Module 5

Support Vector Machine

Basics

→Finite dimensional vector space.

→Hyperplane.

→Linearly separable data.

Finite dimensional vector space

Vector

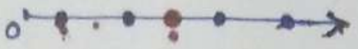
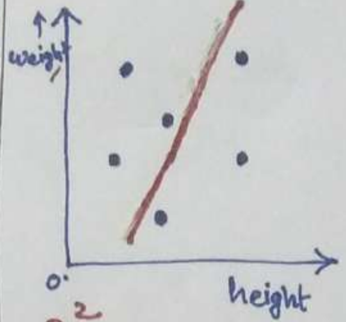
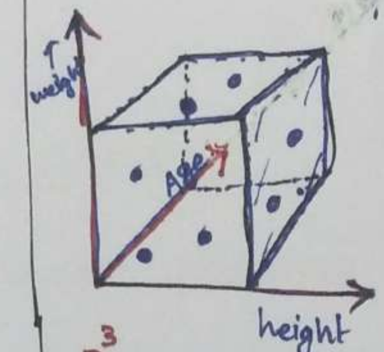
Let n be a positive integer.

* A n-dimensional vector means an ordered n-tuple of real numbers of the form $(x_1, x_2, \dots x_n)$. We can denote this vector by $\vec{x}$.

Vector space

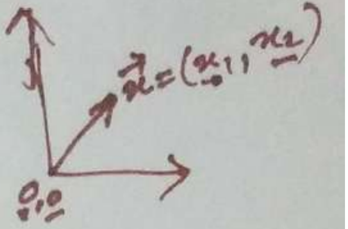
* The set of all n-dimensional vectors with all posible operations on vectors is a n-dimensional vector space.

* Denoted by R^n.

No. of features	Example.	Geometric representation.	Hyperplane
1 [One dimensional]	height	 R^1	point 0-dimen.
2. [two-dimensional]	height, Weight	 R^2	line 1-dimen.
3 [three-dimensional]	height, Weight, Age	 R^3	plane 2-dimen.
Generally. n features.	$(x_1, x_2, \dots \dots x_n)$	R^n [n-dimensional vectorspace]	Hyperplane in (n-1) dimensional

Norm

$$\sqrt{(x_1-0)^2+(x_2-0)^2} = \sqrt{x_1^2+x_2^2}$$



The norm of an n-dimensional vector $\vec{x}=(x_1, x_2, \dots x_n)$, denoted by $\|\vec{x}\|$, is defined by

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

Inner product

The inner product of $\vec{x}=(x_1, x_2, \dots, x_n)$ and $\vec{y}=(y_1, y_2, \dots, y_n)$, denoted by $\vec{x}\cdot\vec{y}$, is defined by

$$\vec{x}\cdot\vec{y} = x_1 y_1 + x_2 y_2 + \cdots\cdots + x_n y_n$$

Hyperplane

* A hyperplane in an n-dimensional space is a flat, (n-1) dimensional subset of that space that divides the space into two separate parts.

* For example,

line is a hyperplane in 2-dimensional space (R^2).

Plane is a hyperplane in R^3 (3-dimensional space).

* Definition

Consider the n-dimensional vector space $\underline{R^n}$.

The set of all vectors $\vec{x} = \underline{(x_1, x_2, \cdots x_n)}$ in R^n whose components satisfy an equation of the form

$$\boxed{\boxed{\alpha_0} + \boxed{\alpha_1 x_1 + \alpha_2 x_2 + \cdots\cdots + \alpha_n x_n} = 0}$$

where $\alpha_0, \alpha_1, \alpha_2, \cdots, \alpha_n$ are scalars,

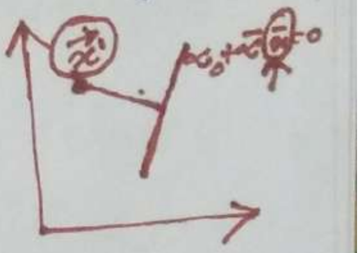
is called a hyperplane in R^n.

$$\boxed{\alpha_0 + \vec{\alpha} \cdot \vec{x} = 0}$$

Distance of a hyperplane from a point

In R^n, the perpendicular distance of a point $\vec{x}' = (x_1', x_2', \dots x_n')$ from a hyperplane $\alpha_0 + \vec{\alpha}\cdot\vec{x} = 0$ is given by

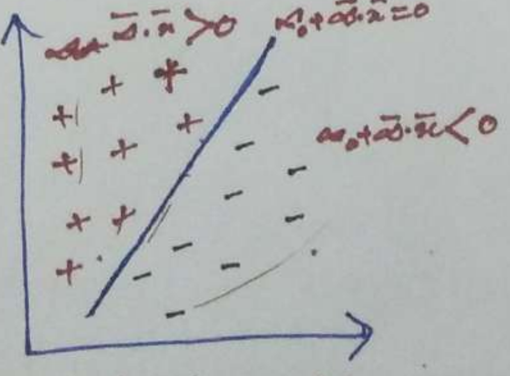
$$\frac{|\alpha_0 + \vec{\alpha}\cdot\boxed{\vec{x}'}|}{\boxed{\|\vec{\alpha}\|}}$$



Linearly separable data

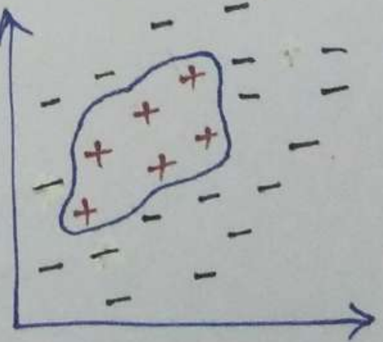
Consider a data set having n features, and two class labels (−1 and +1).

If this dataset can be separated using a hyperplane, we can say that it is a linearly separable data.



$\alpha_0+\vec{\alpha}\cdot\vec{x}>0$
$\alpha_0+\vec{\alpha}\cdot\vec{x}=0$
$\alpha_0+\vec{\alpha}\cdot\vec{x}<0$

Linearly separable data



Not - linearly separable data.

Definition

A data set is linearly separable, if we can find a hyperplane,

$$\alpha_0 + \vec{\alpha}.\vec{x} = 0$$

having the following two properties:

i) For each instance $\vec{x}$ with class-label -1, we have $\alpha_0 + \vec{\alpha}\,\vec{x} < 0$

ii) For each instance $\vec{x}$ with class-label $+1$, we have $\alpha_0 + \vec{\alpha}\,\vec{x} > 0$

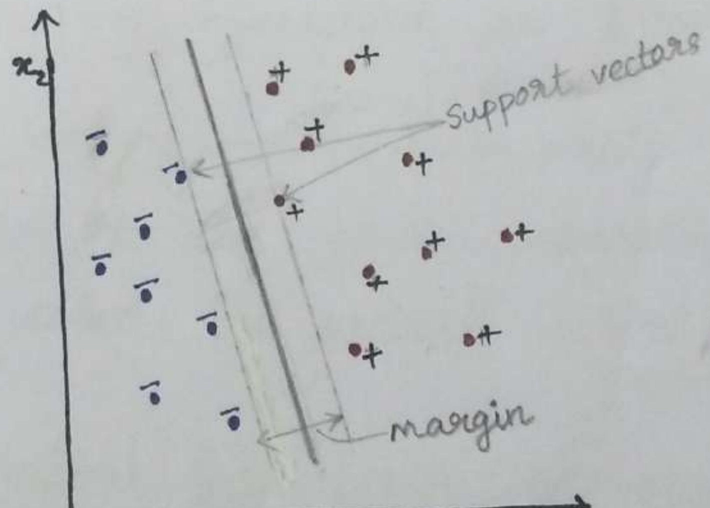
Support Vector Machine (SVM)

→ What is SVM (Optimal Separating hyperplane) ?

→ Mathematical formulation of the SVM problem.

→ Final solution and Algorithm.

Optimal separating hyperplane (SVM)



Consider a linearly separable dataset having two class labels '−1' and '+1'. Also consider a separating hyperplane. H for the data set.

Margin – The double of the smallest perpendicular distance from the training instances to the separating hyperplane is called margin of the separating hyperplane H.

Optimal separating hyperplane – The hyperplane with the maximum margin is called maximal margin hyperplane or optimal separating hyperplane or SVM.

Support vectors – The data points that lie closest to the optimal separating hyperpl are called the support vectors.

Mathematical formulation of the SVM-problem :–

The SVM problem is the problem of finding the equation of the SVM, given a linearly separable two-class data set.

We assume that data set is linearly separable two class data set.

Let the dataset of N points of the form

$$(\vec{x}_1, y_1), (\vec{x}_2, y_2), \ldots\ldots\ldots, (\vec{x}_N, y_N)$$

where y_i's are either +1 or -1.

and each $\vec{x}_i$ is a n-dimensional vector.

Any hyperplane can be written as the set of points $\vec{x} = (x_1, x_2, \ldots . x_n)$ satisfying an equation of the form

$$\vec{w} \cdot \vec{x} + b = 0$$

We can select two parallel hyperplanes that separate the two classes of data, so that the distance between them is large as possible.

The maximum margin hyperplane is the hyperplane that lies halfway between them.

These hyperplanes can be described by the following equations:

$$\vec{w}\cdot\vec{x}+b=+1 \qquad \text{———}\ \textcircled{1}$$

$$\vec{w}\cdot\vec{x}+b=-1 \qquad \text{———}\ \textcircled{2}$$

For any point on or above the hyperplane eq^{n} ①, the class label is +1

i.e $\quad \vec{w}\cdot\vec{x}_i+b \geq +1 \quad$ if $y_i=+1$.

$|||^{ly}$, for any point on or below the hyperplane eq^{n} ②, the class label is −1.

i.e $\quad \vec{w}\cdot\vec{x}_i+b \leq -1 \quad$ if $y_i=-1$.

If we combine the above two conditions, we get:

$$y_i(\vec{w}\cdot\vec{x}_i+b)\geq 1 \text{ for all } 1\leq i\leq N$$

and the distance between the above two hyperplane is

$$\frac{2}{\|\vec{w}\|}$$

So, to maximize the distance between the hyperplanes we have to minimize $\|\vec{w}\|$.

To simplify the further computations, we minimize $\frac{\|\vec{w}\|^2}{2}$.

Definition

We can define the SVM problem as an optimization problem :-

Find a vector $\vec{w}$ and a number b which minimize $\frac{1}{2}\|\vec{w}\|^2$

Subject to. $y_i(\vec{w}\cdot\vec{x}_i - b) \geq 1$ for $i = 1, \ldots, N$

The solution to the SVM problem is a classifier known as SVM classifier.

Let $\vec{w} = \vec{w}^*$ and $b = b^*$ be a solution of the SVM problem. Let $\vec{x}$ be an unclassified data instance.

→ Assign the class label +1 to $\vec{x}$ if $\vec{w}^* \cdot \vec{x} + b^* > 0$

→ Assign the class label −1 to $\vec{x}$ if $\vec{w}^* \cdot \vec{x} + b^* < 0$

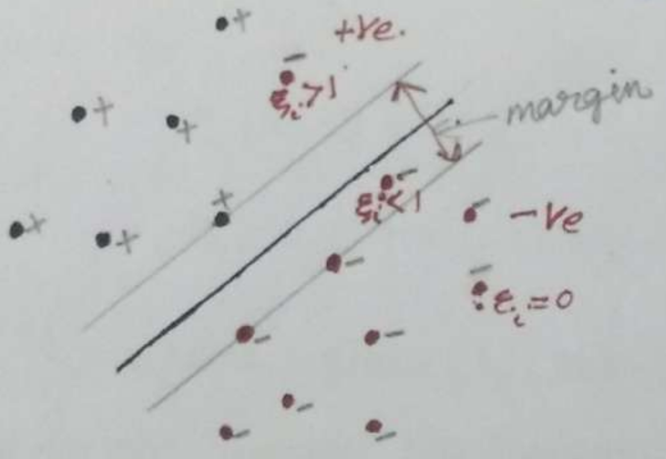
Soft margin hyperplane

* In real life problems, the two-class datasets are only rarely linearly separable.

* There are two types of deviation:

→An instance may lie on the wrong side of the hyperplane and be misclassified.

→An instance may be on the right side but may lie in the margin. [i.e Not sufficiently away from the hyperplane.



In such cases, we introduce additional variables, $\boxed{\xi_i}$ called slack variables which store deviation from the margin.

$\xi_i = 0$	$\bar{x}_i$ is correctly classified.
$0 < \xi_i < 1$	$\bar{x}_i$ is correctly classified; But is in the margin.
$\xi_i \geq 1$	$\bar{x}_i$ is misclassified.

So SVM problem can be reformulated as follows:

Given a two class linearly seperable dataset of N points of the form:

$$(\bar{x}_1, y_1), (\bar{x}_2, y_2), \ldots\ldots, (\bar{x}_N, y_N)$$

Where y_i's are either +1 or −1,

Find vectors $\bar{\omega}$ and $\bar{\xi}$ and a number b. which minimize

$$\boxed{\frac{1}{2}\|\omega\|^2} + C\sum_{i=1}^{N} \xi_i$$

subject to

$$y_i(\bar{\omega}\cdot\bar{x}_i + b) \geq 1 - \xi_i, \text{ for } i = 1, 2, \ldots N$$

$$\xi_i \geq 0, \text{ for } i = 1, 2, \ldots, N$$

The hyperplane given by the equation $\bar{\omega}\cdot\bar{x} + b = 0$ with the values of $\bar{\omega}$ and b obtained as solutions of the reformulated problem, is caled soft margin hyperplane for the SVM problem.

Combining multiple learners

- → Ways to achieve diversity
- → Model combination schemes
  - → Voting
  - → Bagging
  - → Boosting
- → Random forest method

Why combine many learners

There are several reasons why a single learner may not produce accurate results.

- * Set of assumptions that do not hold always.
- * ill-posed problem
- * There may be some instances on which even the best learner is not accurate enough.
- * There is no single learning algorithm that always produces the most accurate output.

Ways to achieve diversity

1) Use different learning algorithms
2) Use the same algorithm with different hyperparameters
3) Use different representations of the input object.
4) Use different training sets to train different base learners.
5) Multiexpert combination methods.
6) Multistage combination methods.

Model combination schemes

1) Voting

* Simplest procedure for combining the outcomes of several learning algorithms.

→ Classification problem (Binary / Multi-class)

Each of the base learners will assign a class label to input x. When a class label is assigned a label, the label gets a vote. In the voting scheme, the class label which gets the maximum number of votes is assigned to x.

→ Regression

Consider L base learners for predicting the value of a variable y. Let $\hat{y}_i$ be the output predicted by the i^{th} base learner.

The final output is computed as.

$$y = w_i \hat{y}_i + w_2 \hat{y}_2 + \cdots\cdots + w_L \hat{y}_L$$

where $w_1, w_2, \ldots w_L$ are the weights and must satisfy the following conditions:

$$w_i \geq 0 \quad \text{for } i = 1, 2, \ldots L$$

and $\sum_{i=1}^{L} w_i = 1$.

This is called weighted voting scheme.

In simple voting scheme, we take

$$w_i = \frac{1}{L} \quad \text{for } i = 1, 2, \ldots, L.$$

2) Bagging (Bootstrap Aggregating)

* Bagging is a model combination scheme whereby base learners are made different by training them over slightly different training sets..

* Generating L slightly different training sets from a given training set is done by bootstrap.

These training sets are similar because they are all drawn from the same original training set, but they are also slightly different due to chance.

* Then these training sets are learned using un unstable algorithm. A learning algorithm is an unstable algorithm if small changes in the training set causes a large difference in the generated learner.

Eg:- Decision tree, multilayer perceptron are unstable algms.

* Bagging (Bootstrap aggregating) uses bootstrap to generate L training sets, trains them using an unstable learning procedure and then during testing, takes an average.

* Bagging can be used both for classification and regression.

3) Boosting

In boosting, the complemetary base-learners are generated by training the next learner on the mistakes of the previous learners.

Let d_1, d_2, d_3 be the three learning algorithm and X be a large training set.

1) Divide X into three sets X_1, X_2, X_3.

2) Train the base learner d_1 using X_1.

3) Then take X_2 and feed it to d_1.

4) Now take all the instances in X_2 that are misclassified by d_1 and use it to train the second base learner d_2.

5) Take X_3 and feed it to d_1 and d_2.

6) The instances on which d_1 and d_2 disagree form the training set of d_3.

7) During testing, given an instance, we give it to d_1 and d_2. If they dgree, that is the response; otherwise the response of d_3 is taken as the output.

Advantage

* Reduced error rate.

Disadvantage

* Requires a very large training sample.

Ensemble learning and Random forest

The word 'ensemble' means a "group of things acting together as a whole".

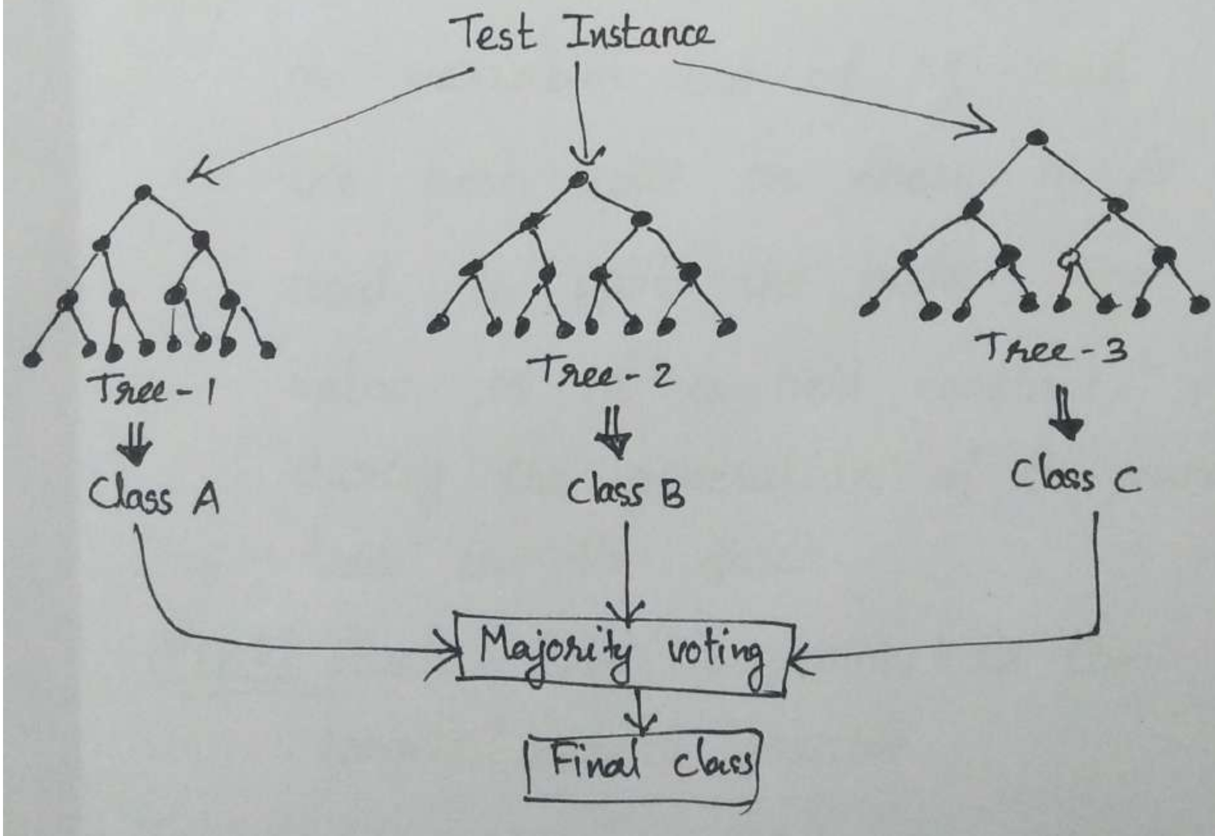
In machine learning, an ensemble learning method consist of the following two steps:

1) create different models for solving a particular problem using a given data.

2) Combine the models created to produce improved results.

Random Forest algorithm

A random forest is an ensemble learning method where multiple decision trees are constructed and then merged to get more accurate prediction.



Each tree in random forest is generated as follows.

Step 1: If the number of examples in the training set is N, randomly select N samples with replacement and use it to generate the tree.

Step 2: If there are M number of input variables, randomly choose m variables out of M and the best split on these m is used to split the node. The value of m is held constant during the generation of the various trees in the forest.

Step 3: Each tree is grown to the largest extent possible.

To classify a new object, give that object as input to each of the trees in the forest. Each tree gives a classification. The class label which gives maximum vote will be the final output.