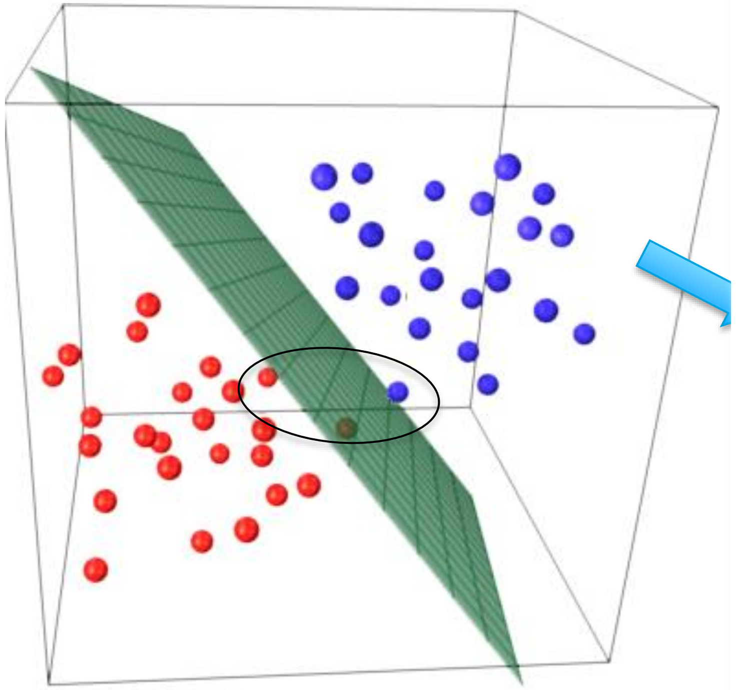
Support Vector Machines

1. Known as maximum-margin hyperplane, find that linear model with max margi. Unlike the liner classifiers, objective is not minimizing sum of squared errors but finding a line/plane that separates two or more groups with maximum margins



http://stackoverflow.com/questions/9480605/what-is-the-relation-between-the-number-of-support-vectors-and-training-data-and

Support Vector Machines

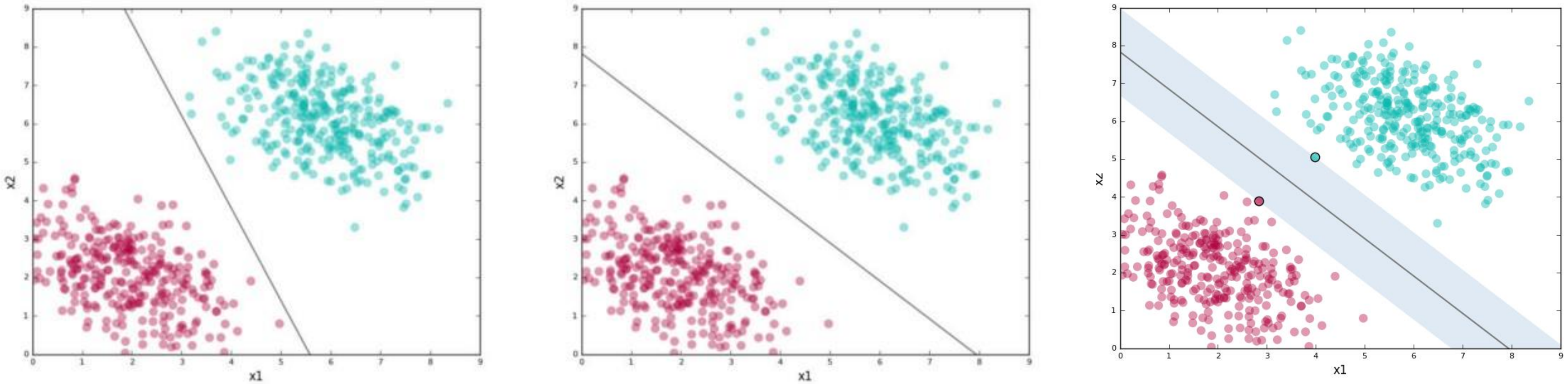
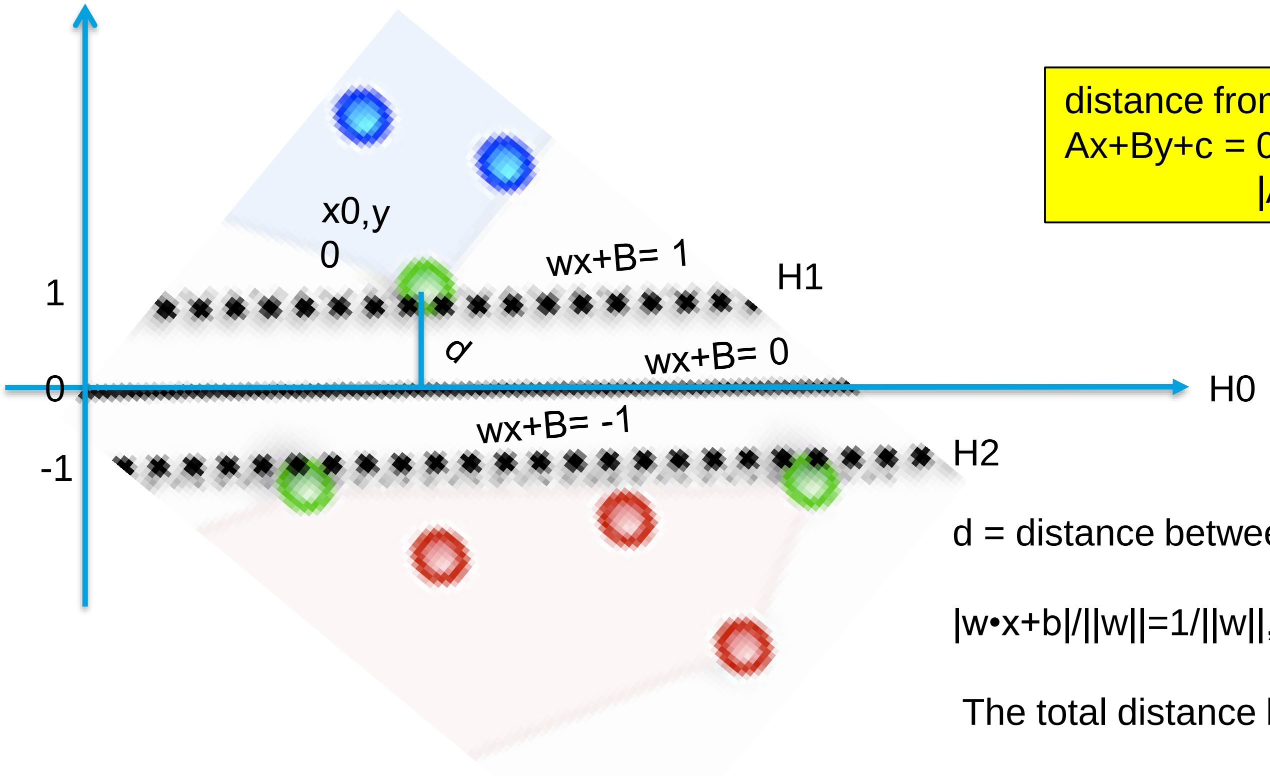


Image Source : https://dzone.com/articles/support-vector-machines-tutorial

1. First line does separate the two sets but id too close to both red & green data points
2. Chances are that when this model is put in production, variance in both cluster data may force some data points on wrong side
3. The second line doesn't look so vulnerable to the variance. The two points nearest from different clusters define the margin around the line and are support vectors
4. SVMs try to find the second kind of line where the line is at max distance from both the clusters simultaneously

Support Vector Machines



distance from a point(x0,y0) to a line:
Ax+By+c = 0 is:
|Ax0 +By0 +c|/sqrt(A^2+B^2)

d = distance between H0 and H1 is

|w•x+b|/||w||=1/||w||,

The total distance between H1 and H2 is thus: 2/||w||

2. Think in terms of multi-dimensional space. SVM algorithm has to find the combination of weights across the dimensions such that they hyperplane has max possible margin around it
3. All the predictor variables have to be numeric and scaled.

Support Vector Machines Allowing Errors

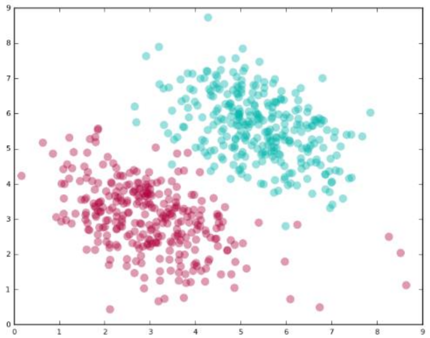
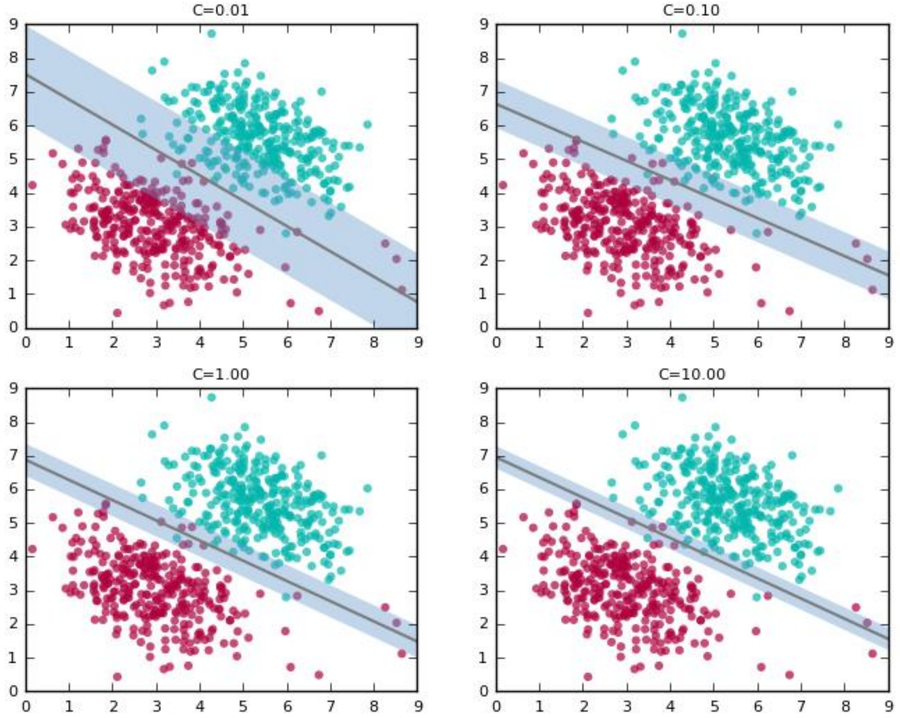
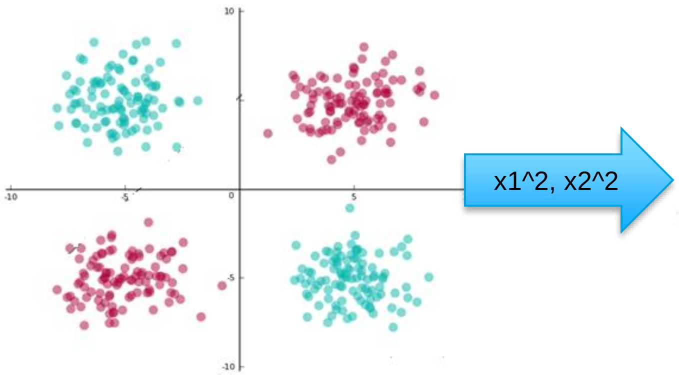


Image Source : https://dzone.com/articles/support-vector-machines-tutorial



1. Data in real world is typically not linearly separable.
2. There will always be instances that a linear classifier can’t get right
3. SVM provides a complexity parameter, a tradeoff between: wide margin with errors or a tight margin with minimal errors. As C increases, margins become tighter

Support Vector Machines Linearly Non Separable Data



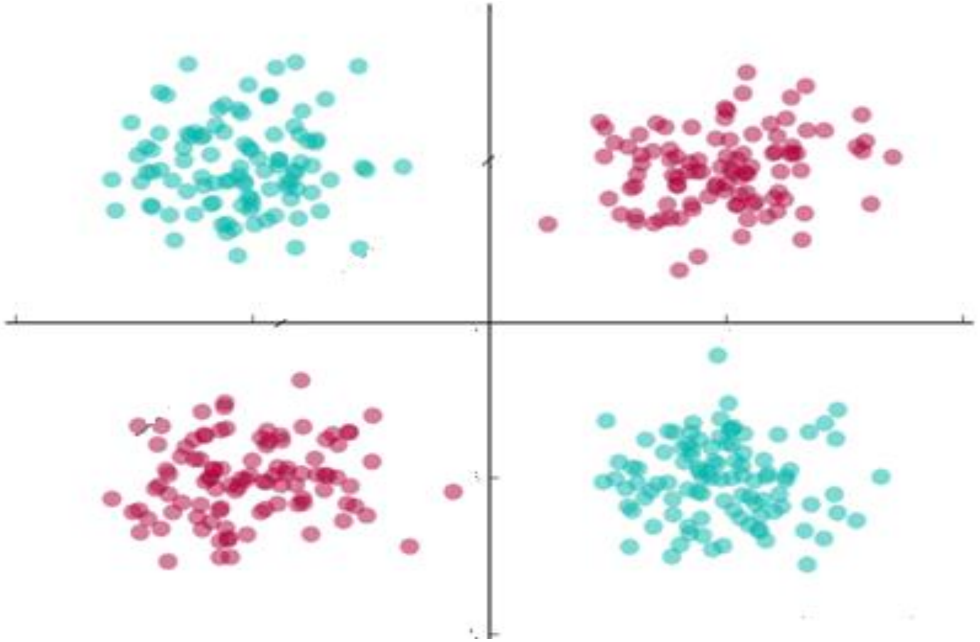


Image Source : https://dzone.com/articles/support vector-machines-tutorial

1. When data is not linearly separable, SVM uses kernel trick to make it linearly separable
2. This concept is based on **<u>Cover's theorem</u>** "given a set of training data that is not linearly separable, with high probability it can be transformed into a linearly separable training set by projecting it into a higher-dimensional space via some non-linear transformation"
3. In the pic above, replace x1 with x1^2, x2 with x2^2 and create a third dimension x3 = sqrt(2x1x2)

Support Vector Machines Linearly Non Separable Data

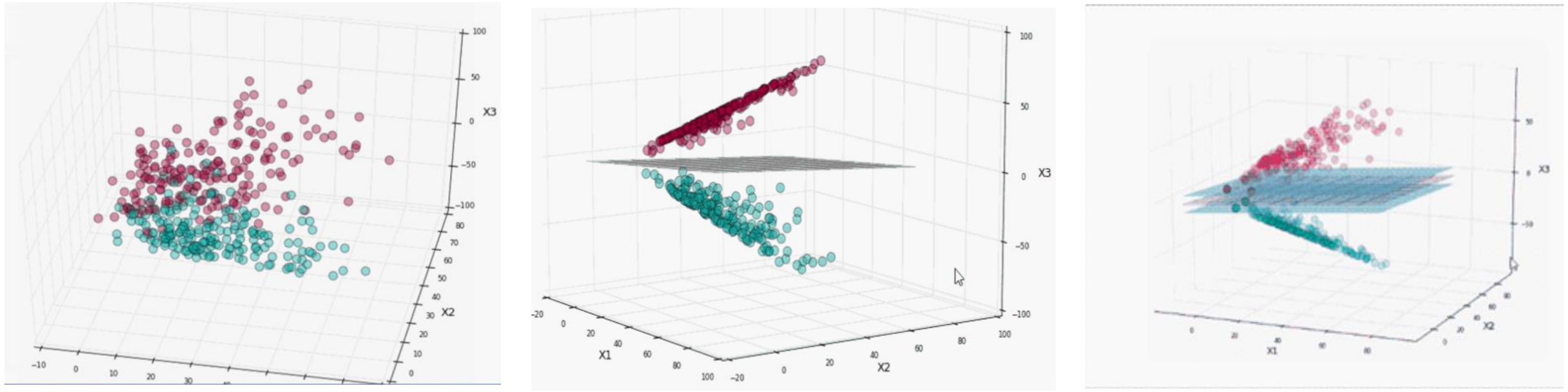


Image Source : https://dzone.com/articles/support-vector-machines-tutorial

1. Using kernel tricks the data points are project to higher dimensional space
2. The data points become relatively more easily separable in higher dimension space
3. SVM can now be drawn between the data sets with a given complexity

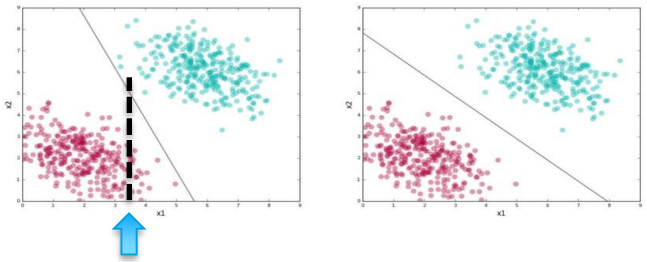
Support Vector Machines Basic Idea

1. Suppose we are given training data $\{(x1, y1),...,(xn, yn)\} \subset X \times R$, where X denotes the space of the input patterns (e.g. X = Rd).
2. Goal is to find a function f(x) that has at most **ε deviation** from the actually obtained targets yi for all the training data, and at the same time is **as flat as possible**
3. In other words, we do not care about errors as long as they are less than ε, but will not accept any deviation larger than this
4. f can take the form **f(x) = (w, x)+ b with w ∈ X, b ∈ R**
5. Flatness means that one seeks a small w. One way to ensure this is to minimize the ||w||^2 = (w, w).

Support Vector Machines Basic Idea

6. The problem can be represented as convex optimization problem

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|w\|^2 \\ \text{subject to} & \begin{cases} y_i - \langle w, x_i \rangle - b & \leq \quad \varepsilon \\ \langle w, x_i \rangle + b - y_i & \leq \quad \varepsilon \end{cases} \end{array}$$



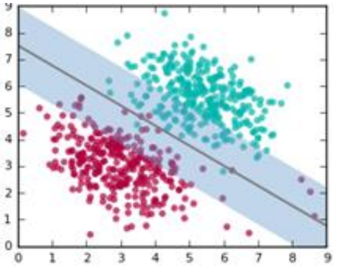
7. In the first picture, ||w||^2 is not minimized, neither the third constraint. Take the pointer to be x value, yi – (w, xi) – b is < e i.e. diff between green dot and the line but (w, xi) + b –yi i.e. diff between line an red dot is not < e.
8. In second picture, all three constraints are met
9. Sometimes, it may not be possible to meet the constraint due to data points not being linearly separable so we may want to allow for some errors.

Support Vector Machines Basic Idea

10. We introduce slack variables ξi, ξ∗ i to cope with otherwise infeasible constraints of the optimization problem and this is known as soft margin classifier

$$\text{minimize} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{\ell}(\xi_i + \xi_i^*)$$

$$\text{subject to} \quad \begin{cases} y_i - \langle w, x_i \rangle - b & \leq & \varepsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i & \leq & \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* & \geq & 0 \end{cases}$$



11. The epsilon term allows some errors i.e. data points lie within the error margins where error margins is e + epsilon

Support Vector Machines Kernel Functins

1. SVM libraries come packaged with some standard kernel functions such as polynomial, radial basis function (RBF), and Sigmoid

2. For degree-*d* polynomials, the polynomial kernel looks like $K(x, y) = (x^{\mathsf{T}} y + c)^d$ where x and y are input vectors in lower dimension space, c is a user specified constant (usually 1). K denotes inner product of x,y in higher dimension space

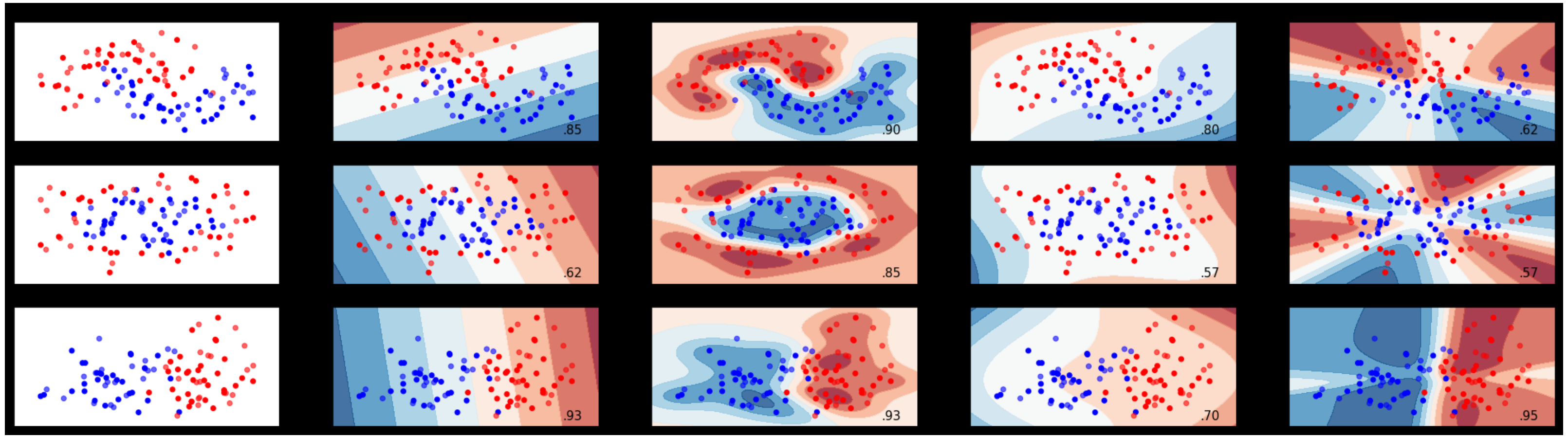
3. RBF (Radial Basis Function) kernel on two samples x and x' is represented as -
$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right)$$

4. It ranges from 0 when distance between x and x' increases (e^-infinity becomes 0) and becomes 1 when x = x' because x – x' = 0 and anything raised to 0 is 1

Support Vector Machines Kernel Functions

5. Sigmoid Kernel looks like **K(X,Y)=tanh(γ·X(transpose)Y+r)**
6. Linear Kernel are of the form that represents linear equation

Feature space | Linear Kernel | RBF Kernel | Polynomial Kernel | Sigmoid Kernel



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

Machine Learning (Support Vector Machines)

Strengths	Weakness
Very stable as it depends on the support vectors only. Not influenced by any other data point including outliers	Computationally intensive
Can be adapted to classification or numeric prediction problems	Prone to over fitting training data
Capable of modelling relatively more complex patterns than nearly any algorithm	Assumes linear relation between dependent and independent variables
Makes no assumptions about underlying data sets	Generally treated as a blackbox model