Machine Learning

Support Vector Machine

SVM

- Commonly used for classification. Also called Support Vector Classifier

- Useful for classification of complex data. For example. non linear separable data

- Though SVM is linear machine learning model. i.e. it uses linear function, it can handle non linear separable data

- All data needed in numerical format

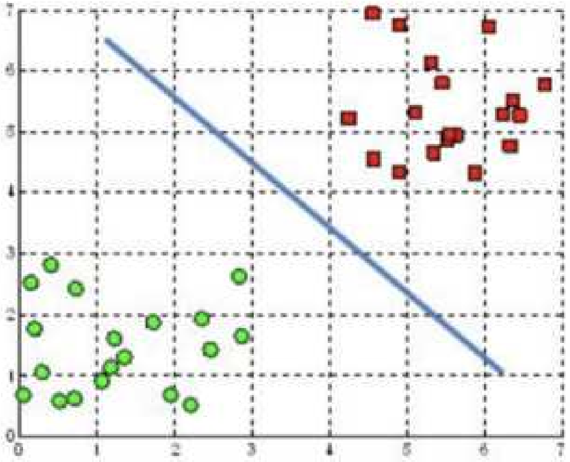
- It attempts to find Hyperplane, linearly separating data. In low dimension space, it is a line: $w_1x_1 + w_2x_2 + b = 0$
- Where x_1, x_2 are features

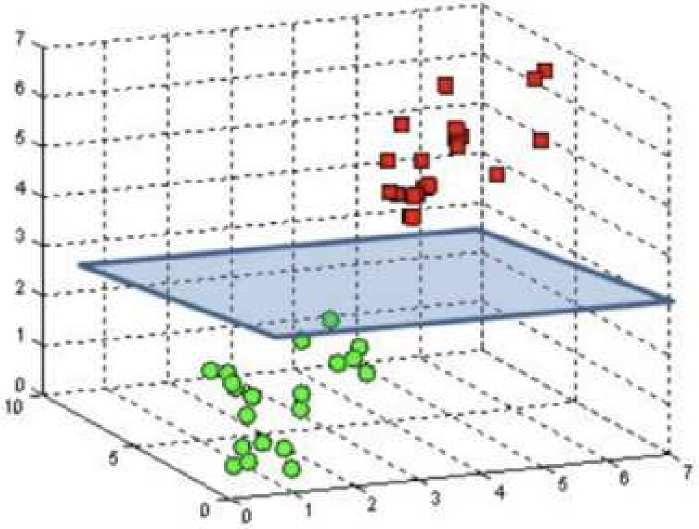
SVM

- Following diagrams shows a hyperplane in 2-D and 3-D respectively

- *A hyperplane is a subspace of one dimension less than its ambient space*

- In 2-D space it is a line, in 3-D space it is a plane





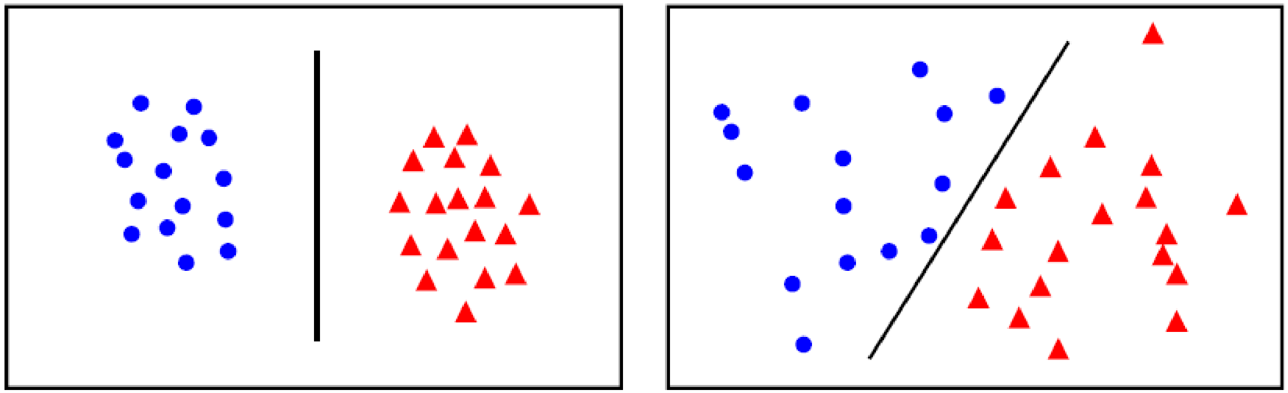
- In higher dimension space, the equation can be represented as
  $w_1x_1 + w_2x_2 + w_3x_3 + \ldots\ldots. + b = 0$
  This will be represented as **(w.x) + b = 0**

SVM History

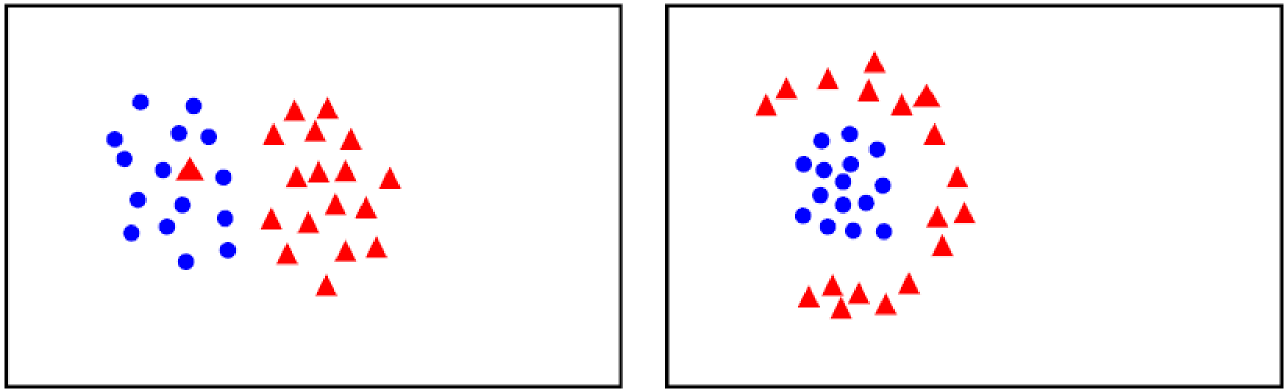
- Perceptron Algorithm –
  - Select random sample from training set as input
  - If classification is correct, do nothing
  - If classification is incorrect, modify the weight vector w
- Repeat the above two steps until the entire training set is classified correctly
- If the data is linearly separable then it will find the line/surface that linearly separate the data

Linear separation

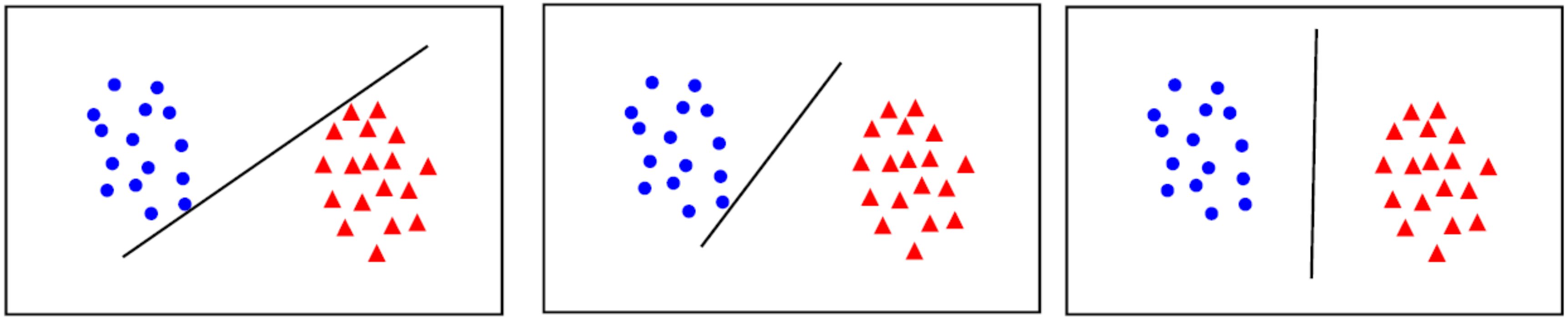
Linearly separable



Linearly **not** separable



Best Linear Classifier?

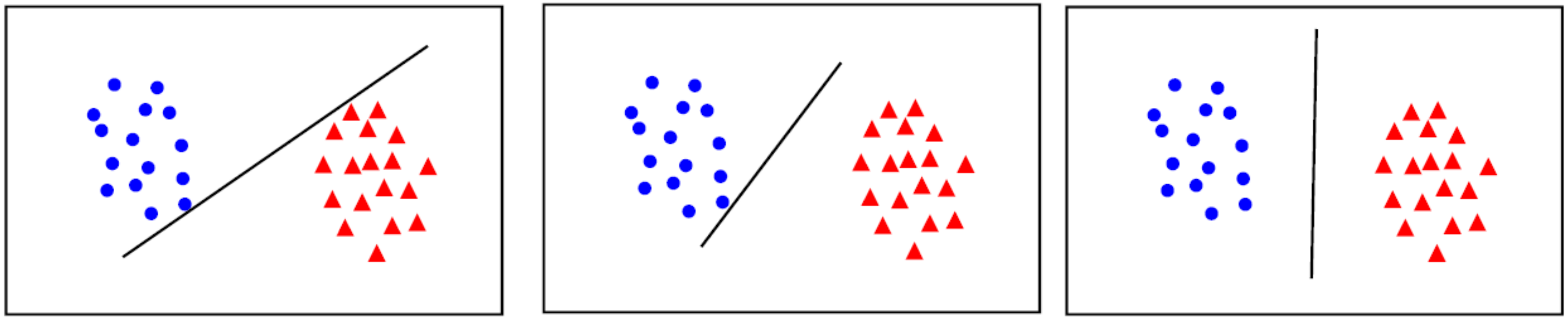


- Which of the above line fits best?

- All will give 100% accuracy with the above data

- Will there be difference in performance in production?

Highest Margin Classifier

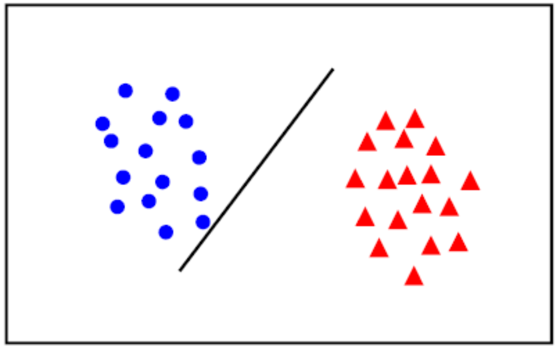


- The first line looks most vulnerable to the variance and the third line looks least vulnerable.

- The two points nearest from different clusters define the margin around the line and are support vectors

- SVMs try to find the third kind of line where the line is at max distance from both the clusters simultaneously

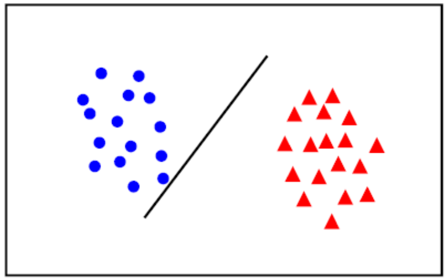
Concept of Equation of Line



(following is the traditional geometry notation)

- Consider equation of line x – y – 1 = 0
- Consider points (2,1), (5,4), (2,2), (2,6), (3,1), (6,1). Use these points in the equation of the hyperplane
- Observe that the points on the hyperplane give 0. Points on one side result in negative value and points on other side result in positive value. Also higher magnitude for points that are further away

Finding Hyperplane



- Convention for notations we will use is
- (w.x) + b = 0, where

  x is vector of features and w is vector of coefficients

  y_i is label of i^{th} row of data

- Label y_i is -1 or +1 (it is -1 for the side on which we get negative value when a point is substituted in the equation of hyperplane)

- For each point, by substituting the values in the equation of hyperplane, we can determine
  - which side of the hyperplane the point is located (based on whether the value is positive or negative) and
  - measure of the distance from the hyperplane

- This will help us find the hyperplane with maximum min margin. i.e. Maximum distance from the points that are closest to the hyperplane.

Finding Hyperplane

- These points are called support vectors.

- By finding optimum hyperplane with maximum distance from support vectors we would be able to find the desired classifier

---

- However, if we take the 'distance' of the points calculated earlier (by substituting values in equation), we can make error as the distance on one side gives negative value

- To prevent that, we multiply the equation by y_i.

$$f = y(\mathbf{w} \cdot \mathbf{x} + b)$$

- The sign of *f* will be:
  - Positive if the point is correctly classified
  - Negative if the point is incorrectly classified

Finding Hyperplane

- This is called Functional Margin. So the objective is to find a Hyperplane with highest functional margin

- The problem with functional margin is that it is not scale invariant.

- The algorithm can get into unnecessary evaluations of w and b. To prevent this, we normalize the expression by dividing it by magnitude of w

$$y\left(\frac{\mathbf{w}}{||\mathbf{w}||}\cdot\mathbf{x}+\frac{b}{||\mathbf{w}||}\right)$$

- This is called Geometric Margin.

- Thus the objective is to find a hyperplane which has maximum of minimum GM. i.e. Maximum Geometric margin from support vectors

Slack variable

- The optimization problem for SMV is

$$\text{minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2 \qquad \text{subject to} \quad y_i(\mathbf{w}\cdot\mathbf{x}_i + b) >= 1$$

- In 1995, Vapnik and Cortes introduced slack variable which allows data points to be on the wrong side of the margins

- To give user a control over the slack variable while building a model, another term 'C' is introduced and the optimization looks like

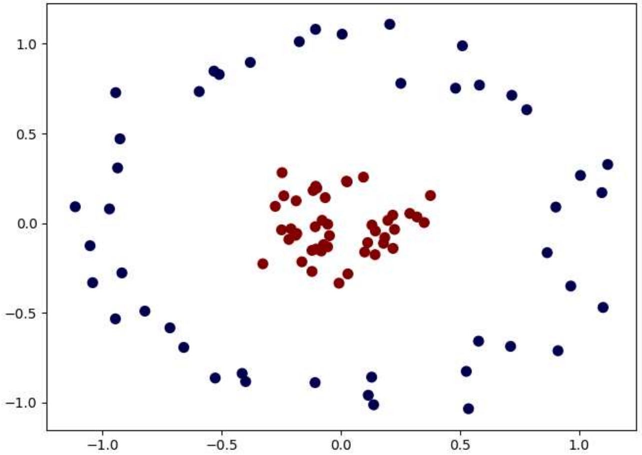
$$\text{minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\zeta_i$$

$$\text{subject to} \quad y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \zeta_i$$

  - small C allows constraints to be easily ignored
  - large C makes constraints hard to ignore

SVM – Kernel

- Limitation of SVM: what happens if the data are not linearly separable?



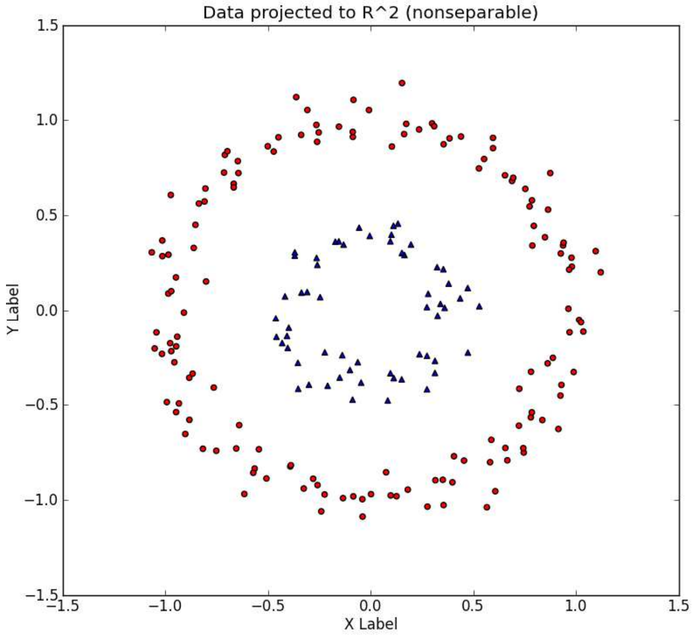
- There is a logical separator but it is not linear. Regular algorithm SVM cannot effectively divide such data

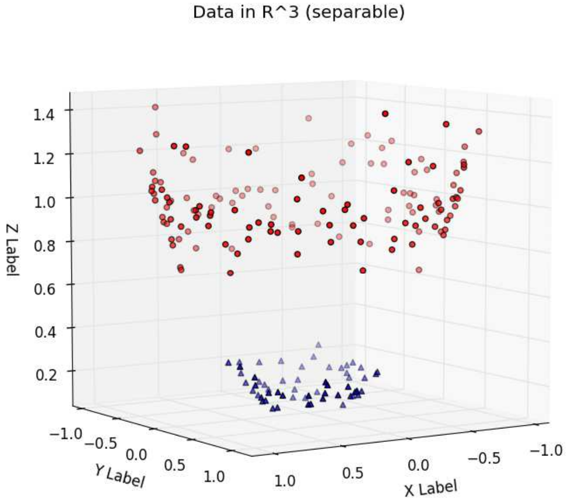
- SVM uses kernel trick to make it linearly separable

SVM - Kernel

- The concept of Kernel is based on Cover’s theorem.

- If a dataset is not linearly separable, it can be transformed into a linearly separable training set by projecting it into a higher-dimensional space using non-linear transformation



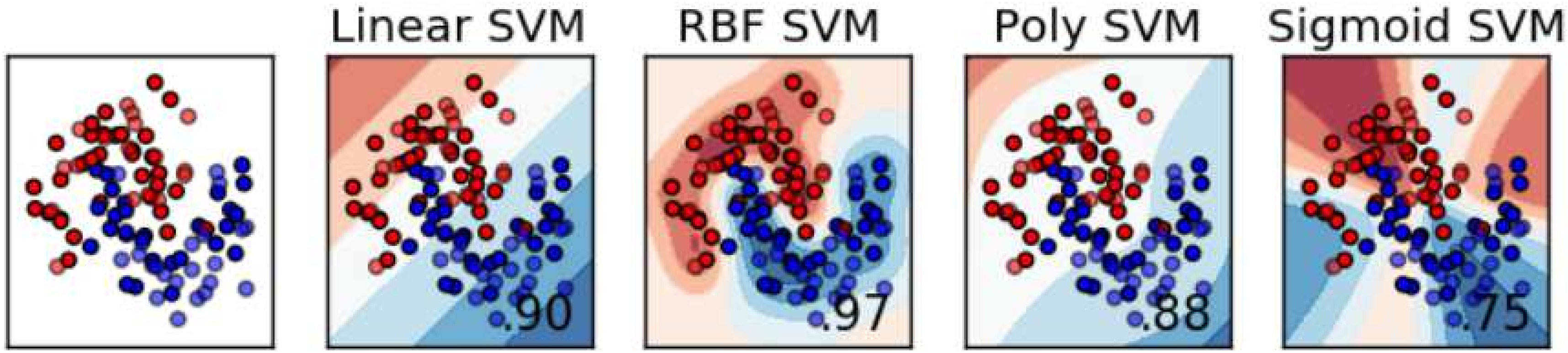


SVM - Kernel

- The idea is mapping the non-linear separable data-set into a higher dimensional space where we can find a hyperplane that can separate the samples.

- Different types of kernels
  - Polynomial kernel: $(x_i \cdot x_j + 1)^p$
  - Gaussian kernel : $e^{\frac{-1}{2\sigma^2}(x_i - x_j)^2}$
  - RBF kernel : $e^{-\gamma(x_i - x_j)^2}$ (Radial Basis Function)
  - Sigmoid kernel : $\tanh(\eta\, x_i \cdot x_j + \nu)$

SVM - Kernel



- Source: https://gist.github.com/WittmannF/60680723ed8dd0cb993051a7448f7805

SVM

- Advantages
  - Makes no assumptions about underlying data sets
  - Very stable as it depends on the support vectors only.
  - Powerful classifier - capable of modelling relatively more complex patterns
  - Not influenced by outliers
- Disadvantages
  - Computationally intensive
  - Generally treated as a black-box model
  - No probabilistic estimate available

SVM

- Hand-on exercise