Support Vector Machines

Dr Muhammad Atif Tahir

Professor

School of Computer Science

National University of Computing & Emerging Sciences

Karachi Campus

Classification Techniques

- Decision Tree based Methods
- Rule-based Methods
- Memory based reasoning
- Neural Networks
- Naïve Bayes and Bayesian Belief Networks
- Support Vector Machines

Support Vector Machines

- Theoretically well motivated algorithm: developed from Statistical Learning Theory (Vapnik & Chervonenkis) since the 60s

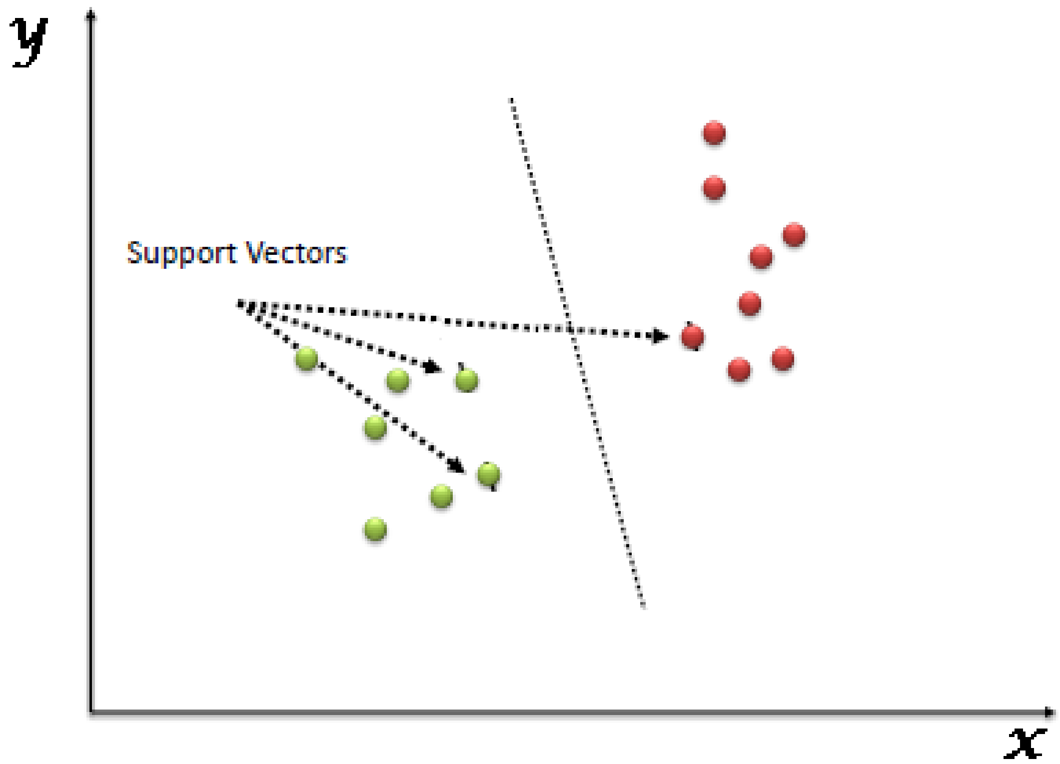
- Empirically good performance: successful applications in many fields (bioinformatics, text, image recognition, . . .)

Support Vector Machines

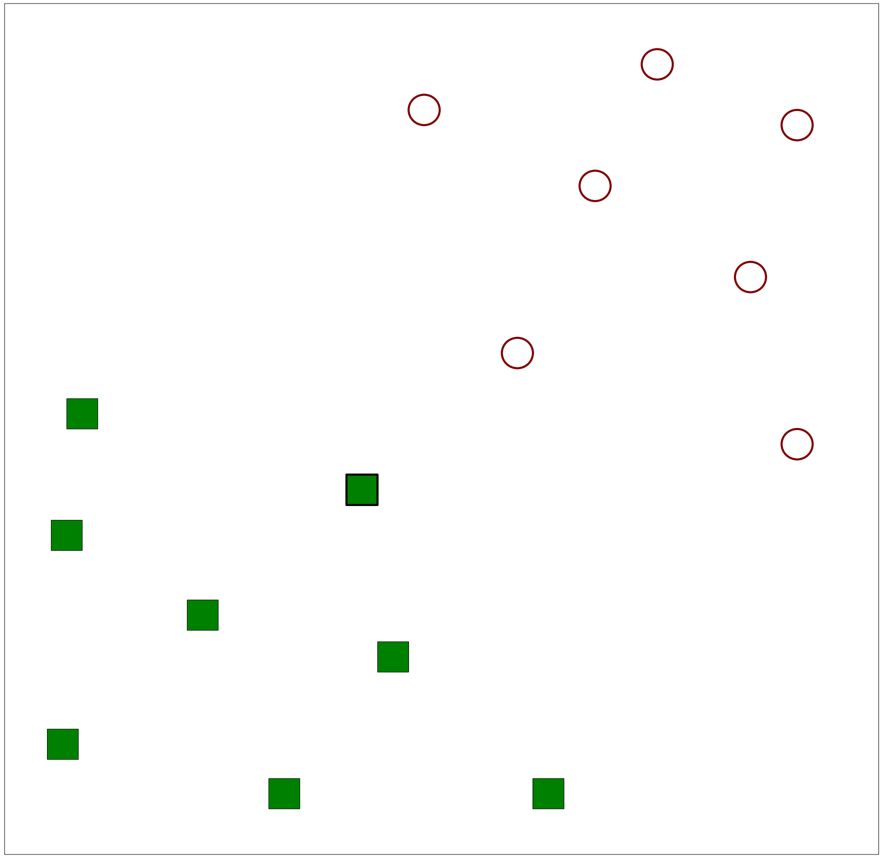
- Centralized website: www.kernel-machines.org.
- Several textbooks, e.g. ”An introduction to Support Vector Machines” by Cristianini and Shawe-Taylor is one.
- A large and diverse community work on them: from machine learning, optimization, statistics, neural networks, functional analysis, etc

Support Vector Machine

- *The goal of a support vector machine is to find the optimal separating hyperplane which maximizes the margin of the training data*
- Support Vector Machine” (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges
- However, it is mostly used in classification problems

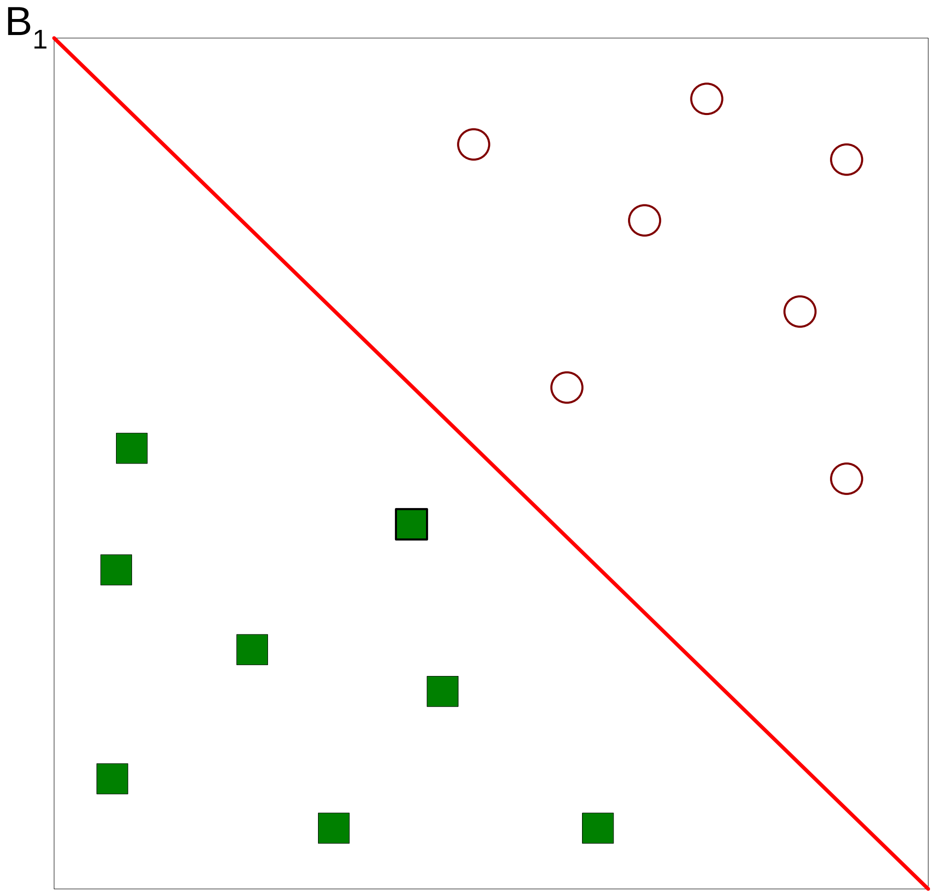

y
Support Vectors
x

Support Vector Machines



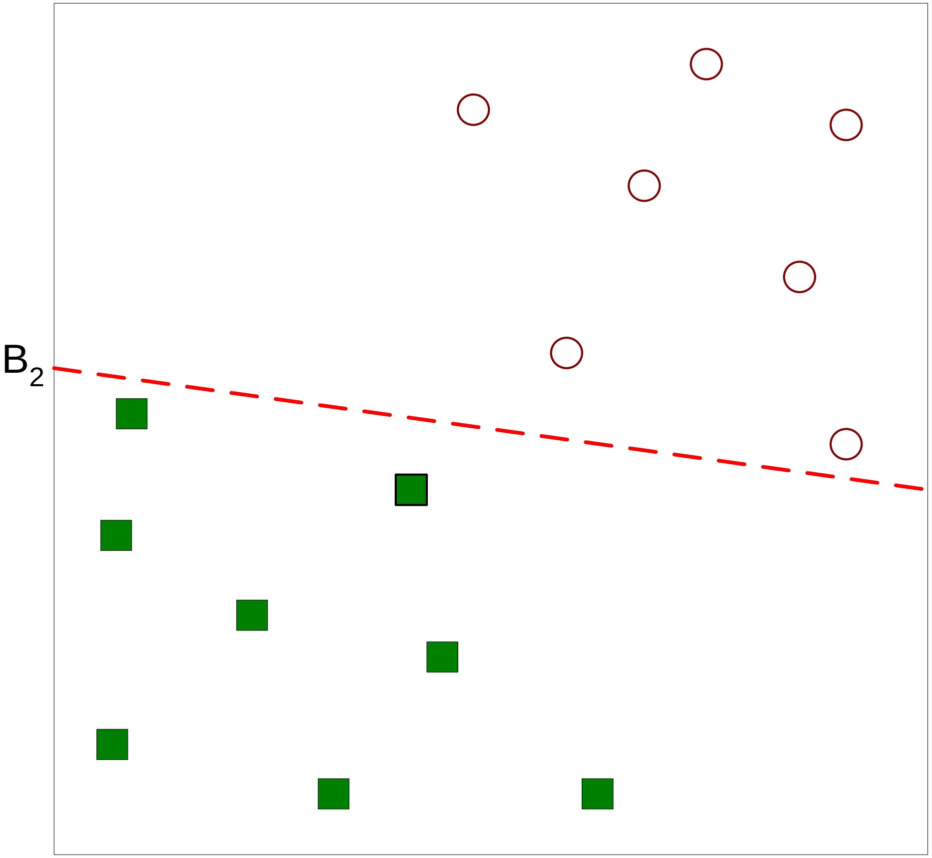
- Find a linear hyperplane (decision boundary) that will separate the data

Support Vector Machines



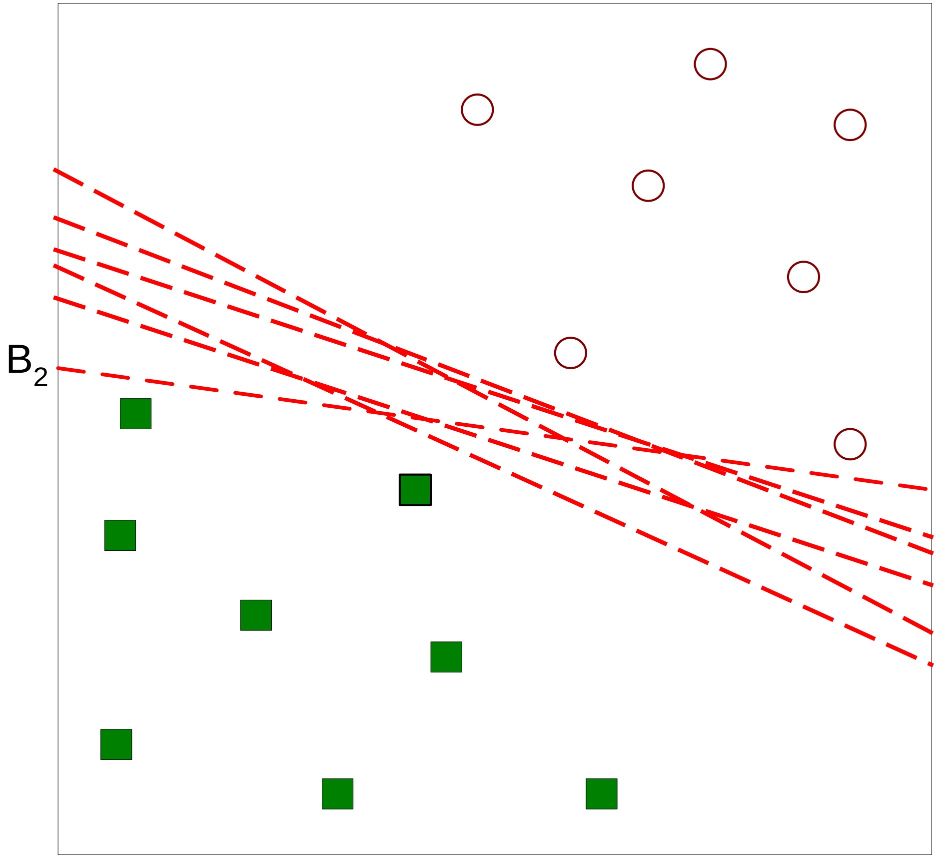
- One Possible Solution

Support Vector Machines



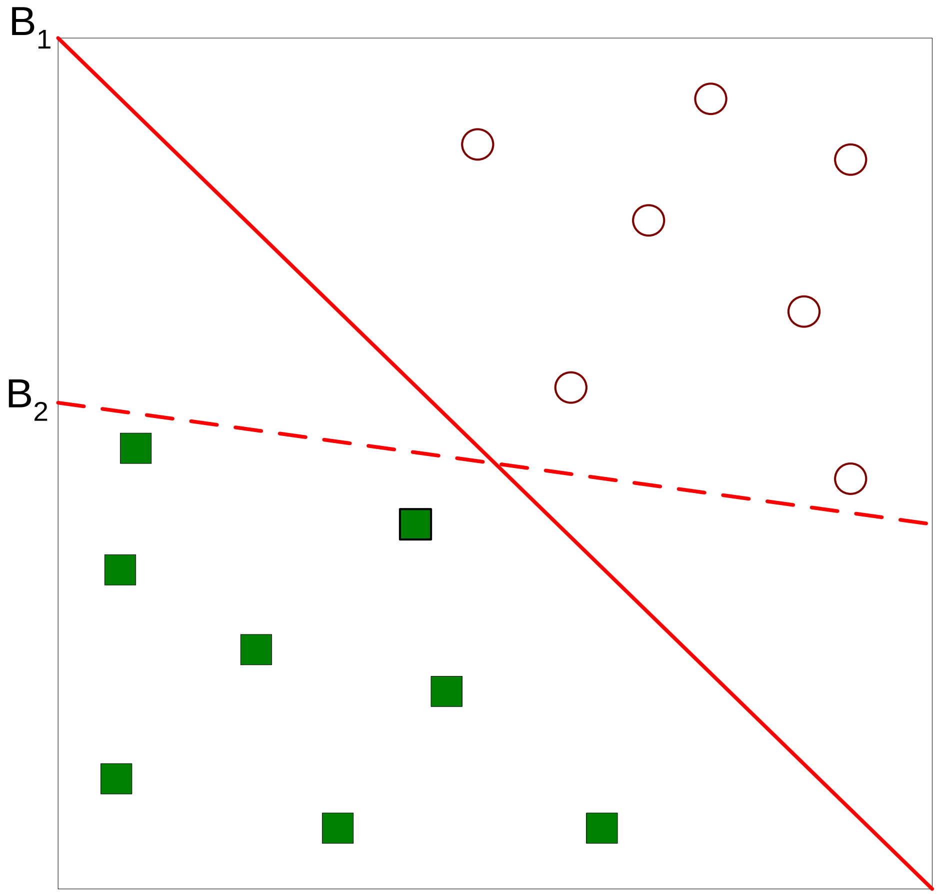
- Another possible solution

Support Vector Machines



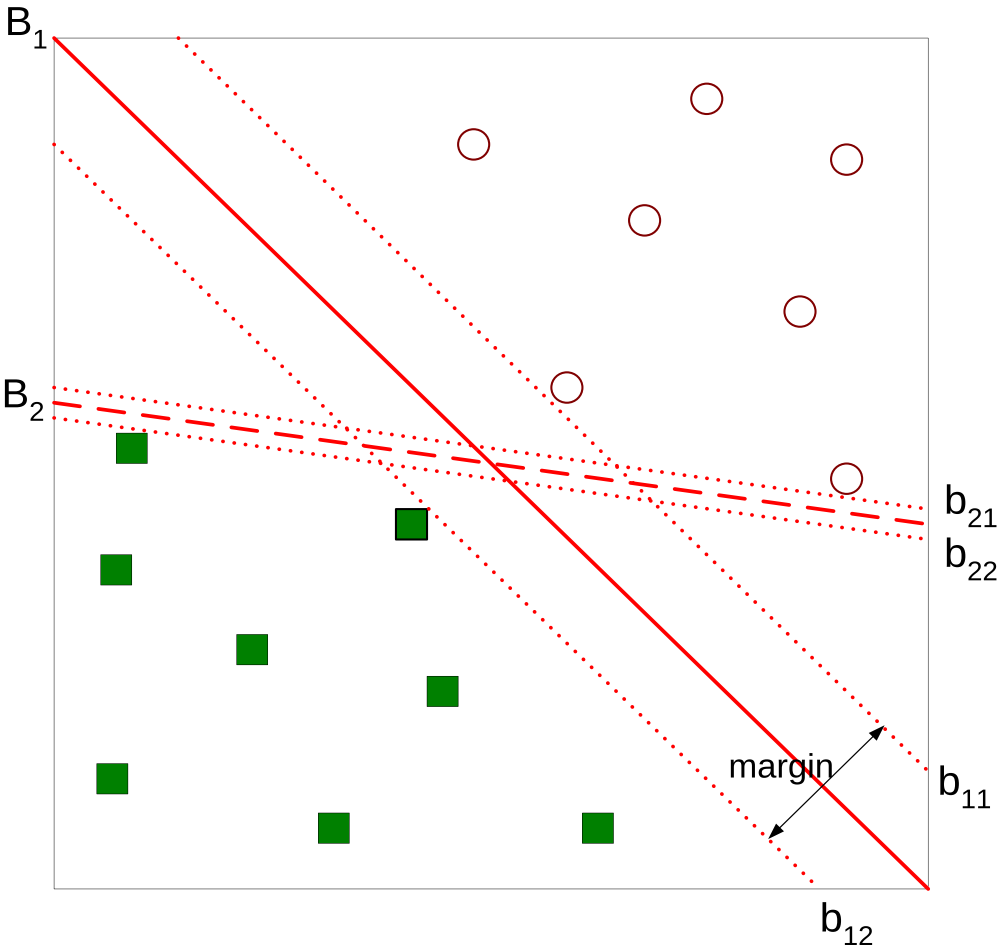
- Other possible solutions

Support Vector Machines



- Which one is better? B1 or B2?
- How do you define better?

Support Vector Machines



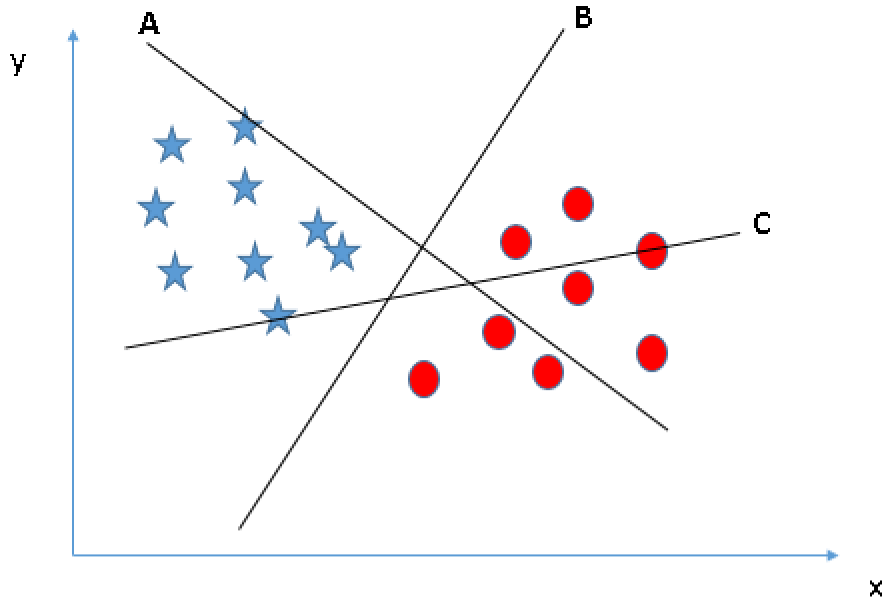
- Find hyperplane maximizes the margin => B1 is better than B2

What is a Hyperplane

An hyperplane is a generalization of a plane

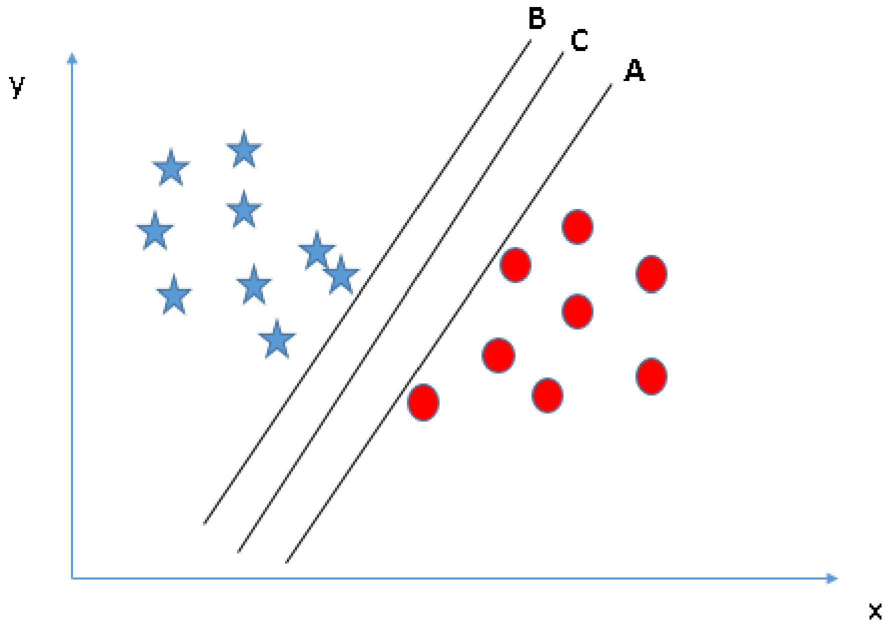
- in one dimension, an hyperplane is called a point
- in two dimensions, it is a line
- in three dimensions, it is a plane
- in more dimensions you can call it an hyperplane

Identify the right hyper-plane (Scenario-1)



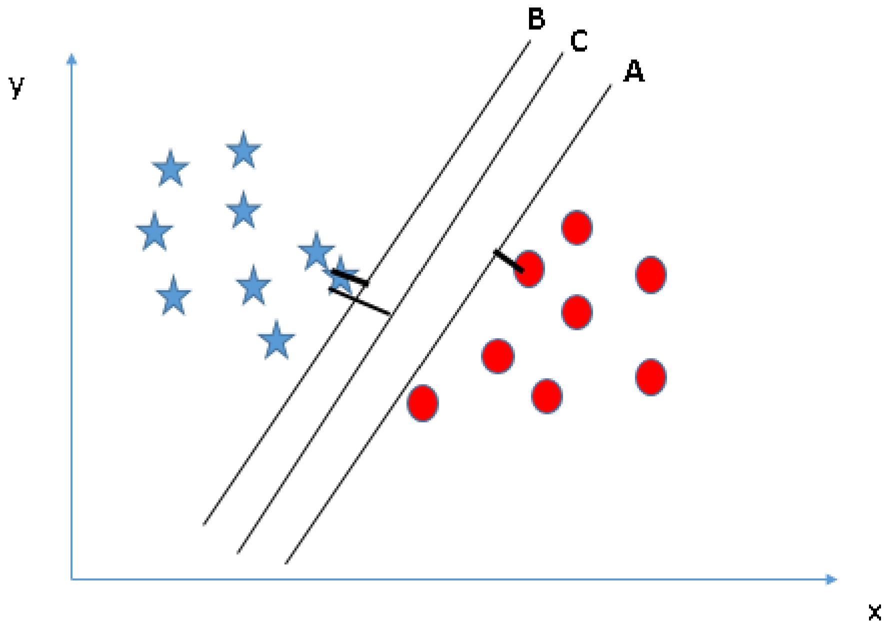
Select the hyper-plane which segregates the two classes better”. In this scenario, hyper-plane “B” has excellently performed this job

Identify the right hyper-plane (Scenario-2)



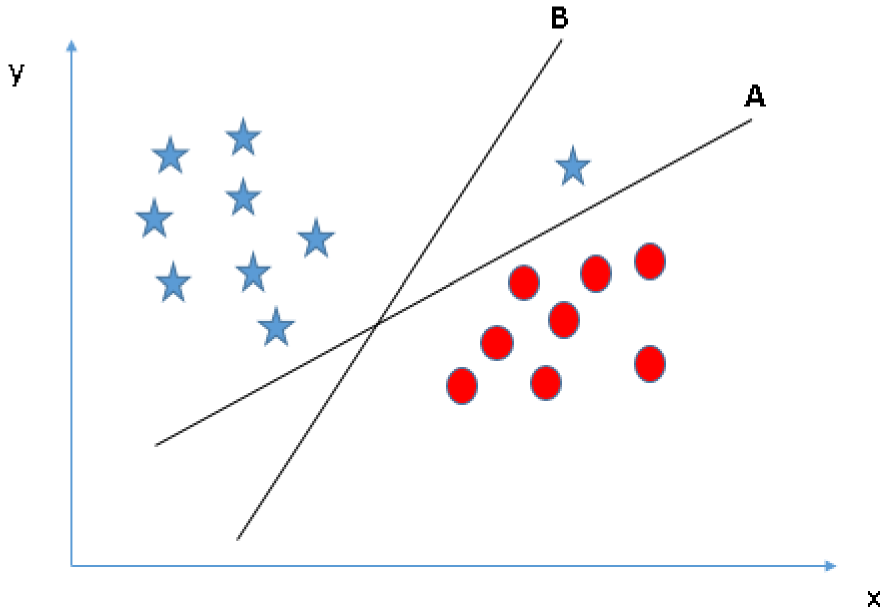
- A, B and C are all good hyperplanes
- Here, maximizing the distances between nearest data point (either class) and hyper-plane will help us to decide the right hyper-plane
- This distance is called as **Margin**

Identify the right hyper-plane (Scenario-2)



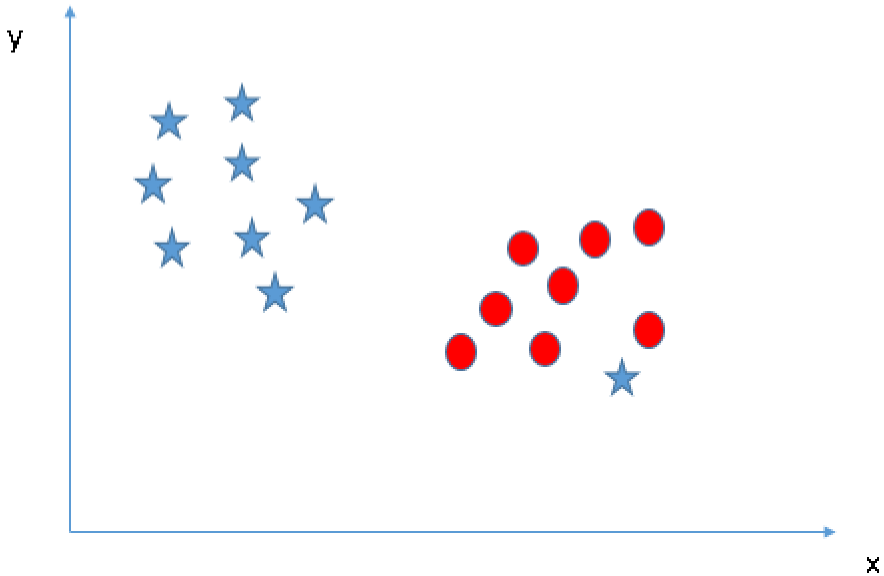
- Margin for hyper-plane C is high as compared to both A and B
- Hence, we name the right hyper-plane as C
- Another lightning reason for selecting the hyper-plane with higher margin is robustness
- If we select a hyper-plane having low margin then there is high chance of miss-classification

Identify the right hyper-plane (Scenario-3)



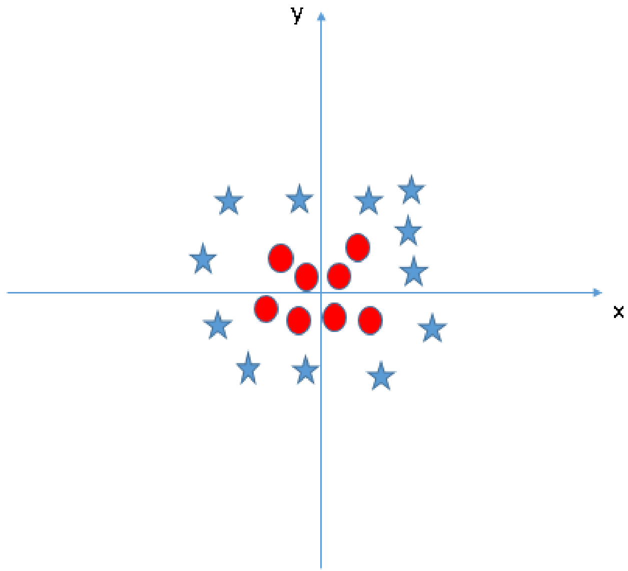
- Hyper-plane **B** may be selected due to higher margin compared to **A**
- But, here is the catch, SVM selects the hyper-plane which classifies the classes accurately prior to maximizing margin
- Here, hyper-plane B has a classification error and A has classified all correctly. Therefore, the right hyper-plane is **A**

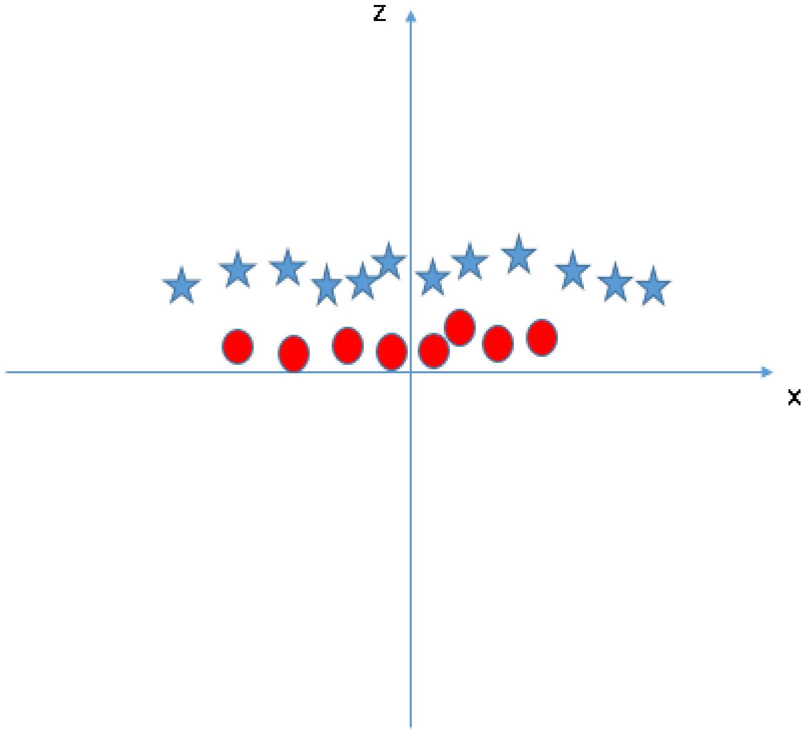
Identify the right hyper-plane (Scenario-4)



- Outliers ignored by SVM

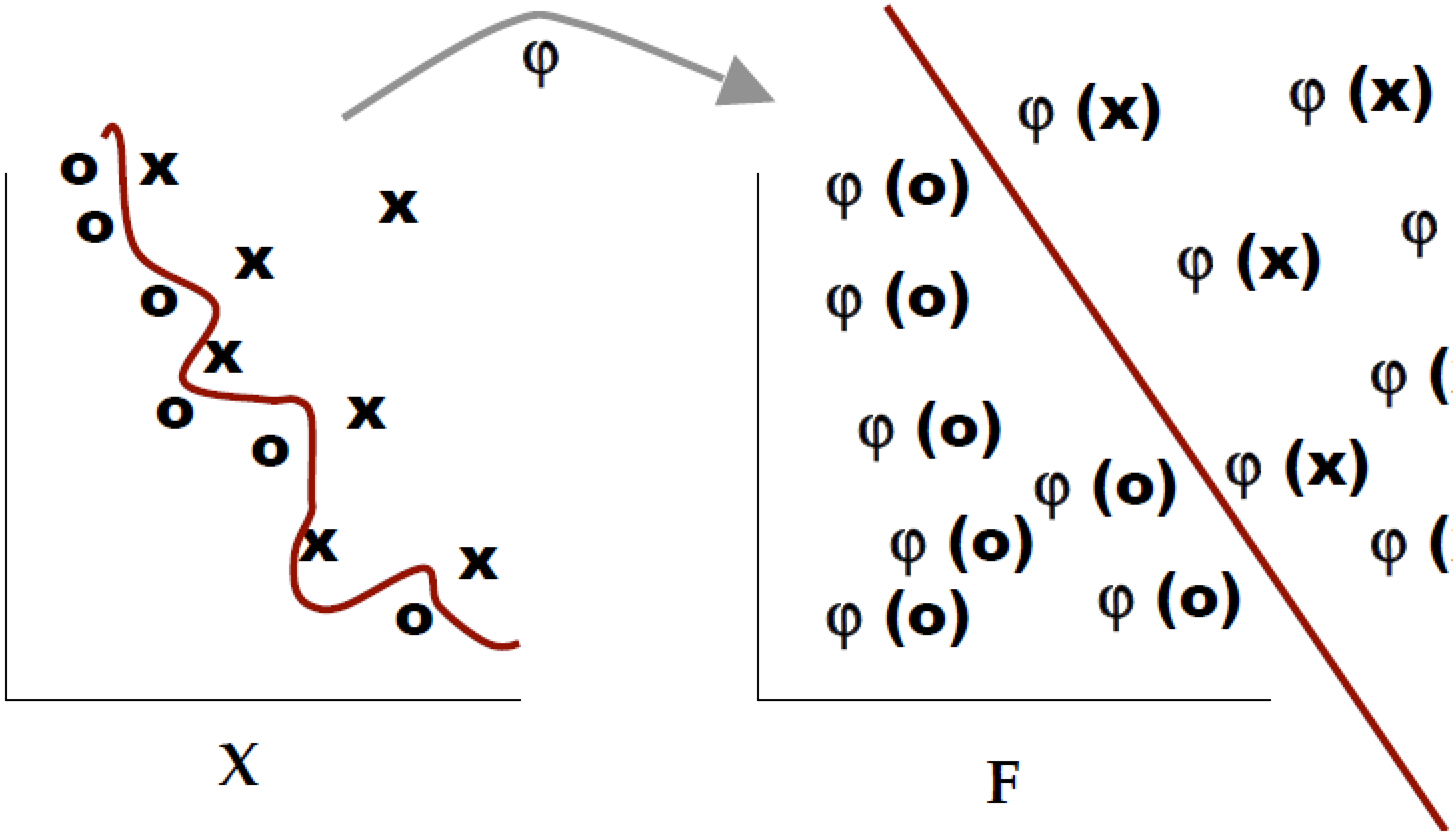
Identify the right hyper-plane (Scenario-5)





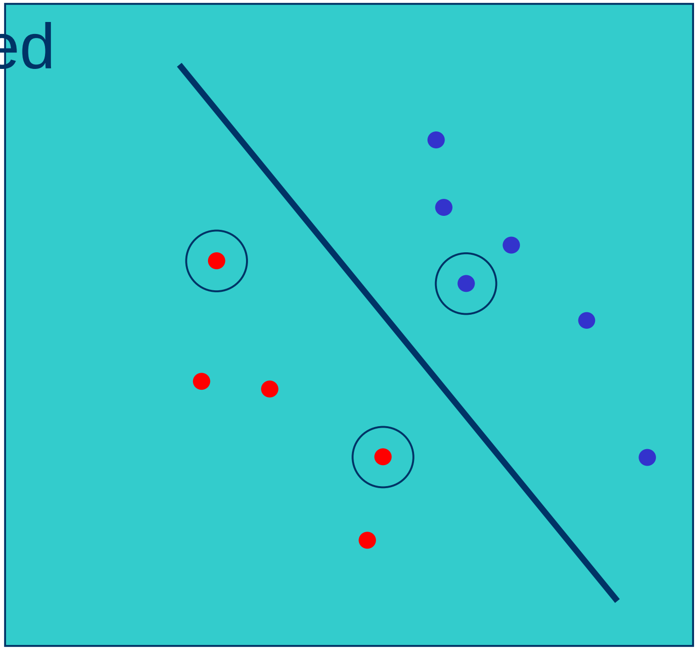
- z=x^2+y^2

Transformation to separate



Support Vector Machines

- The support vectors are indicated by the circles around them



- Datapoints in this subset are called “support vectors”

- It will be useful computationally if only a small fraction of the data points are support vectors,

- Since, we use the support vectors to decide which side of the separator a test case is on

General input/output for SVMs just like for neural nets, but for one important addition…

Input: set of (input, output) training pair samples; call the input sample features $x_1, x_2 \ldots x_n$, and the output result y. Typically, there can be lots of input features x_i.

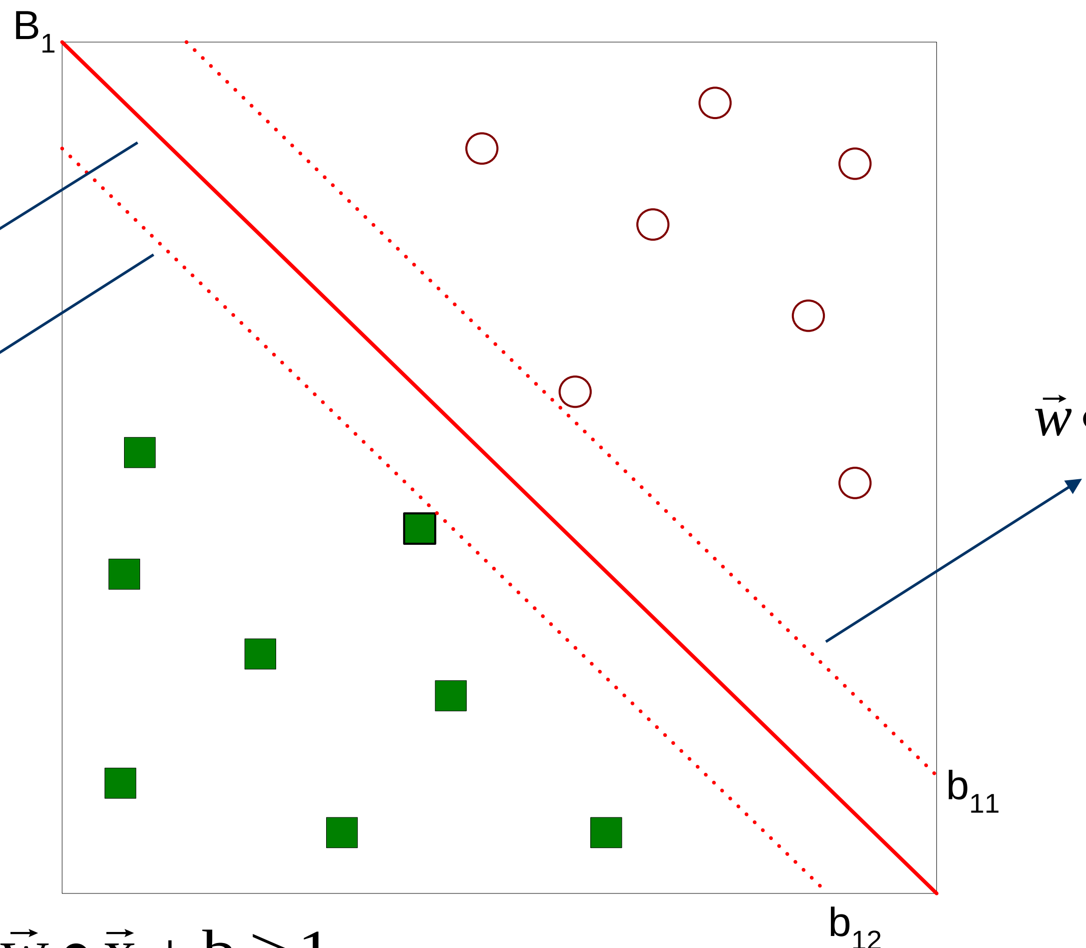
Output: set of weights $\mathbf{w}$ (or w_i), one for each feature, whose linear combination predicts the value of y. (So far, just like neural nets…)

Important difference: we use the optimization of maximizing the margin ('street width') to reduce the number of weights that are nonzero to just a few that correspond to the important features that 'matter' in deciding the separating line(hyperplane)…these nonzero weights correspond to the support vectors (because they 'support' the separating hyperplane)

Support Vector Machines

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

$$\vec{w} \bullet \vec{x} + b = -1$$



$$\vec{w} \bullet \vec{x} + b = +1$$

$$f(\vec{x}) = \begin{cases} 1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}} + \mathrm{b} \geq 1 \\ -1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}} + \mathrm{b} \leq -1 \end{cases}$$

$$\text{Margin} = \frac{2}{\|\vec{w}\|^2}$$

Support Vector Machines

- We want to maximize: $$\text{Margin} = \frac{2}{\|\vec{w}\|^2}$$
  - Which is equivalent to minimizing: $$L(w) = \frac{\|\vec{w}\|^2}{2}$$
  - But subjected to the following constraints:

$$f(\vec{x}_i) = \begin{cases} 1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_\mathrm{i} + \mathrm{b} \geq 1 \\ -1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_\mathrm{i} + \mathrm{b} \leq -1 \end{cases}$$

- This is a constrained optimization problem
  - Numerical approaches to solve it (e.g., quadratic programming)

We now must solve a <u>quadratic</u> programming problem

- Problem is: <u>minimize</u> $||\mathbf{w}||$, **s.t.** discrimination boundary is obeyed, i.e., min $f(x)$ s.t. $g(x)=0$, which we can rewrite as:
  min f: $\frac{1}{2} ||w||^2$ (Note this is a <u>quadratic</u> function)
  s.t. g: $y_i(\mathbf{w} \cdot \mathbf{x}_i) - \mathbf{b} = \mathbf{1}$ or $[y_i(\mathbf{w} \cdot \mathbf{x}_i) - \mathbf{b}] - \mathbf{1} = \mathbf{0}$

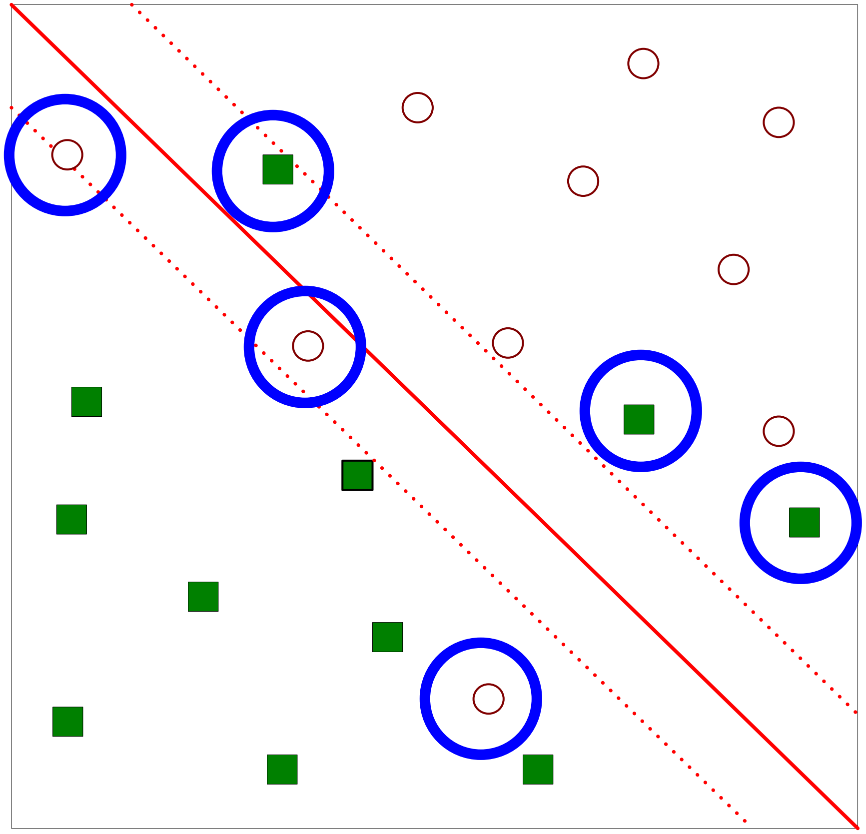
This is a **<u>constrained optimization problem</u>**

It can be solved by the Lagrangian multipler method

Because it is <u>quadratic</u>, the surface is a paraboloid, with just a single global minimum (thus avoiding a problem we had with neural nets!)

Support Vector Machines

- What if the problem is not linearly separable?



Support Vector Machines

- Introduce slack variables
  - Need to minimize:

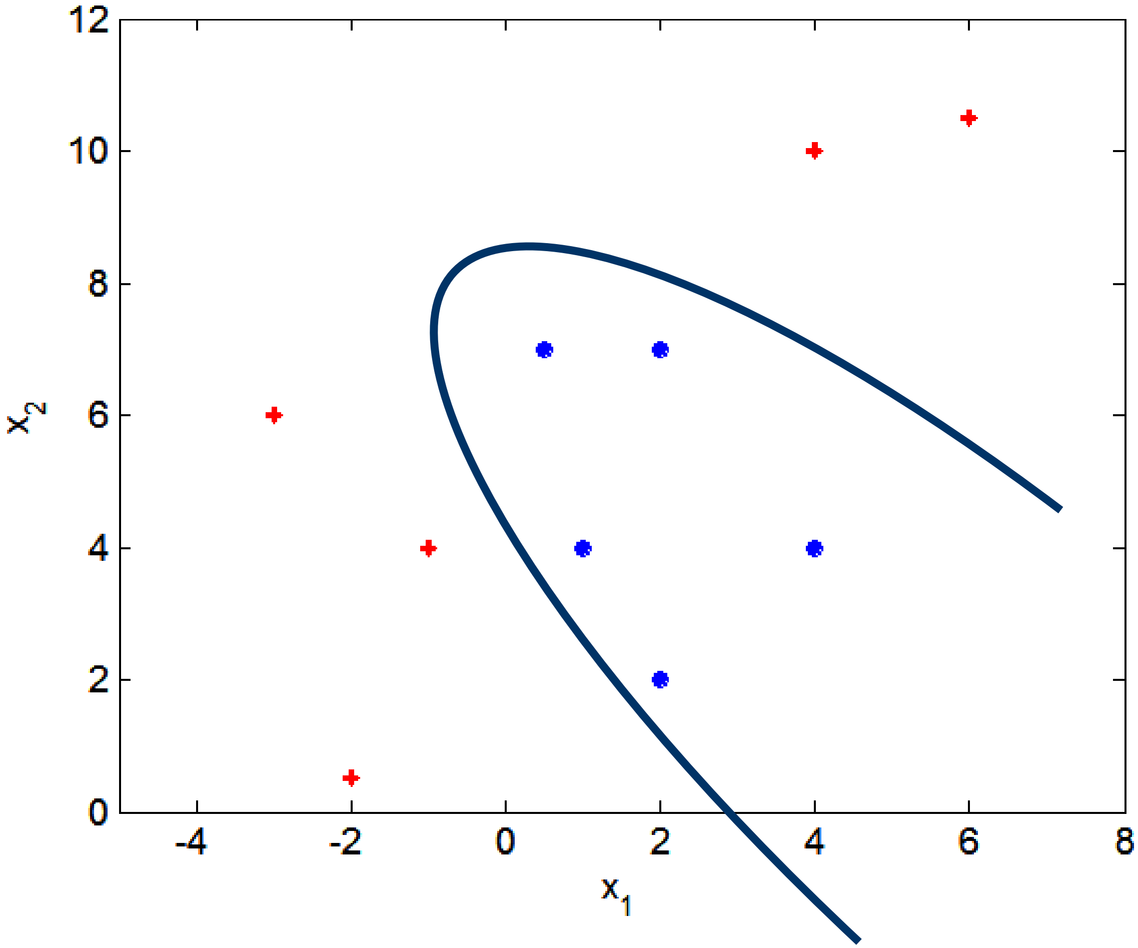
$$L(w) = \frac{\|\vec{w}\|^2}{2} + C\left(\sum_{i=1}^{N} \xi_i^k\right)$$

  - Subject to:

$$f(\vec{x}_i) = \begin{cases} 1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_{\mathrm{i}} + \mathrm{b} \geq 1 - \xi_{\mathrm{i}} \\ -1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_{\mathrm{i}} + \mathrm{b} \leq -1 + \xi_{\mathrm{i}} \end{cases}$$

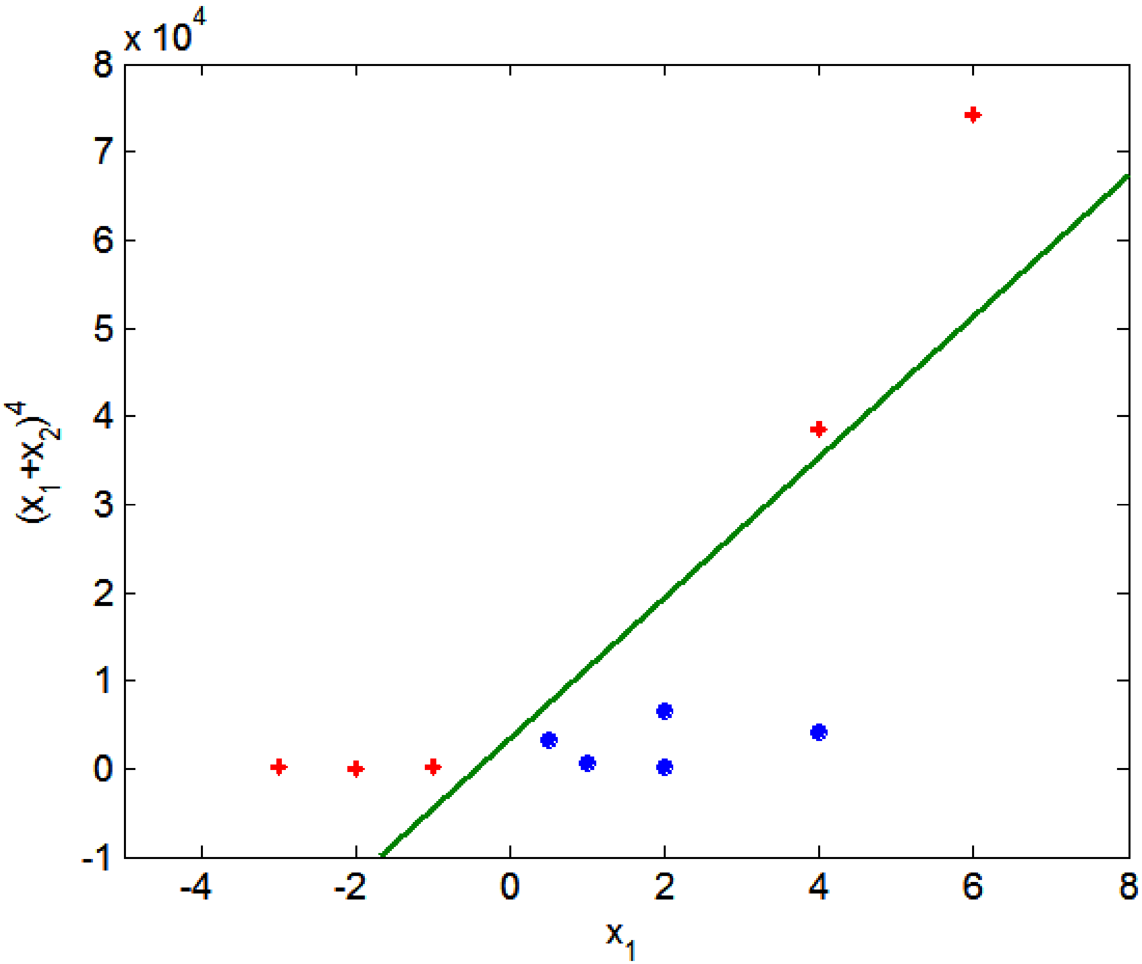
Nonlinear Support Vector Machines

- What if decision boundary is not linear?



Nonlinear Support Vector Machines

- Transform data into higher dimensional space



Nonlinear Support Vector Machines

- Kernel Trick
  - SVM has a technique called the kernel **trick**
  - These are functions which takes low dimensional input space and transform it to a higher dimensional space
  - i.e. it converts not separable problem to separable problem, these functions are called kernels
  - It is mostly useful in non-linear separation problem
  - http://crsouza.com/2010/03/17/kernel-functions-for-machine-learning-applications/

References

- Introduction to Data Mining by Tan, Steinbach, Kumar (Lecture Slides)
- https://www.analyticsvidhya.com/blog/2015/10/understaing-support-vector-machine-example-code/
- http://www.svm-tutorial.com/2014/11/svm-understanding-math-part-1/

Questions!