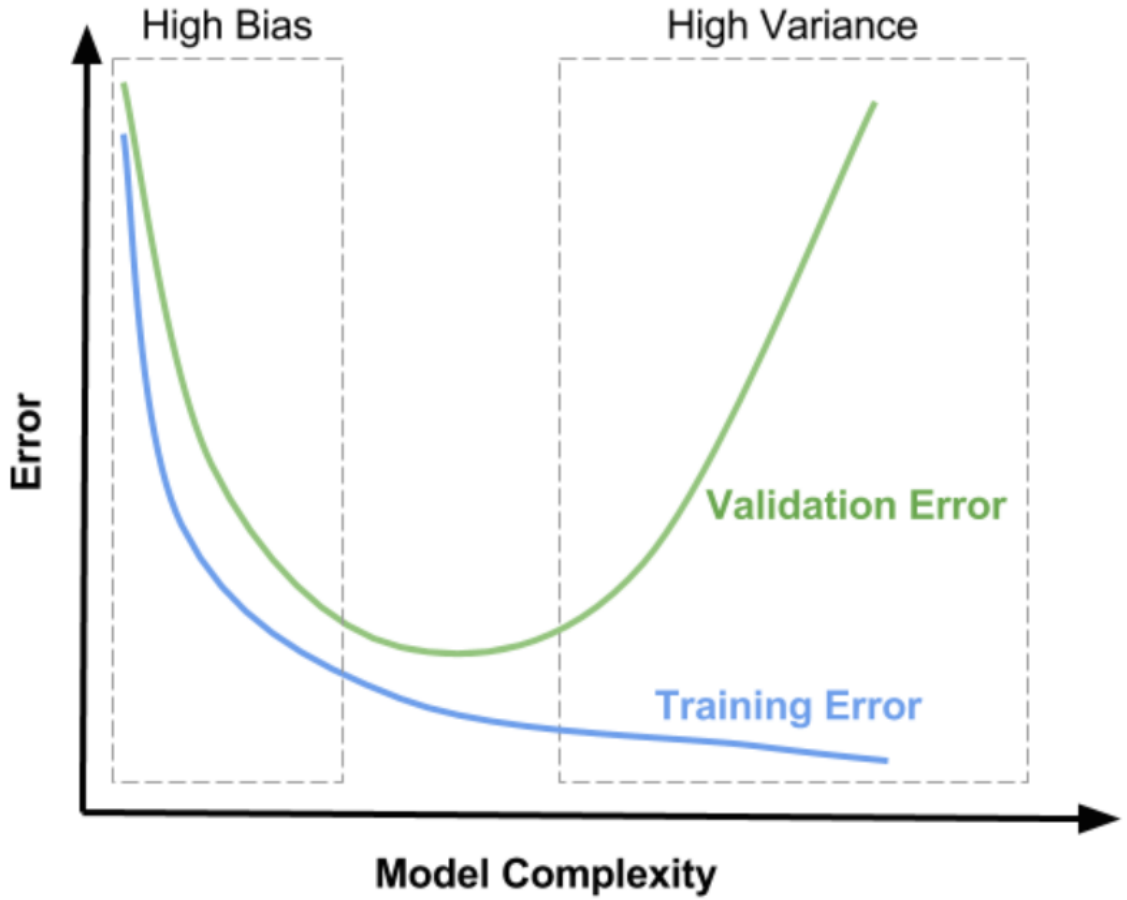
Decision Trees

- If the Decision Tree is not too deep, then it is easy to interpret the classifier.
- Decision Tree can handle both categorical and continuous data
- Constructing decision trees are computationally inexpensive, making it possible to quickly construct a model when training data size is very large. Once the tree is build the classification of the test record is extremely fast
- If the size of the tree is high then it generally overfits to the data. This is why generally decision tree does not generalize well
- In general decision tree have high variance



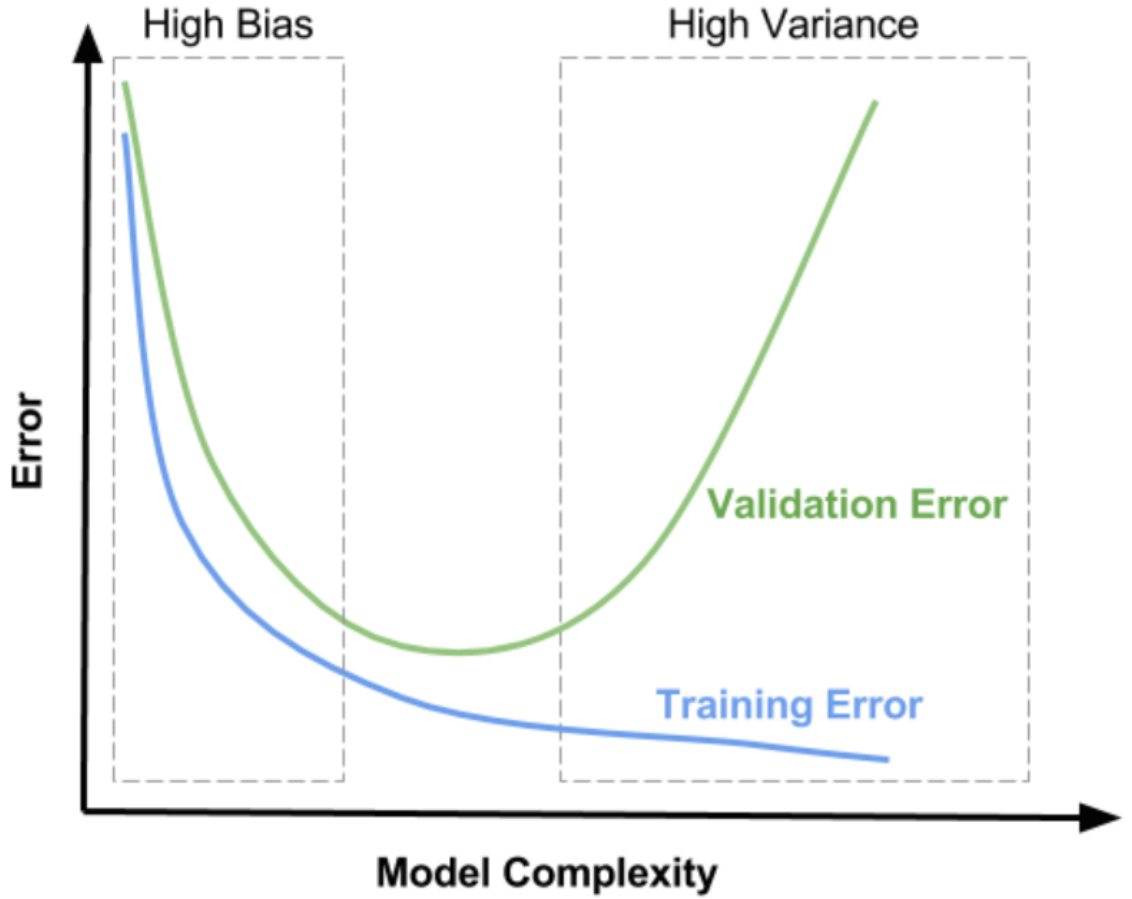
Bias vs Variance

- As model complexity increases the training error reduces
- This is because of overfitting to training data.
- Due to overfitting, the model has high variance and does not generalize well to the test data
- On the other hand models with low complexity, has high training error due to under fitting
- Due to under fitting on the training data it leads to high



Bias vs Variance

- Variance of decision trees increase as the size of the tree increases
- Decision tree which has zero training error has high variance and does not generalize well
- Sometimes in order to restrict the high variance the size of the tree is restricted to fixed value
- But this is may not be an ideal way to reduce the variance of the model



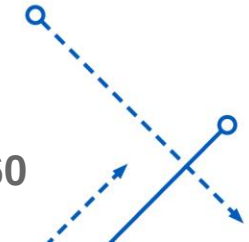
Ensemble Models

- Ensemble learning is the process of using multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone

- Methods used for ensemble learning can be classified in two
  - Bagging
  - Boosting

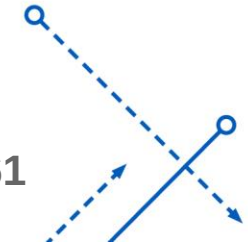
- Bagging is the ensemble learning method in which multiple weak classifier are combined for reducing the high variance of the individual classifier without affecting the bias

- Random Forest algorithm is a modified version of bagging applied to decision trees



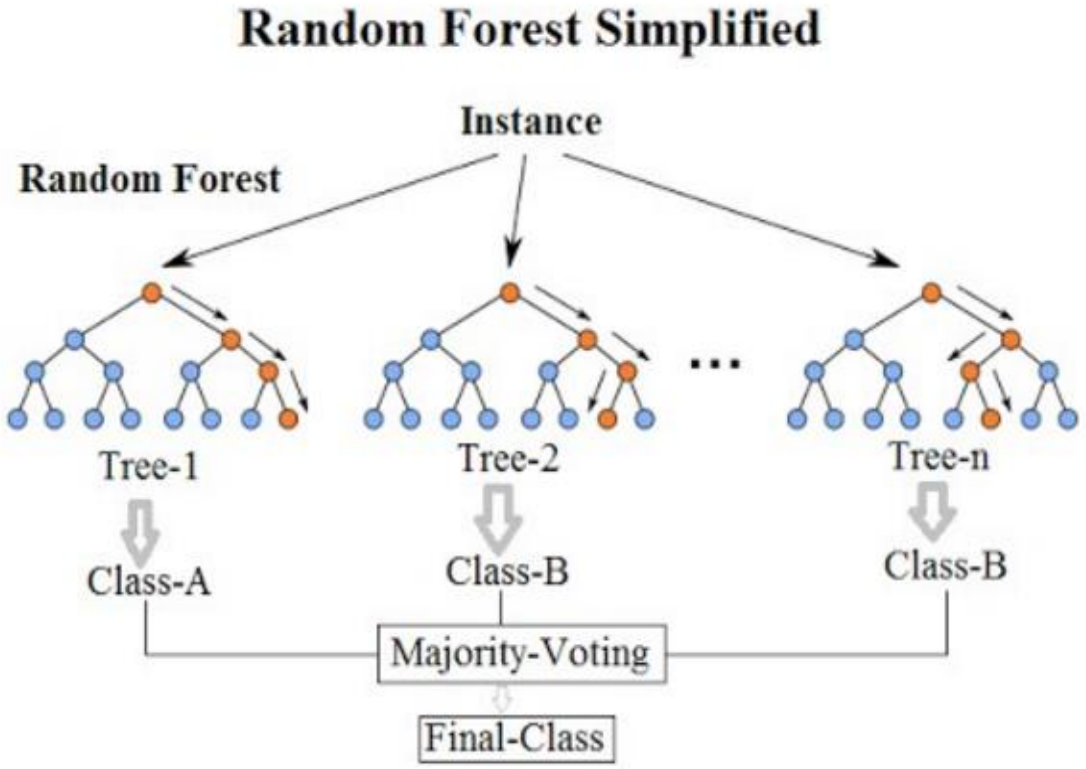
Bagging

- Bagging or bootstrap aggregation, averages a given procedure over many samples, to reduce its variance
- If C(D) be a classifier trained on Dataset D. Let y be the prediction of the classifier on the input x
- In order to create a bagging algorithm, ‘M’ bootstraps of the dataset D are created.
- Each of the ‘M’ datasets is of size N same as that of the original dataset D and is created by random sampling(with replacement) the original dataset D
- A new classifier is learned on each of this M datasets and a majority vote is taken on the prediction of these M new classifier.



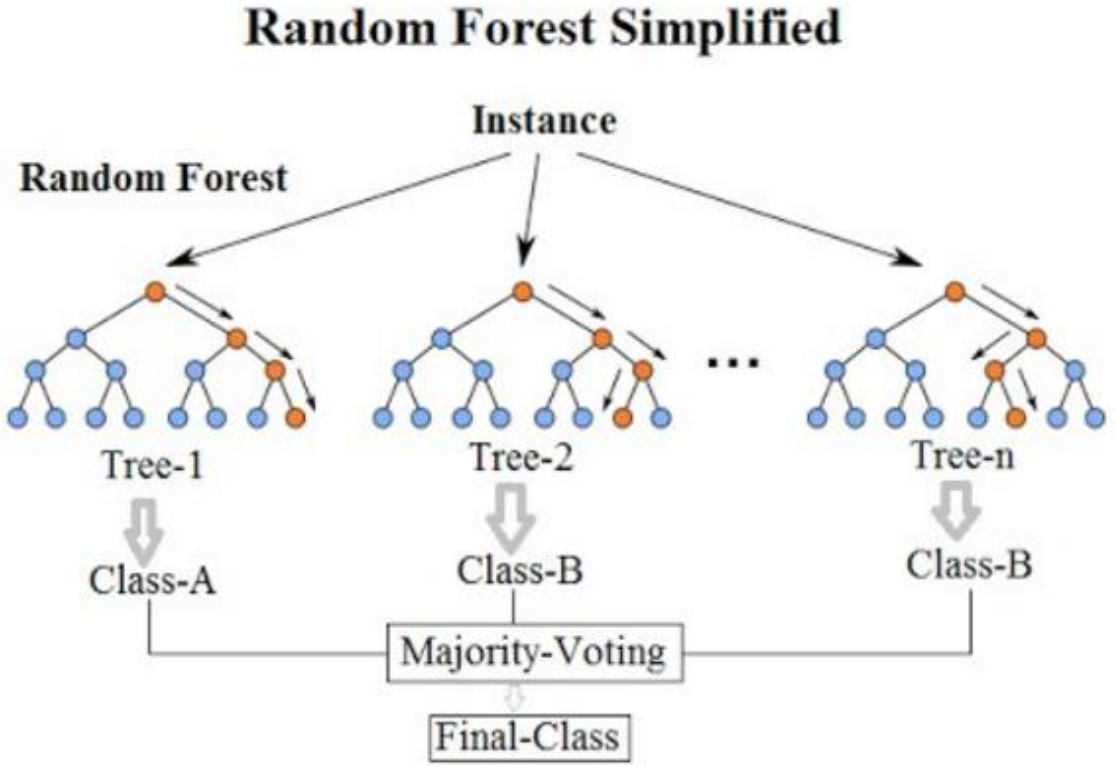
Random Forest

- A modified bagging approach is used to create a Random Forest Classifier.
- M datasets are created in the same bootstrap manner discussed earlier
- At each tree split, a random sample of p features is drawn, only those p features are considered for splitting
- Typically p is set to value around the square root of the number of dimensions d



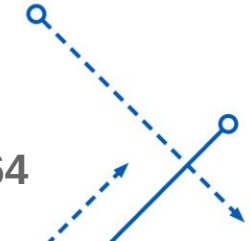
Random Forest

- This will lead to creation of very different decision tree classifiers
- Each tree classifier would have seen different combination of the data points from the dataset
- Each individual classifier is overfit, with high variance
- Final prediction is decided by majority voting on the m different classifiers, which averages out the predictions and reducing the variance



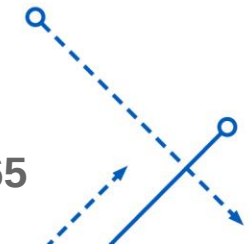
Boosting

- Consider a binary classification problem. Let the two classes be [-1,1].
- Lets try an experiment. Toss an unbiased coin for every data point in the training set and the classes of the data points are decided based on the outcome of the toss.
- This ideally would result in a classification accuracy of 0.5
- Any machine learning classifier which performs at least slightly better than the coin toss model is called as a weak classifier.
- Can we use multiple such weak classifier to create a strong classifier ?



Boosting

- Instead of learning a single (weak) classifier, learn many weak classifiers that are good at different parts of the input space
- Take a weighted vote of the classifiers. The average prediction obtained by taking the vote would be better than an individual classifier
- How to make the classifiers learn different part of the input space ?
- In Bagging we were giving equal weights to the classifier prediction. How do we weight the contribution of each classifier ?



AdaBoost

- AdaBoost is an algorithm for constructing a ”strong” classifier as linear combination of week classifiers

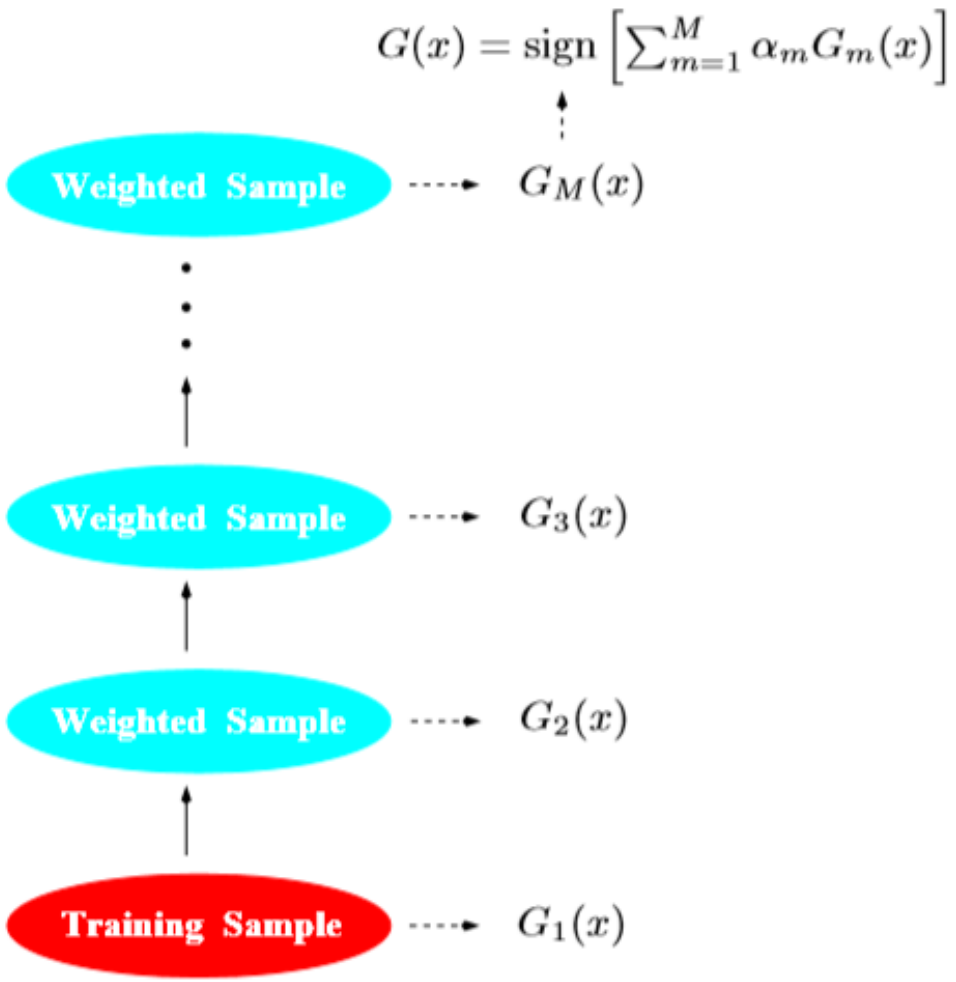
$$f(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$$

- Here $h_t(x)$ is a weak classifiers. $h_t(x)$ is some time referred as G(x)

- H(x) = sign(f(x)) ‘’strong” or final classifier/hypothesis

- This is because we are considering a binary classification problem with [-1 and +1] as classes. So the sign of f(x) will determine the class

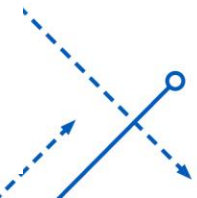
AdaBoost


$G(x) = \text{sign}\left[\sum_{m=1}^{M} \alpha_m G_m(x)\right]$
Weighted Sample
$G_M(x)$
Weighted Sample
$G_3(x)$
Weighted Sample
$G_2(x)$
Training Sample
$G_1(x)$

AdaBoost

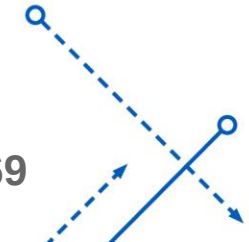
AdaBoost$(S=((x_1,y_1),\ldots,(x_m,y_m)))$

1 **for** $i \leftarrow 1$ **to** m **do**
2 $\quad D_1(i) \leftarrow \frac{1}{m}$
3 **for** $t \leftarrow 1$ **to** T **do**
4 $\quad h_t \leftarrow$ base classifier in H with small error $\epsilon_t = \Pr_{i\sim D_t}[h_t(x_i) \neq y_i]$
5 $\quad \alpha_t \leftarrow \frac{1}{2}\log\frac{1-\epsilon_t}{\epsilon_t}$
6 $\quad Z_t \leftarrow 2[\epsilon_t(1-\epsilon_t)]^{\frac{1}{2}}$ $\quad \triangleright$ normalization factor
7 $\quad$ **for** $i \leftarrow 1$ **to** m **do**
8 $\quad\quad D_{t+1}(i) \leftarrow \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$
9 $\quad f_t \leftarrow \sum_{s=1}^{t}\alpha_s h_s$
10 **return** $h = \operatorname{sgn}(f_T)$



AdaBoost

- D_t also referred to as W_t are the weights associated with each sample.
- Each round the weight of the misclassified items are increased.
- The standard practice is to use decision trees, quite often just decision stumps (trees of depth one)
- Robust to overfitting
- Test set error decreases even after training error is zero



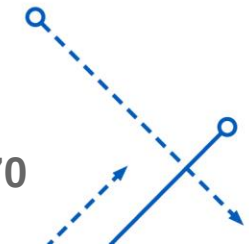
Unsupervised Learning

- Unsupervised learning is a type of machine learning that looks for previously undetected patterns in a data set with no pre-existing labels

- Two of the main methods used in unsupervised learning are
  - cluster analysis
  - principal component

- Useful to detect patterns like customer shopping patterns, regions in an image, market segmentations etc

- Unsupervised methods are used when labels associated with corresponding features are not available. Majority of the data obtained from the real world does not have categories or other supervision signal associated with them



Clustering

- A cluster is a collection of data items which are “similar” between them, and “dissimilar” to data items in other clusters

- The main idea of clustering is to group similar points into clusters such that
  - High intra-cluster similarity
  - Low inter-cluster similarity

- Two of the clustering methods that we will focus on
  - K-means Clustering
  - Hierarchical Clustering

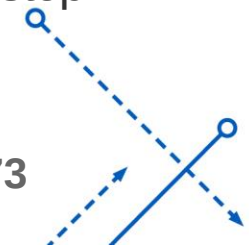
- Clustering is done as a stand-alone tool to get insight into data distribution or as a preprocess step for another algorithm

K-means Clustering

- The aim of K-means clustering is to construct a partition of a database D of n objects into a set of K clusters
- Given K, can we optimize the way data is partitioned so that into these K clusters, so that intra cluster similarity is high and inter cluster similarity is low
- K-means algorithms is a heuristic based partitional clustering algorithm.
- Let the set of data points D be $\{x_1, x_2, \ldots, x_n\}$, where each point xi is feature vector of d dimensions
- K-means algorithm divides the points in K clusters, each of this clusters have a center called the centroid
- K is number of clusters and is a user specified value

K-means Clustering

- Choose the number of clusters K
- The centers (centroids/means) are initialized to some value (by some strategy)
- Reassign all the points in the data to the closest centroid thus creating clusters around the centroid
- Recalculate the centroid based on this assignment
- The above reassign-recalculate process continues until there is no change in centroid values, or points stop switching clusters

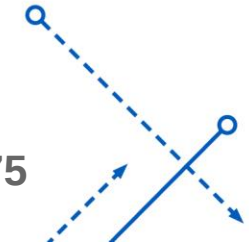


K-means Clustering

for $k = 1, \ldots, K$ let $\mathbf{r}(k)$ be a randomly chosen point from D;
while changes in clusters C_k happen **do**
    form clusters:
    for $k = 1, \ldots, K$ **do**
        $C_k = \{\mathbf{x} \in D \mid d(\mathbf{r}_k, \mathbf{x}) \leq d(\mathbf{r}_j, \mathbf{x})$ for all $j = 1, \ldots, K, j \neq k\}$
    end;
    compute new cluster centers:
    for $k = 1, \ldots, K$ **do**
        $\mathbf{r}_k =$ the vector mean of the points in C_k
    end;
end;

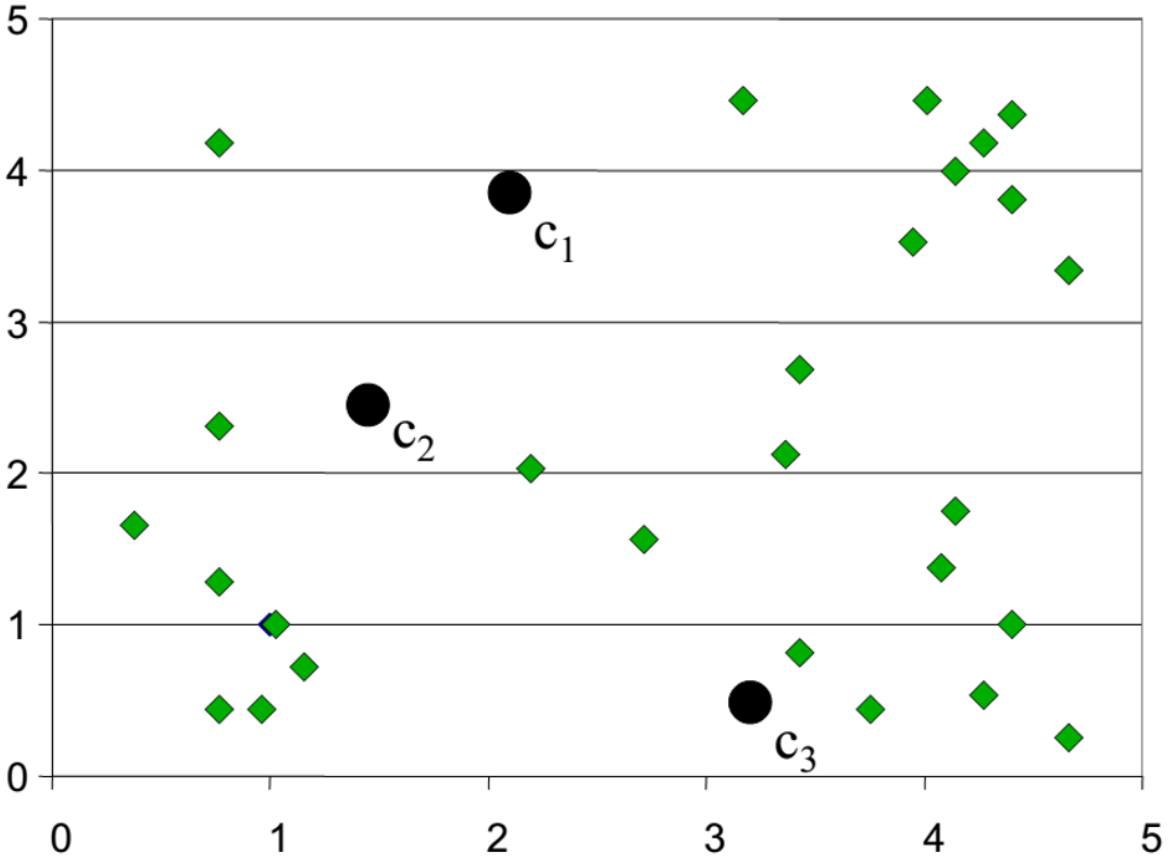
K-means Clustering

- K-means searches for the minimum sum of squares assignment
- It minimizes unnormalized variance (=total_SS) by assigning points to cluster centers
- In order for k-means to converge, two conditions are considered
  - Re-assigning points reduces the sum of squares
  - Re-computing the mean reduces the sum of squares
- As there is only finite number of combinations, you cannot infinitely reduce this value and the algorithm must converge at some point to a local optimum

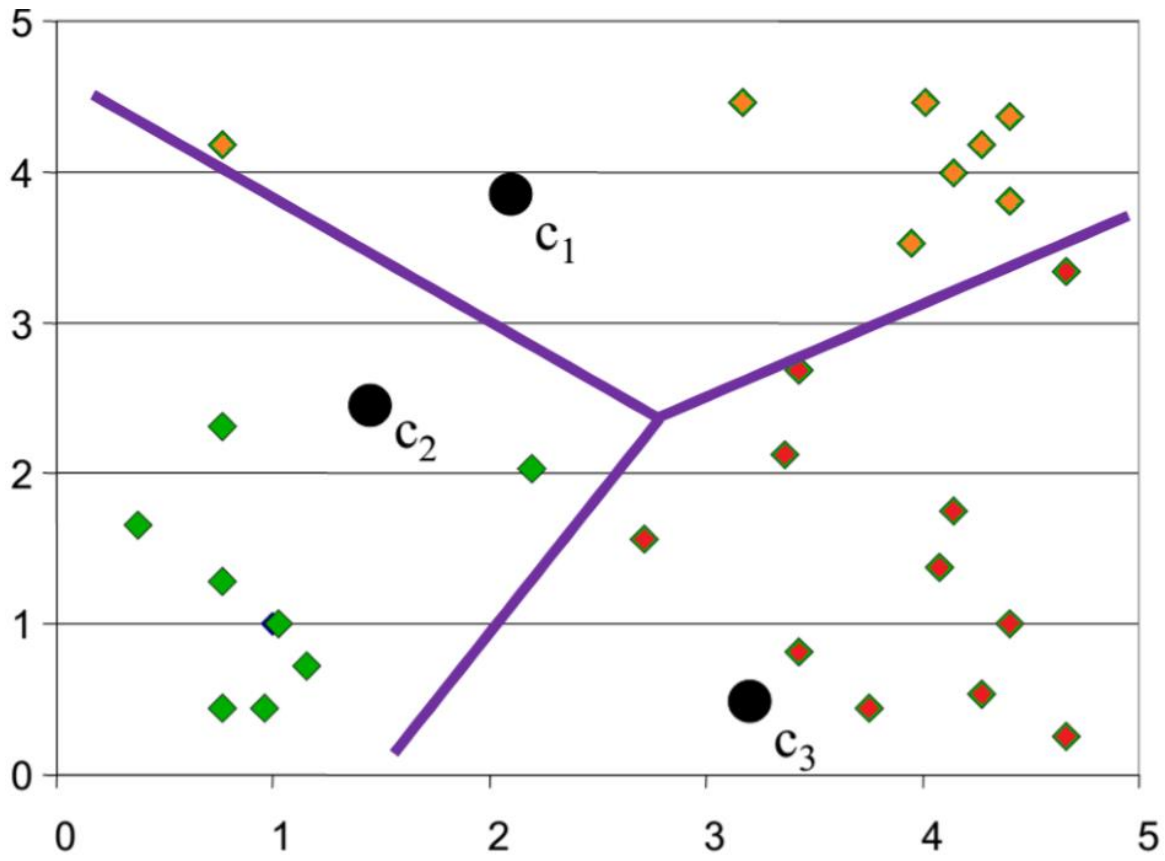


K-means Clustering

Step – 1, Random Initialization

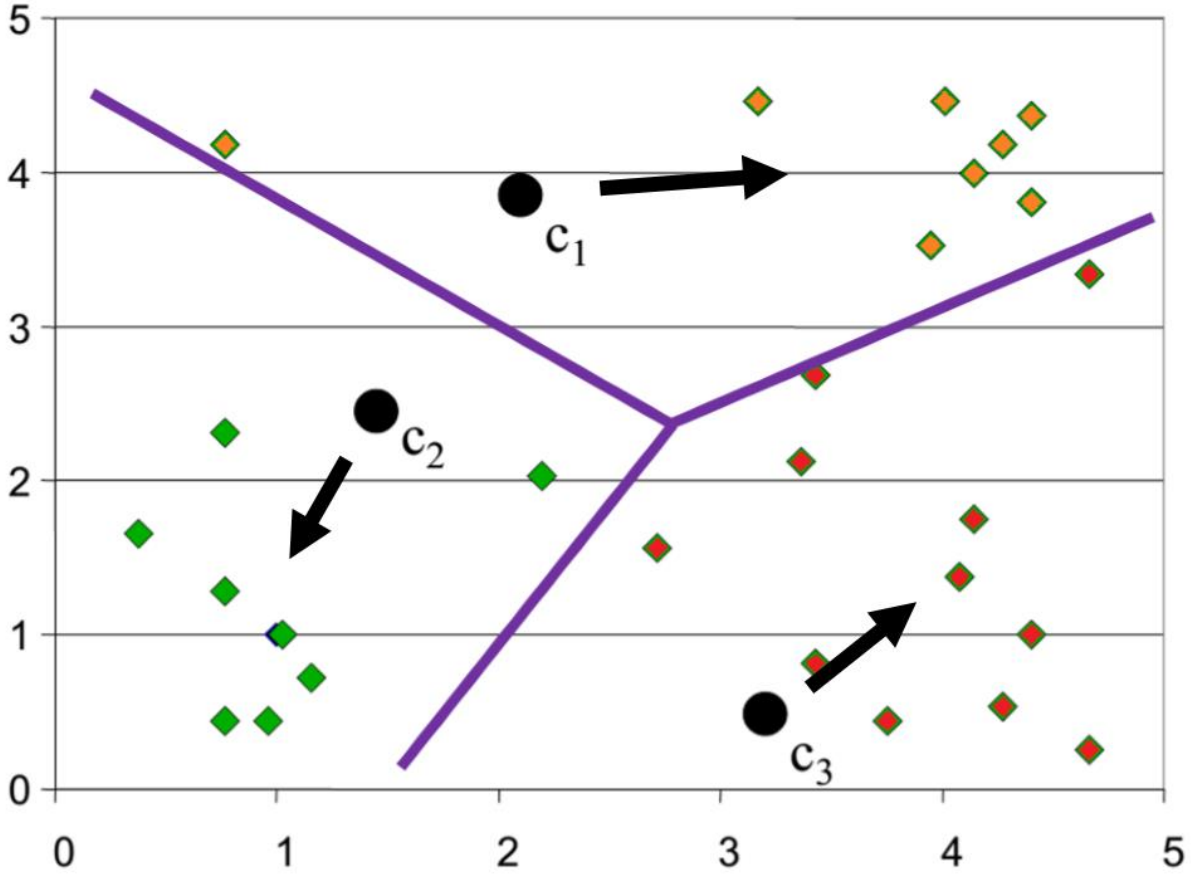


Step – 2, Cluster Membership

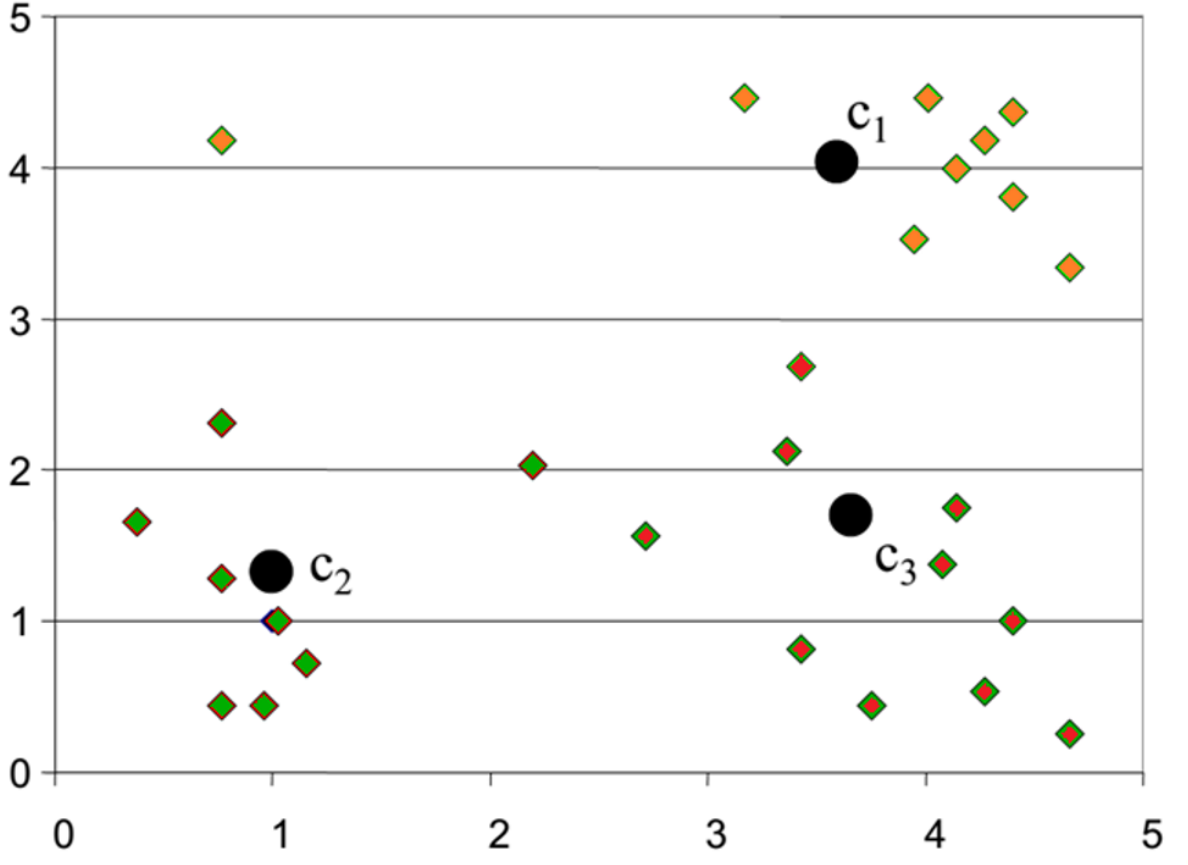


K-means Clustering

Step – 3, Re-estimate cluster center

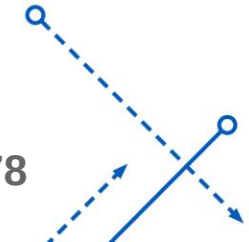


Step – 4, After first iteration



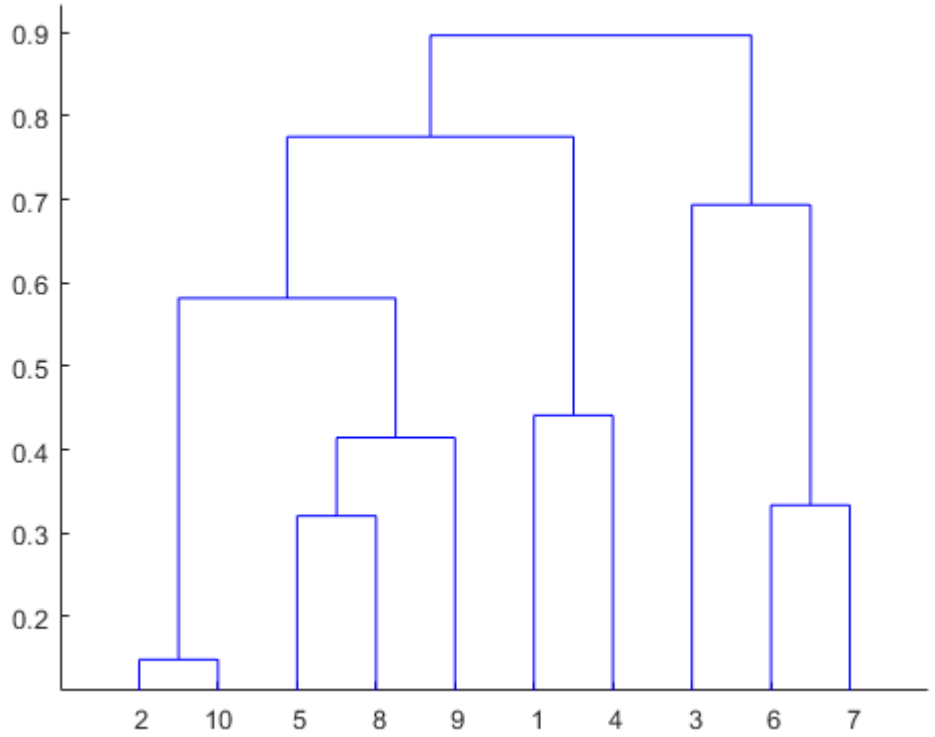
K-means Clustering

- K-means is the most popular clustering algorithm
- Easy to implement and understand
- K-means is sensitive to outliers
- K-means depends on the user specific value of K. The optimal value of K is hard to infer from data
- K-means terminates at a local optimum if SSE is used. The global optimum is hard to find due to complexity



Hierarchical Clustering

- For some data, hierarchical clustering works better compared to flat clustering like K-means.
- The agglomerative hierarchical clustering algorithm builds a cluster hierarchy that is commonly displayed as a tree diagram called a dendrogram
- At each step, the two clusters that are most similar are joined into a single new cluster. Once fused, objects are never separated



Hierarchical Clustering

- Given two clusters of points. The decision of finding closest points in another cluster is done mostly in 3 ways

- Single linkage: the similarity of the closest pair

$$d_{SL}(G,H) = \min_{i \in G, j \in H} d_{i,j}$$

- Complete linkage: the similarity of the furthest pair

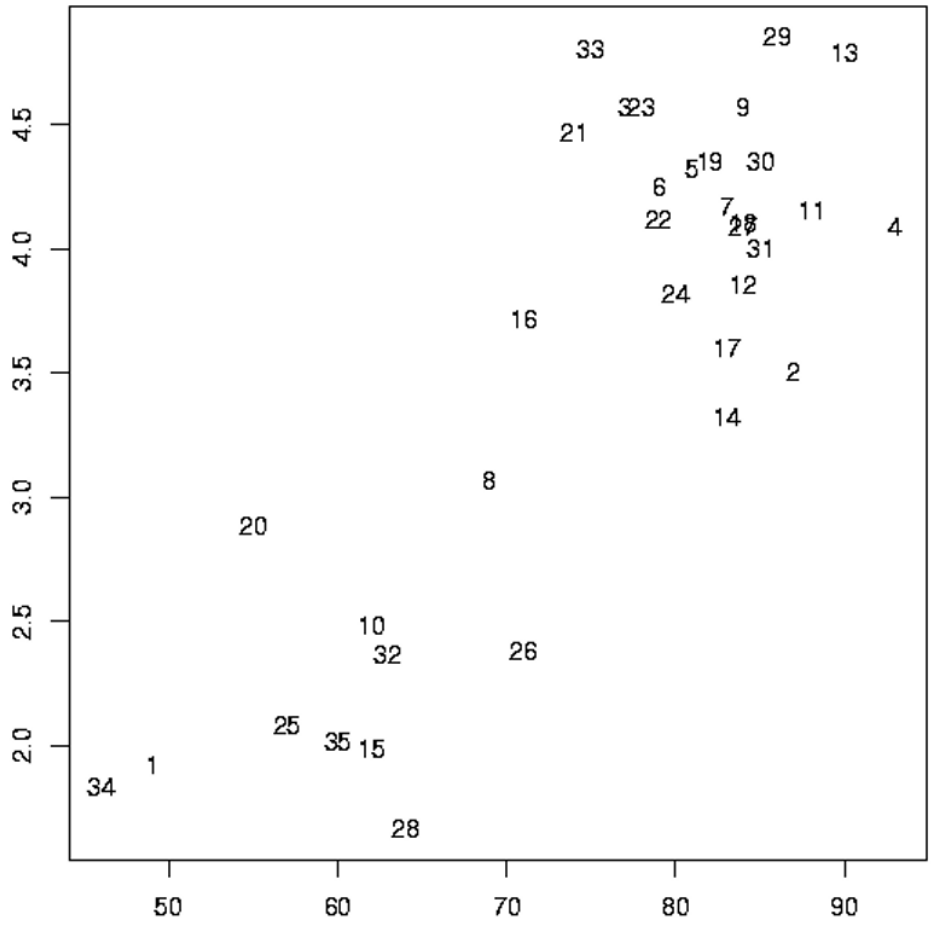
$$d_{CL}(G,H) = \max_{i \in G, j \in H} d_{i,j}$$

- Group-average: the average similarity between groups

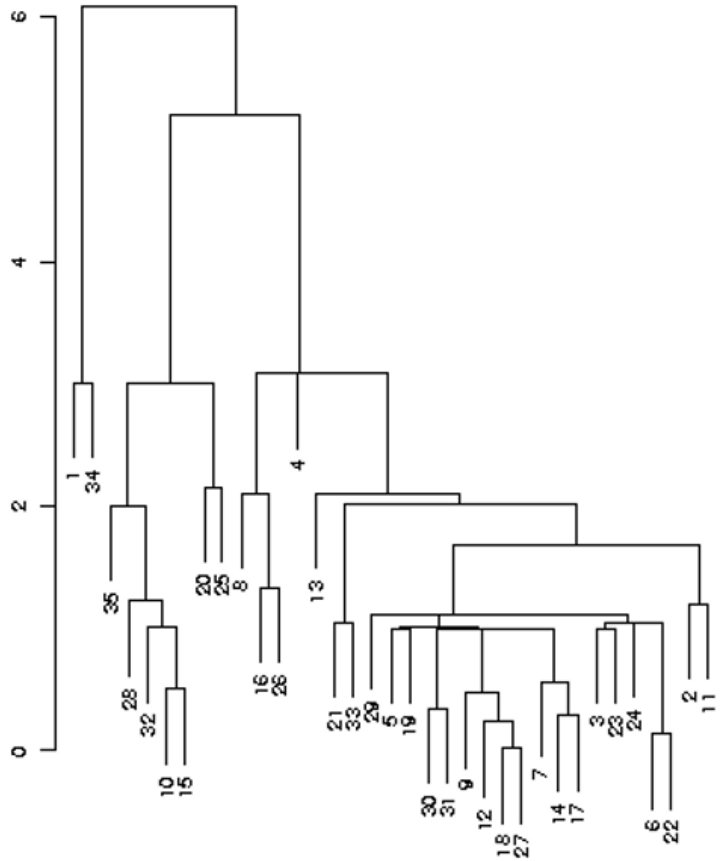
$$d_{GA} = \frac{1}{N_G N_H} \sum_{i \in G} \sum_{j \in H} d_{i,j}$$

Hierarchical Clustering

Initial Points



Dendrogram of hierarchical clustering

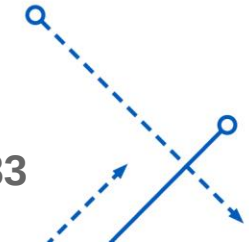


Hierarchical Clustering

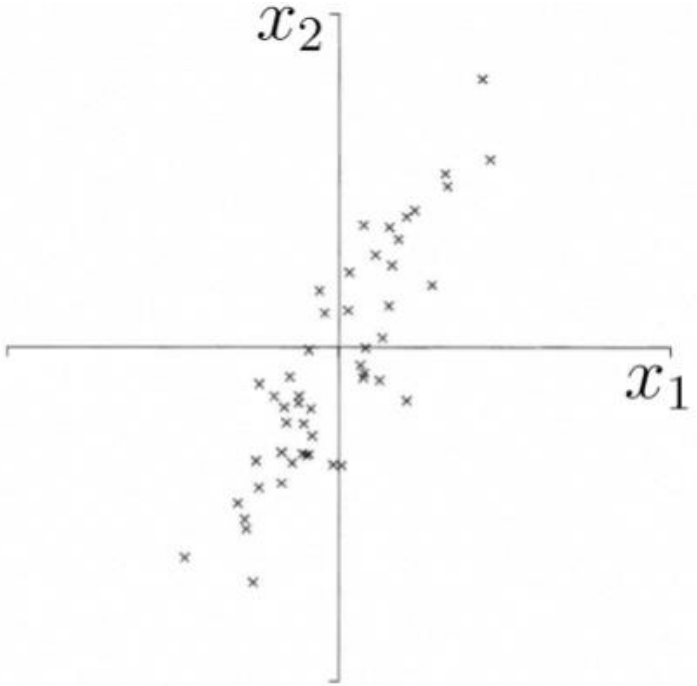
- Agglomerative hierarchical clustering does not require the size of the clusters to be know before hand.
- The number of clusters can be decided based the cohesion of points in the cluster
- Cohesion can be calculated either as
  - Diameter of the new cluster
  - Maximum distance to the centroid
  - Based on Density
- Once a decision is made to combine two clusters, it cannot be undone

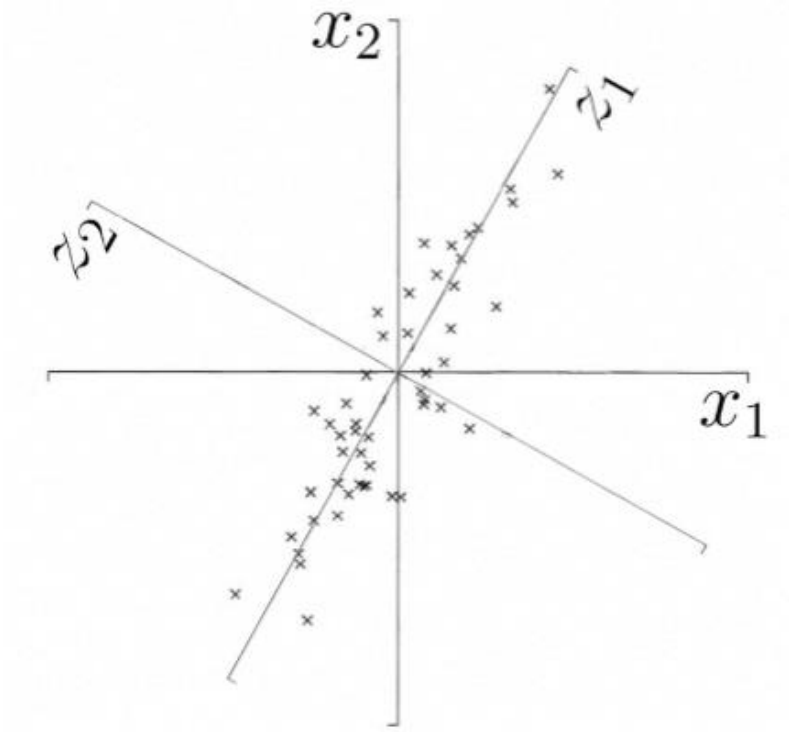
Principal Component Analysis

- Principal component analysis (PCA) is a technique that is useful for the compression of data
- The purpose is to reduce the dimensionality of a data set (sample) by finding a new set of variables, smaller than the original set of variables, that nonetheless retains most of the sample's information
- By information we mean the variation present in the sample
- The new variables, called principal components (PCs), are uncorrelated, and are ordered by the fraction of the total information each retains.



Principal Component Analysis

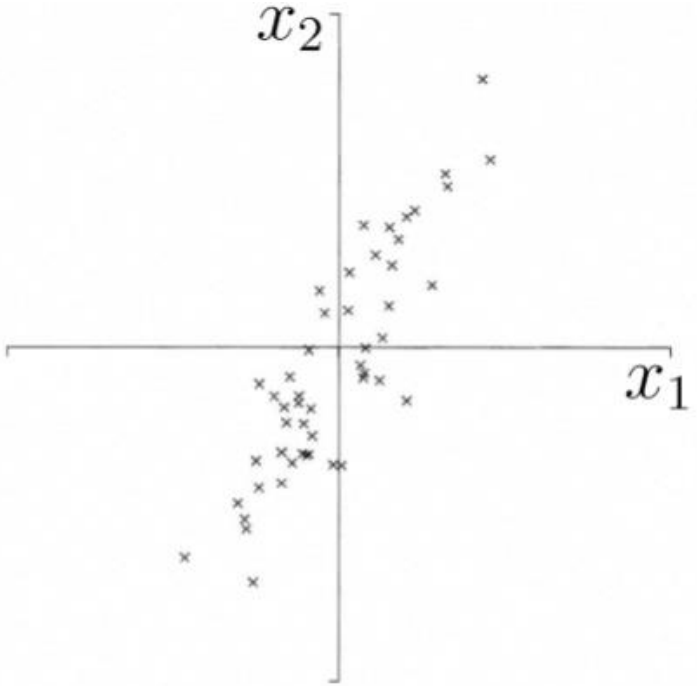


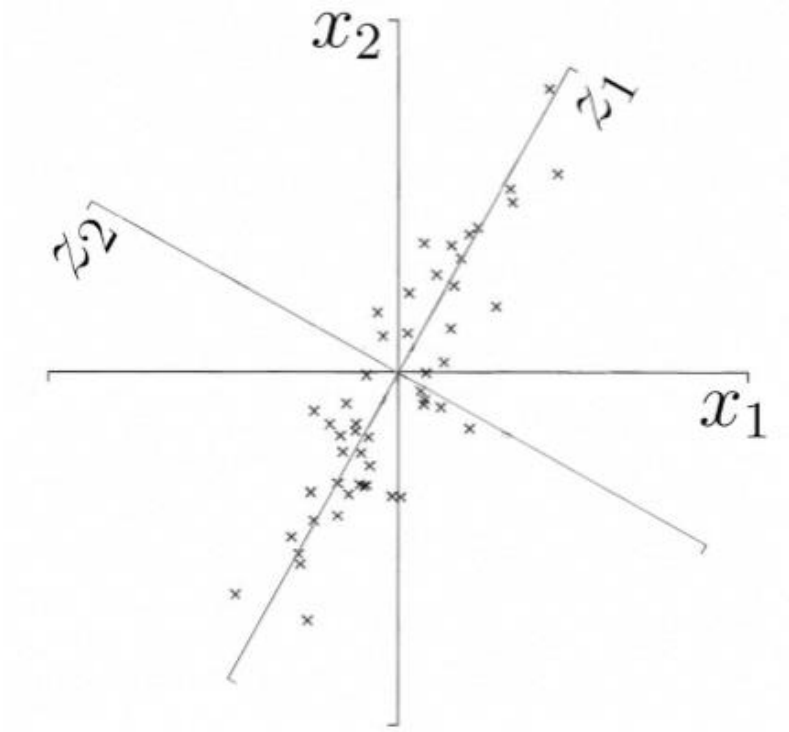


- Let the data under consideration be a set of points in the 2-D space represented by (x_1,x_2)

- The geometric idea of principal component analysis is to come up with a new set of features z_1 and z_2 which capture the variance in the data but are perpendicular to each other

Principal Component Analysis





- PCs are a series of linear least squares fits to a sample, each orthogonal to all the previous

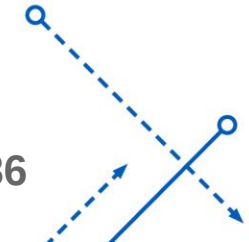
- Z_1 is a least square fit which corresponds to the first principal component, Z_2 corresponds to second principal component which is perpendicular to Z_1

Principal Component Analysis

- Given n observations of each of d dimensions. I.e. (x_1 , x_2 , x_3, …..x_d)
- The first principal component can be defined as

$$z_1 = a_1^T x = \sum_{i=1}^{d} a_{1i} x_i$$

- Where a_1 refers to the vector (a_{11},a_{12},a_{13},……a_{1d}) chosen so as that Var[z_1] is maximum
- So in general $z_k = a_k^T x$ where a_k is chosen such that Var[z_k] is maximum and subjected to the covariance of z_k with z_l where $k > l > 1$ and $a_k^T * a_k = 1$



Principal Component Analysis

1. Compute the mean feature vector

   $\mu = \frac{1}{p}\sum_{k=1}^{p} x_k$, where, x_k is a pattern ($k = 1$ to p), p = number of patterns, x is the feature matrix

2. Find the covariance matrix

   $C = \frac{1}{p}\sum_{k=1}^{p}\{x_k - \mu\}\{x_k - \mu\}^T$ where, T represents matrix transposition

3. Compute Eigen values λ_i and Eigen vectors v_i of covariance matrix

   $Cv_i = \lambda_i v_i$ $(i = 1, 2, 3,....q)$, q = number of features

4. Estimating high-valued Eigen vectors

   (i) Arrange all the Eigen values (λ_i) in descending order

   (ii) Choose a threshold value, θ

   (iii) Number of high-valued λ_i can be chosen so as to satisfy the relationship

   $$\left(\sum_{i=1}^{s}\lambda_i\right)\left(\sum_{i=1}^{q}\lambda_i\right)^{-1} \geq \theta$$, where, s = number of high valued λ_i chosen

   (iv) Select Eigen vectors corresponding to selected high valued λ_i

5. Extract low dimensional feature vectors (principal components) from raw feature matrix.

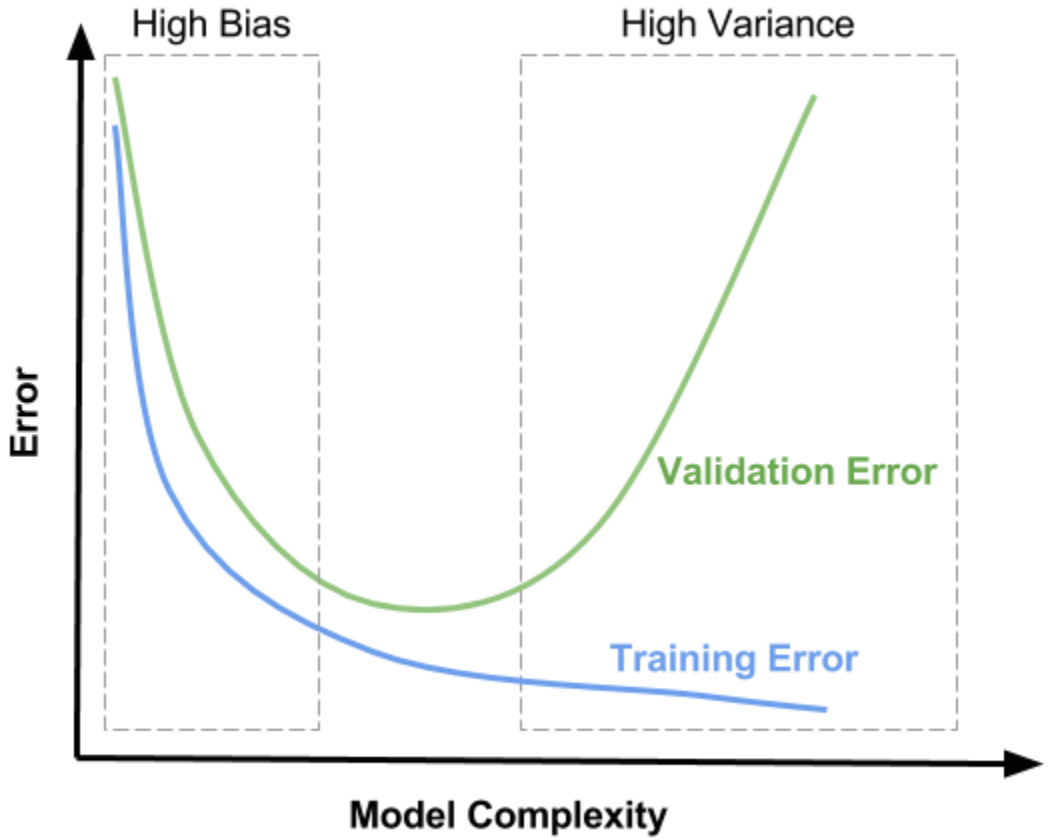
   $P = V^T x$, where, V is the matrix of principal components and x is the feature matrix

SOME DETAILS

University at Buffalo
Department of Computer Science and Engineering
School of Engineering and Applied Sciences

Bias vs Variance Trade off

- Bias vs Variance trade off is important to understand overfitting vs underfitting
- Model should be trained until validation loss reduces as training loss reduces.
- Model is overfitting when training loss reduces but validation loss does not reduce
- Regularizations techniques are used to avoid overfitting

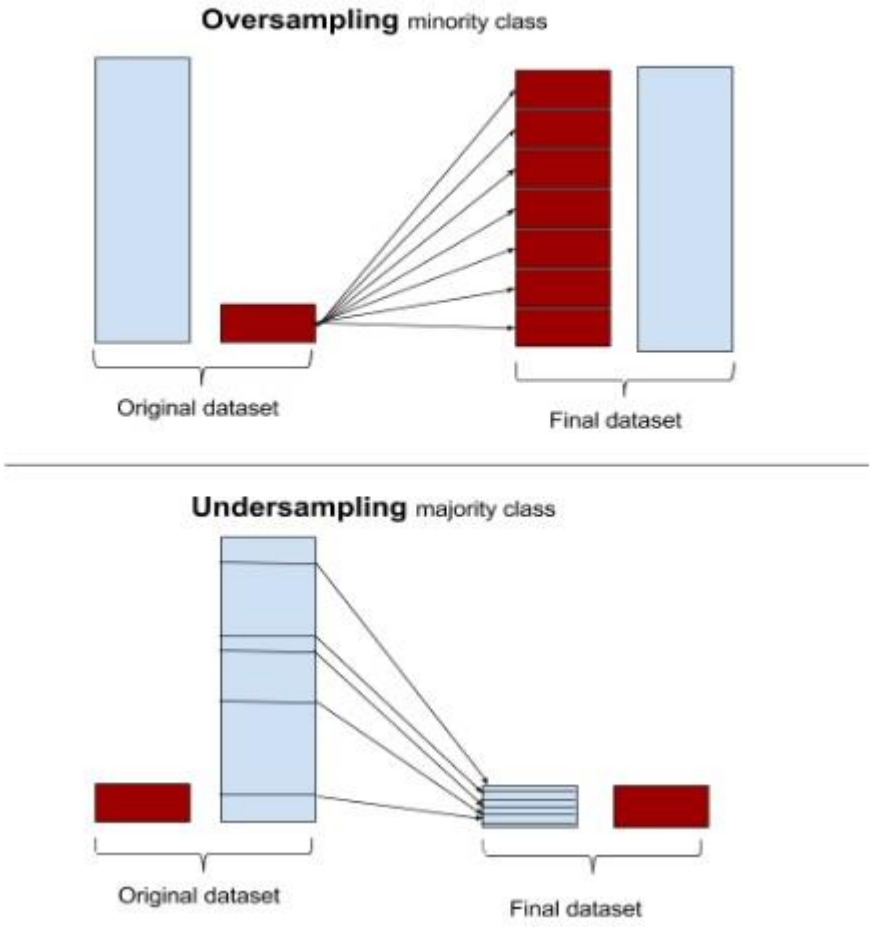


Class Imbalance

- Class imbalance problems occur when some classes in the dataset have much more data samples than other classes

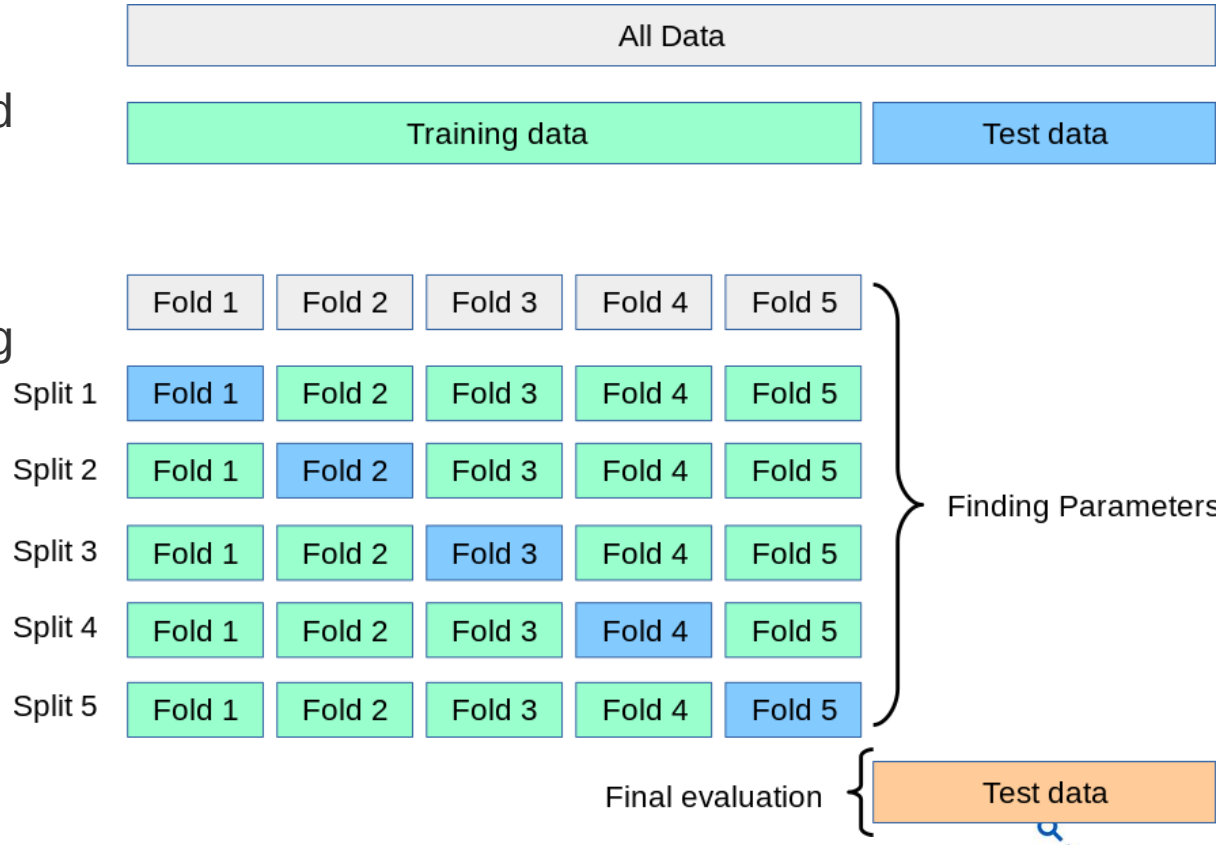
- The methods to address class imbalance are
  - Oversampling
  - Undersampling
  - Class weighting

- Most of the Sklearn classification function accept class weights as a input argument.

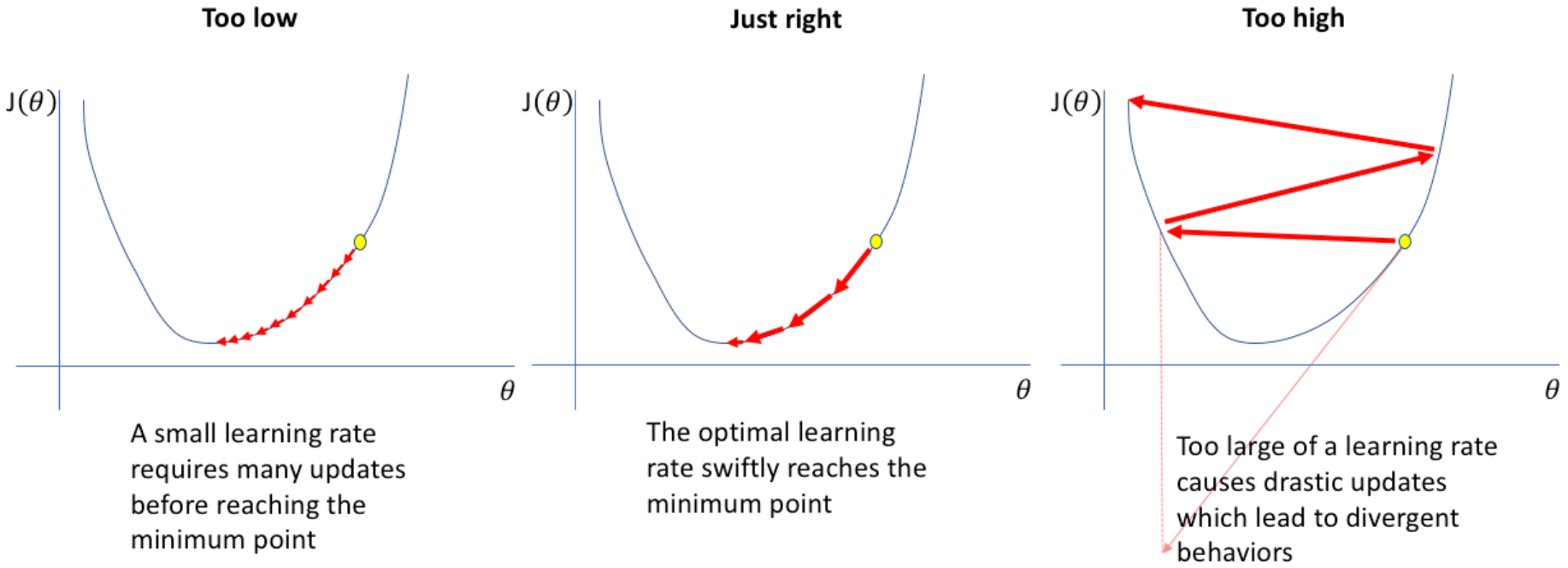


K-Fold Cross Validation

- General practice to split data into train, validation and test splits
- K-fold cross validation is a approach in which training data is divided into K different splits.
- The model is trained on K-1 splits and validated on the Kth split.
- Average out the performance of models created in k splits on the testing data.



Learning rate decay



- The way in which the learning rate changes over time is referred to as the learning rate schedule or learning rate decay
- Initially you start of with a higher learning rate and you reduce the rate as more iteration on your training data
- Multiple methods of reduction available in sklearn

KNN Implementation

- Imagine the train matrix is A and test matrix as B
- The objective is to compute what distance of each row of the test data to that of training data
- Using for loops to do this would take around 30 mins
- Since A and B have same dimension, can we use this to our advantage
- So can we use the expansion of $(A-B)^2$

Confusion matrix

- Iterative way of computing confusion matrix might be slower
- Vectorized approach is much more faster but less intuitive
- The precision and recall has to be computed per class
- The average the precision of each class to get macro precision, similar with recall

		True/Actual		
		Cat (🐱)	Fish (🐟)	Hen (🐔)
Predicted	Cat (🐱)	4	6	3
	Fish (🐟)	1	2	0
	Hen (🐔)	1	2	6

Confusion matrix

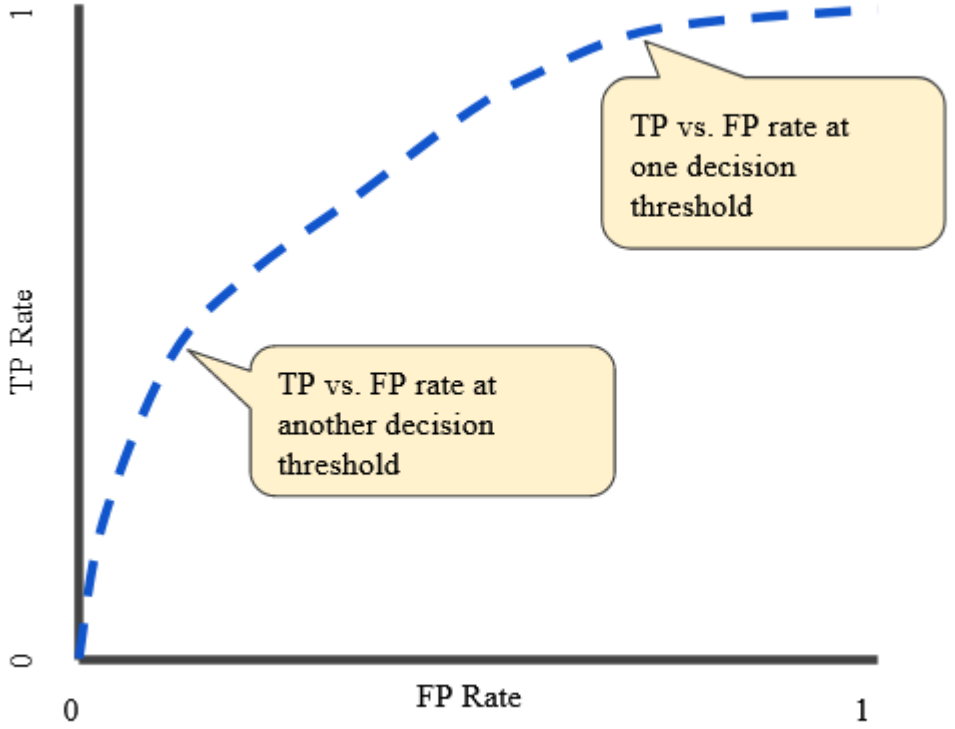
- True Positive Rate of a Classifier

$$TPR = \frac{TP}{TP + FN}$$

- False Positive Rate of a Classifier

$$FPR = \frac{FP}{FP + TN}$$

- What is an ROC Curve ?



References

- https://www-users.cs.umn.edu/~kumar001/dmbook/ch4.pdf
- http://web.mit.edu/6.034/wwwbob/svm-notes-long-08.pdf
- https://www.cc.gatech.edu/~hic/CS7616/pdf/lecture5.pdf
- https://www.youtube.com/watch?v=a3ioGSwfVpE
- https://www.learnopencv.com/bias-variance-tradeoff-in-machine-learning
- https://web.stanford.edu/~hastie/TALKS/boost.pdf
- https://www.youtube.com/watch?v=UHBmv7qCey4
- http://www.cs.cmu.edu/~tom/10601_fall2012/slides/boosting.pdf
- http://people.csail.mit.edu/dsontag/courses/ml12/slides/lecture14.pdf
- http://www.mit.edu/~9.54/fall14/slides/Class13.pdf
- http://www.cs.cmu.edu/afs/andrew/course/15/381-f08/www/lectures/clustering.pdf
- https://www.jeremyjordan.me/nn-learning-rate/
- https://scikit-learn.org/stable/modules/cross_validation.html
- https://stats.stackexchange.com/questions/351638/random-sampling-methods-for-handling-class-imbalance
- http://web.ipac.caltech.edu/staff/fmasci/home/astro_refs/PrincipalComponentAnalysis.pdf
- https://towardsdatascience.com/multi-class-metrics-made-simple-part-i-precision-and-recall-9250280bddc2