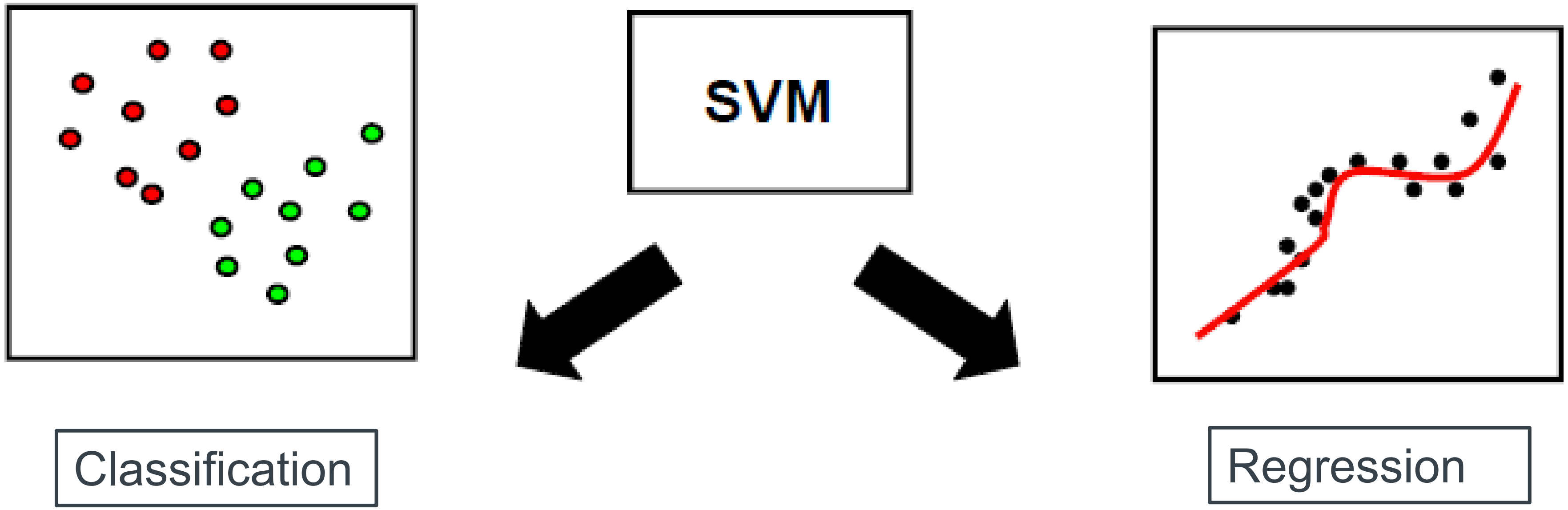
Introduction to Machine Learning

Machine Learning - II

GEORGETOWN UNIVERSITY McDonough SCHOOL of BUSINESS

SUPPORT VECTOR MACHINES

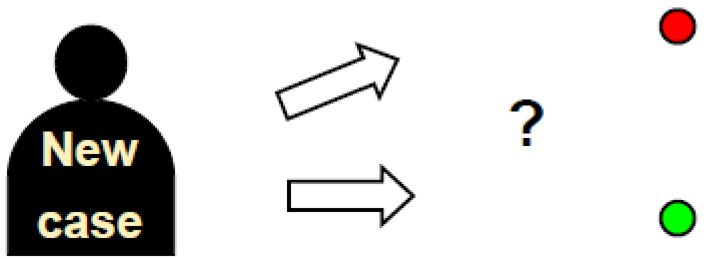
Support Vector Machines



SVM: We perform classification by finding the hyper-plane (~line) that differentiates the classes very well

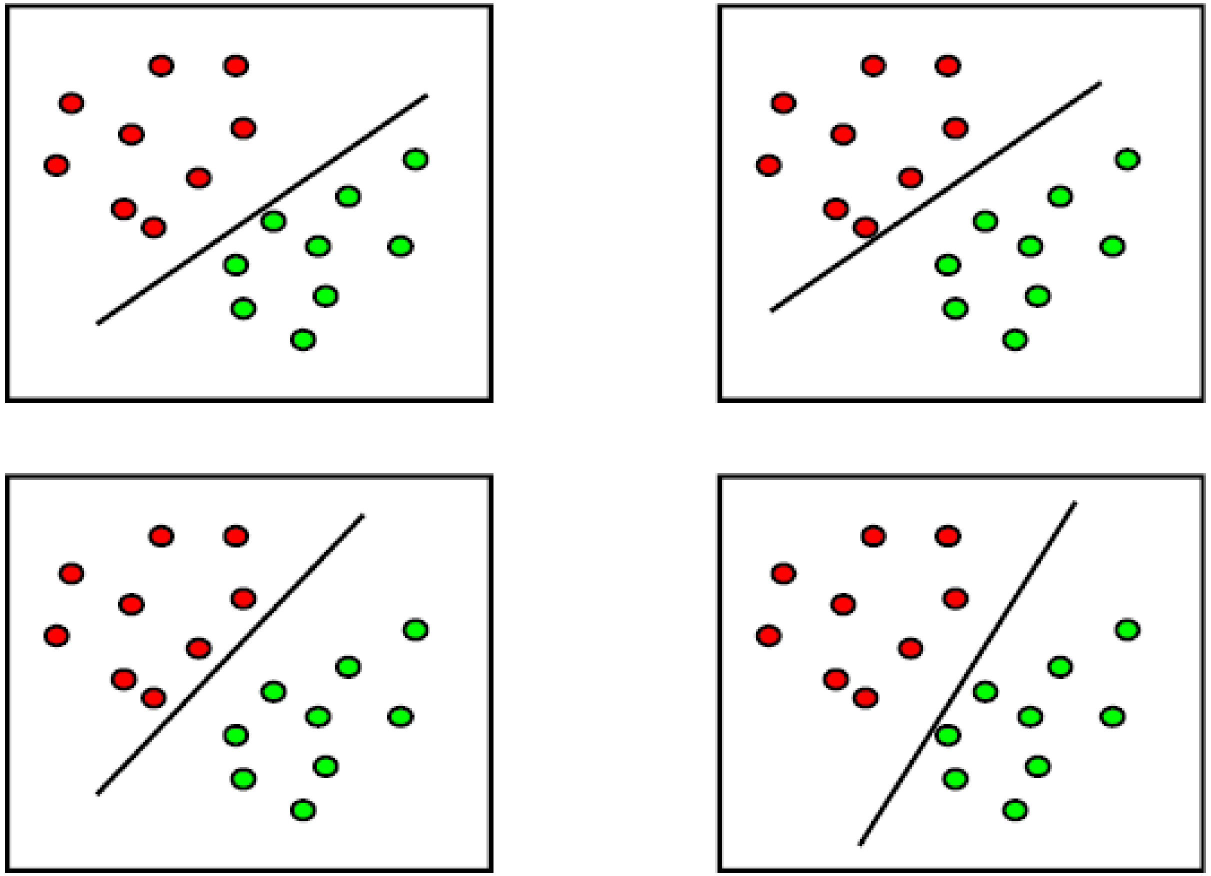
Classification: Starting Point

- Training dataset: patients with known diagnosis
- Input variables: data about patients
- Response variable: two diseases (y coded as +1, -1)
- Classification function: diagnosis = f(new patient) – scoring new patients using fitted model)



Classify Red vs Green Target Categories

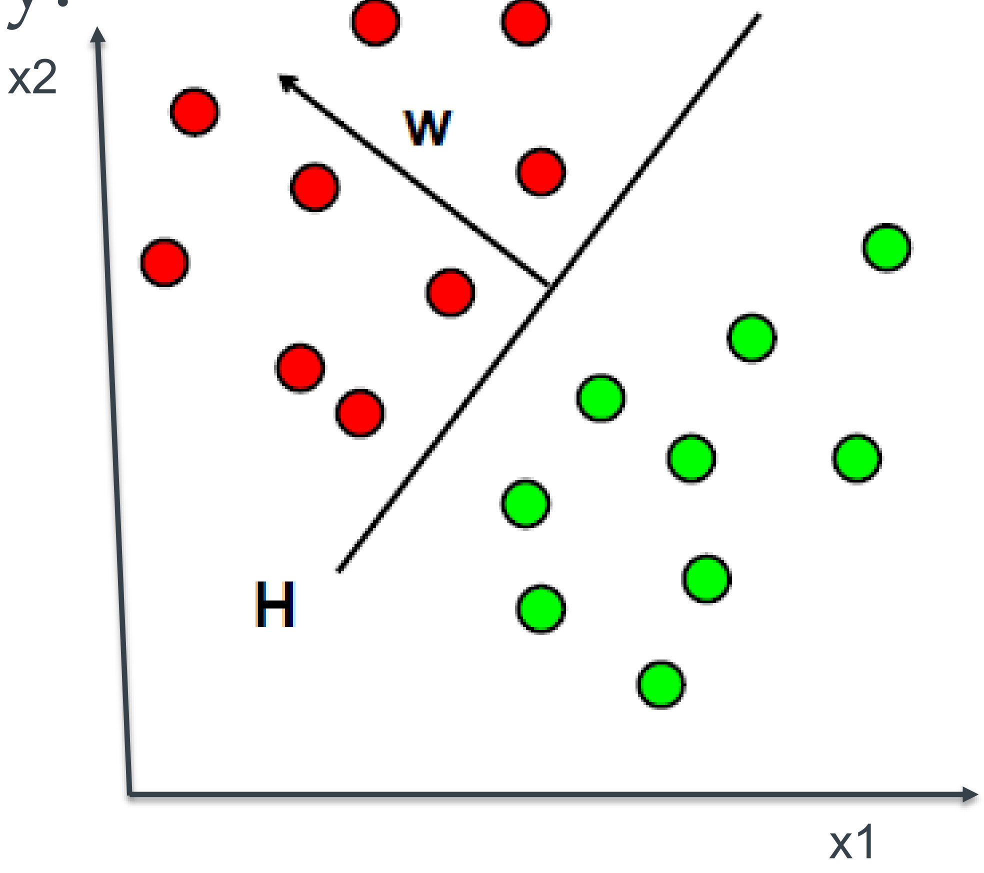
- The goal of Support Vector Machines is to identify a boundary or partition so that observations are separated well on the target classes



Linear Separation of Training Data

- A separating hyperplane H is given by:
  - the normal vector w,
  - an additional parameter, b, called bias

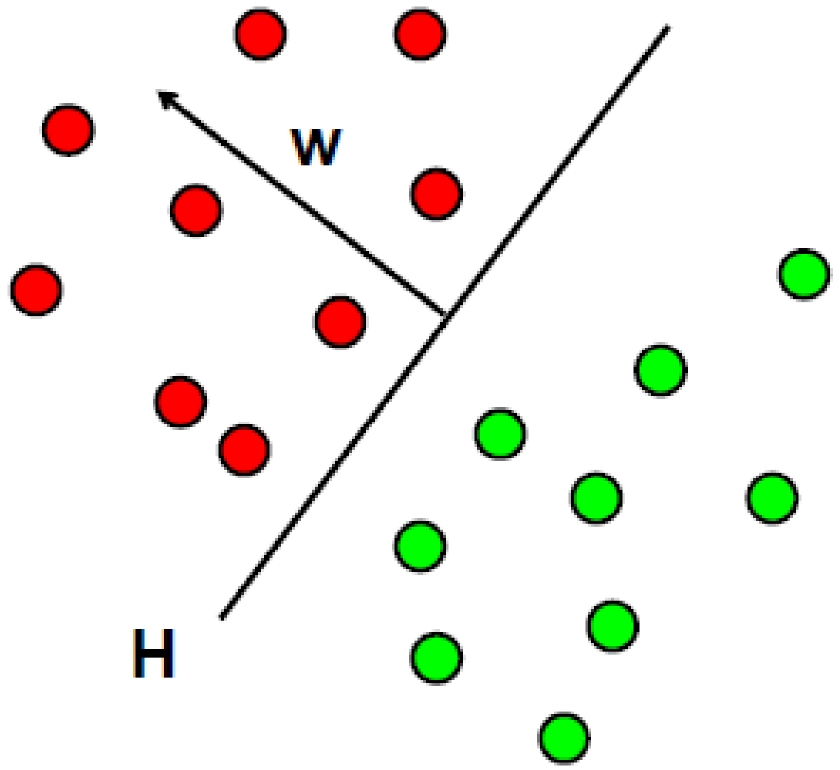
$$H = \{x | \underbrace{\langle w, x \rangle}_{\text{Dot product}} + b = 0\}$$



Given x variable, H represents what is the probability you will assign to each of the observations

Training vs. Prediction

Positive (should be your category of interest)



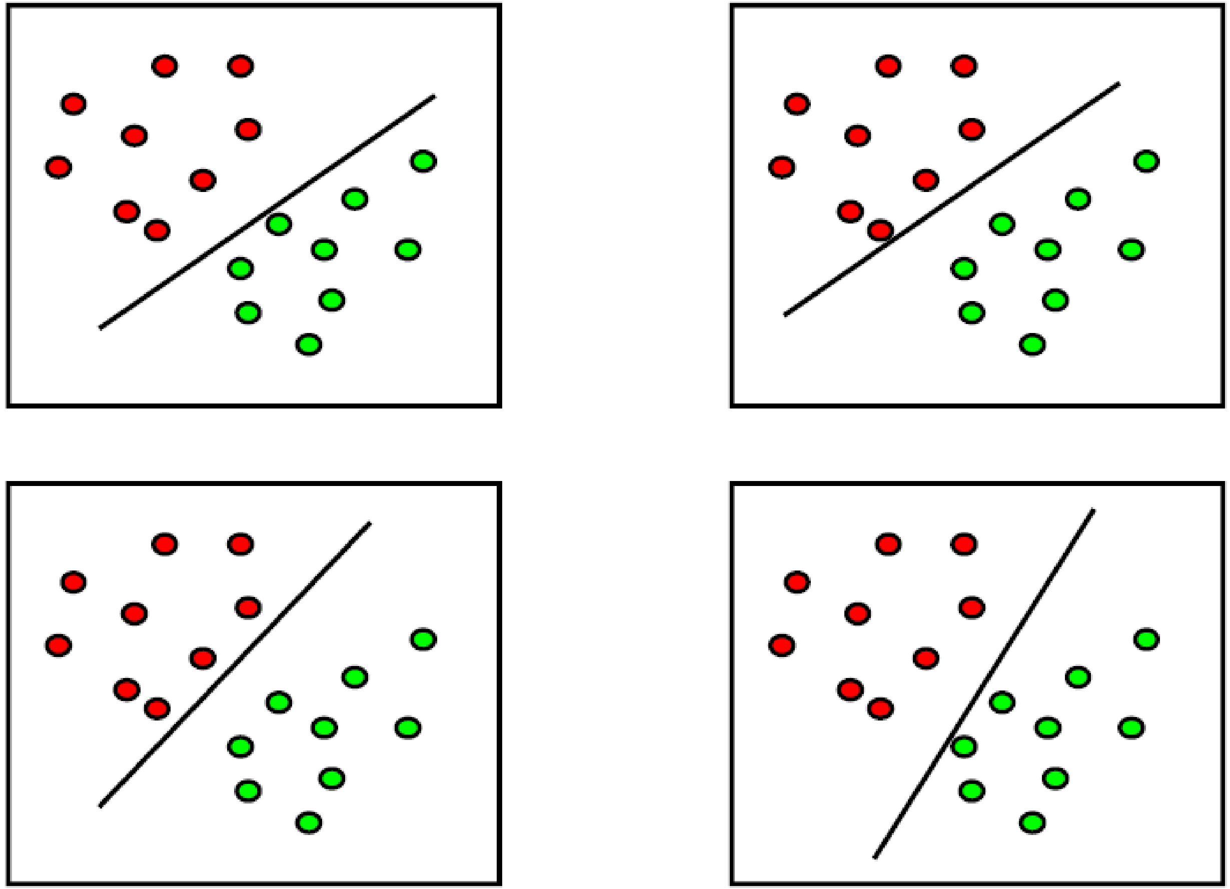
- Training:
  - Select w and b in such a way that the hyperplane separates the training data – that is, construction of a hyperplane
- Prediction of the class for a new patient:
  - On which side of the hyperplane is the new data point located?

OPTIMAL HYPERPLANE

Which Hyperplane is the Best One?

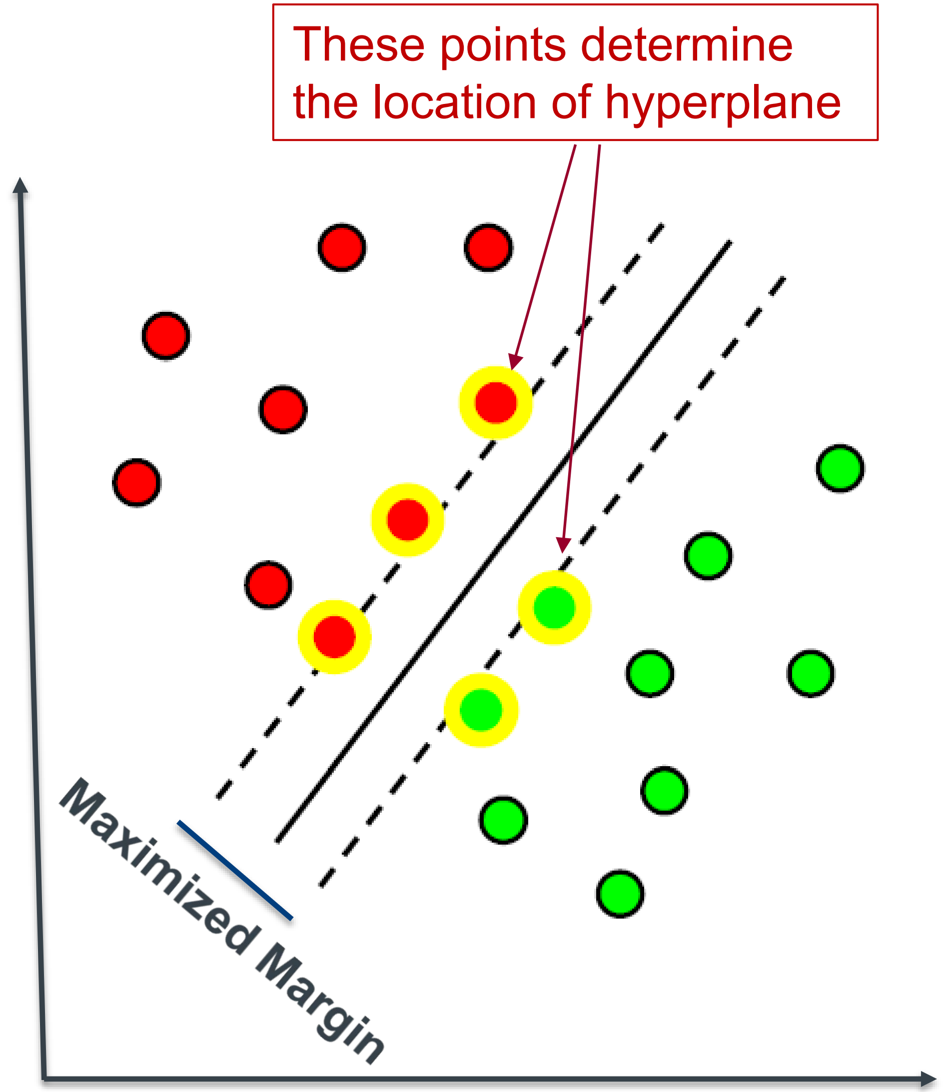
If the data points are *linearly separable*, then infinite hyperplanes (classification rules) exist

Observations on the boundary of the line are called "support vectors" – they support the prediction ability of the model



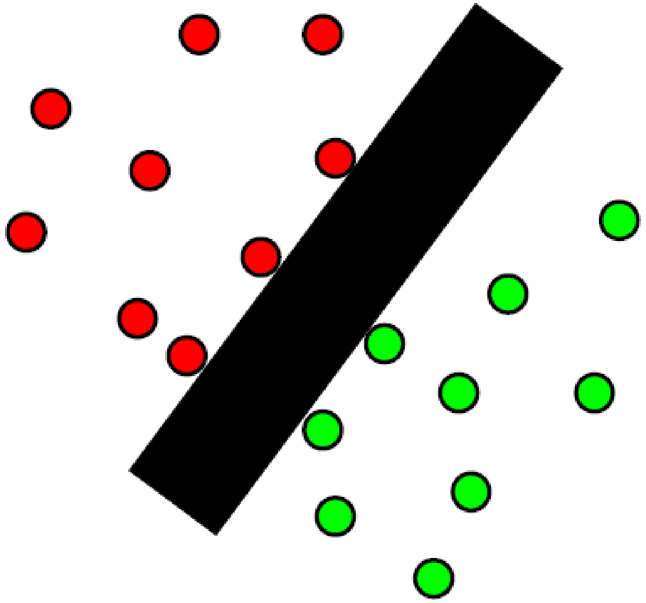
What are the Support Vectors?

- "Carrying vectors"
- The points, located closest to the hyperplane
- Determining the location of the hyperplane

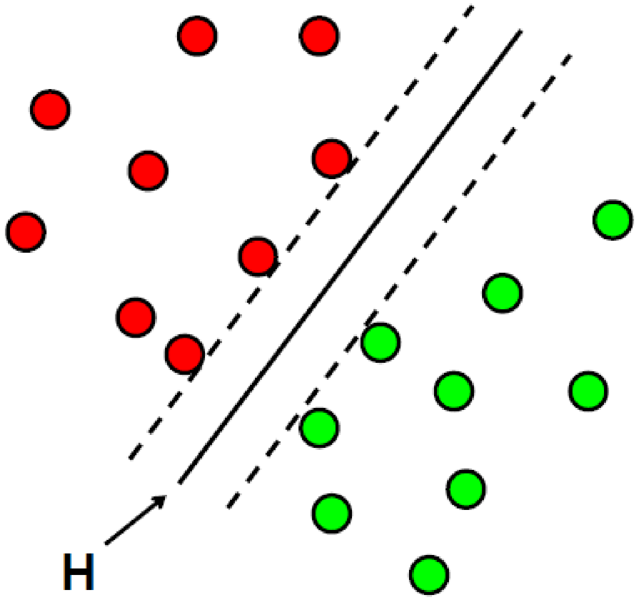


A "Wide" and a Maximum-Margin Hyperplane

A wide hyperplane is the best line (more forgiving for error)



Among all the hyperplanes, one of them has the maximum margin

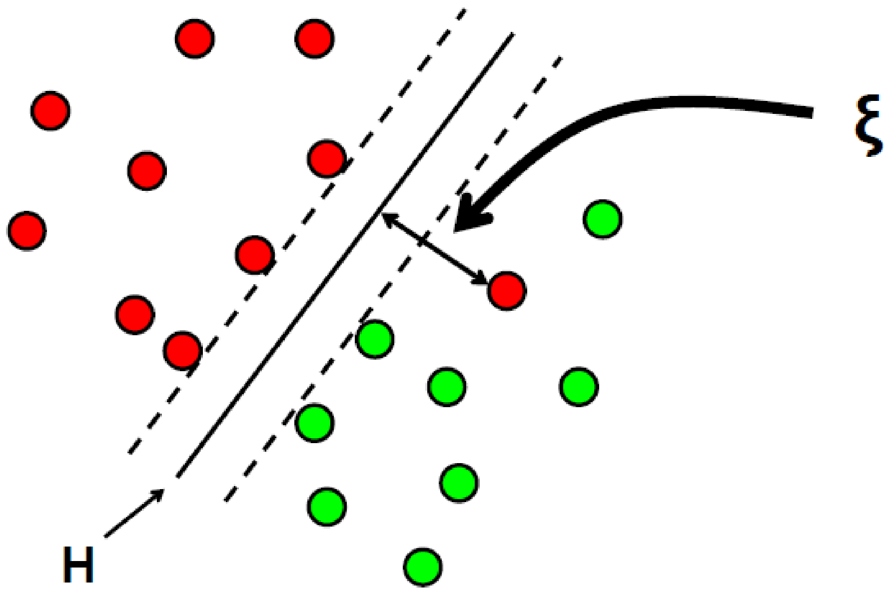


LINEAR CLASSIFIER AND MESSY DATA

Training Data not Linearly Separable

- If the datapoints are not linearly separable, we have a *soft margin* hyperplane
- SVM allows one to specify how many errors are acceptable by the model
- You can provide a parameter called C that helps tradeoff between:
  - Having a wide margin
  - Correctly classifying data

  A high value of C implies you want less errors
- C is an error weight (regularization parameter)



Penalty: C* (Distance to hyperplane) $= C \,.\, \sum_i \xi_i$

Training Data not Linearly Separable

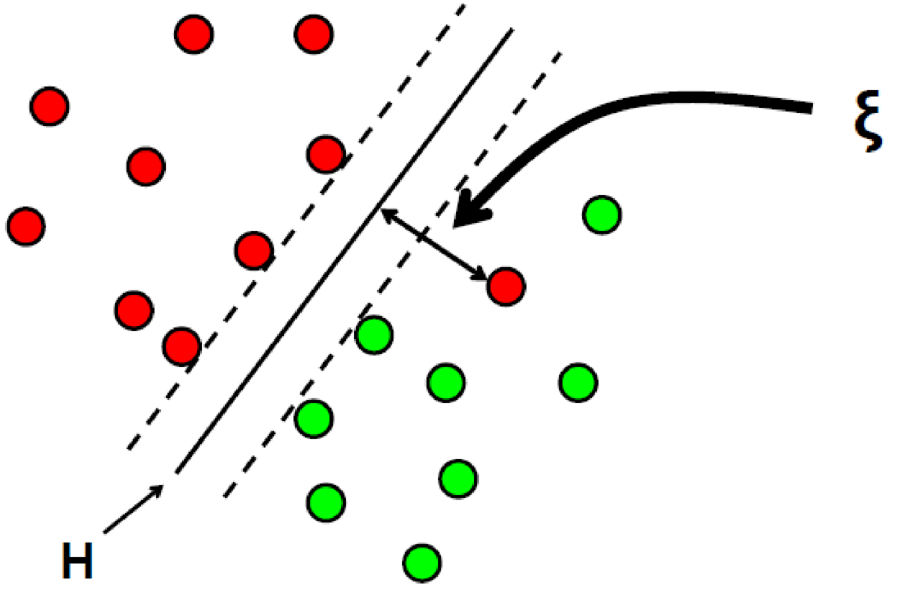
- Penalty: C* (Distance to hyperplane)− $\mathrm{C} \cdot \sum_i \xi_i$

- Optimization:

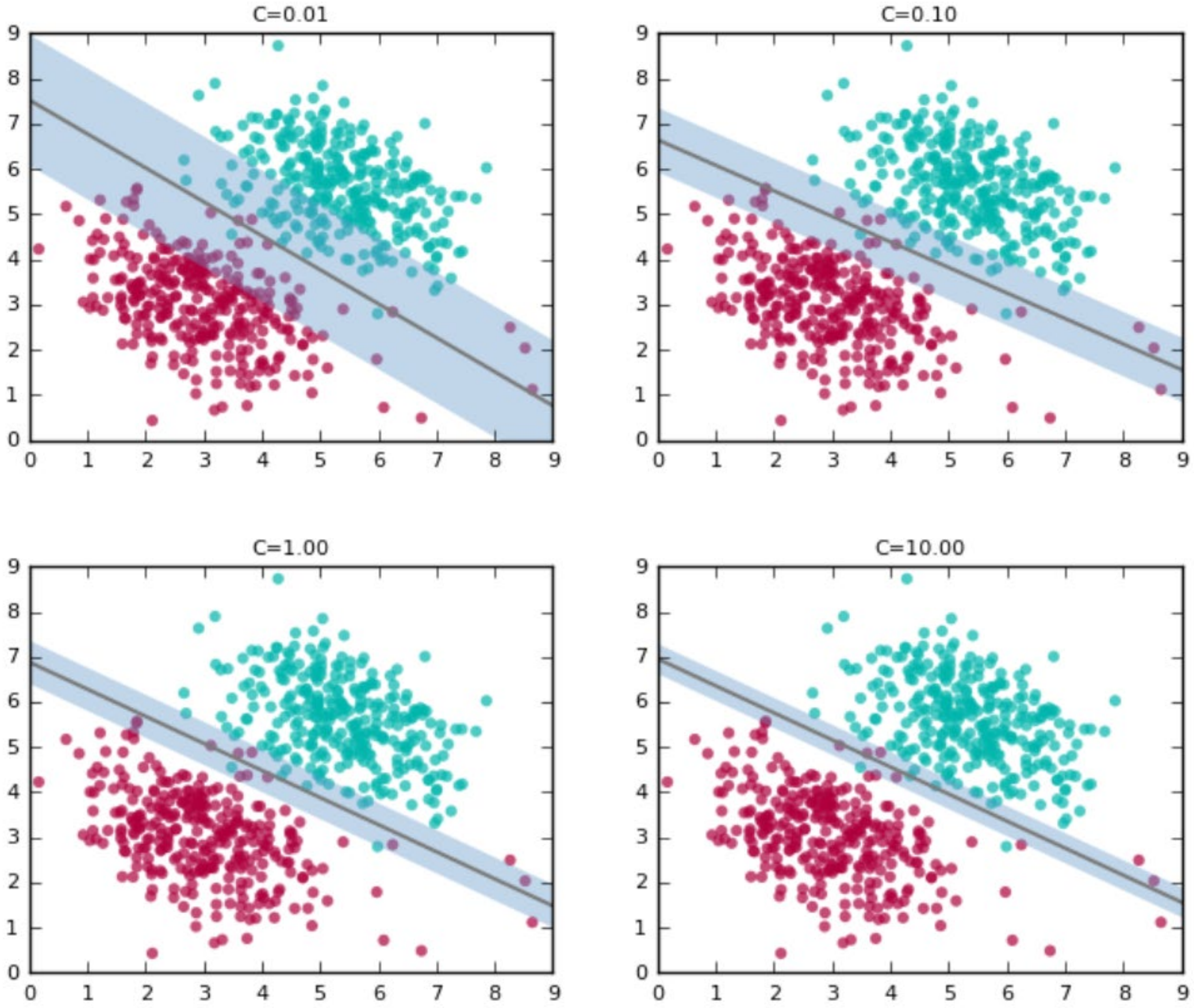
$$\boxed{Minimize\ \|w\|^2 + \mathrm{C} \cdot \textstyle\sum_i \xi_i}$$

under the condition

$$y_i \cdot (\langle w, x_i \rangle + b) \geq 1 - \xi_i, \qquad \xi_i \geq 0$$



Regularization Parameter C



The plots you see here show how the classifier and margin vary as we increase the value of C

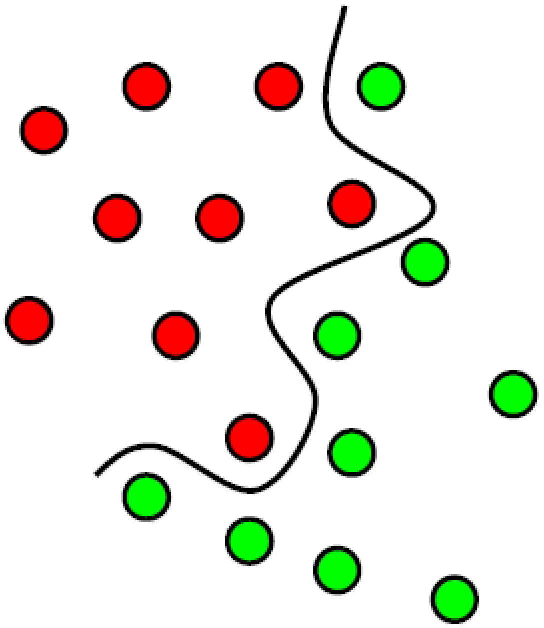
Note how the width of the margin shrinks as we increase the value of C

Deciding on a good value of C: use cross-validation

NON LINEARLY SEPARABLE DATA WITH SVM

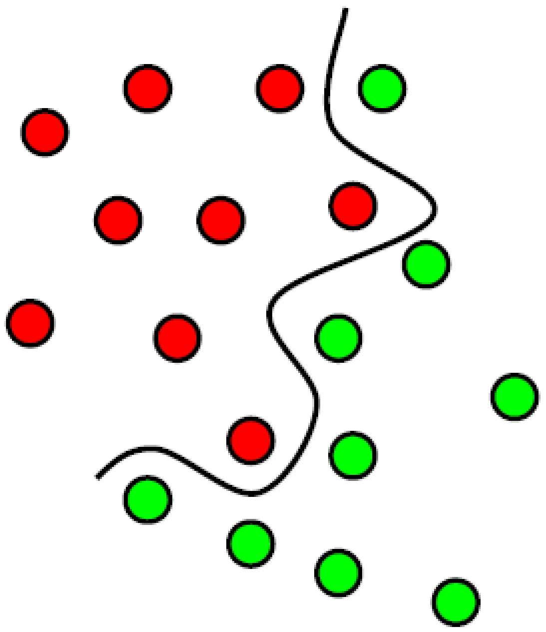
Problem: Not Linearly Separable Data Points

Input space 2-D



Idea: Feature Space

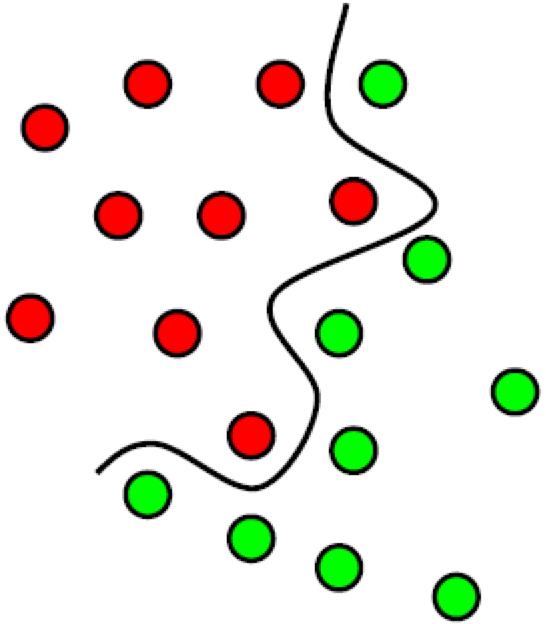
Input space 2-D



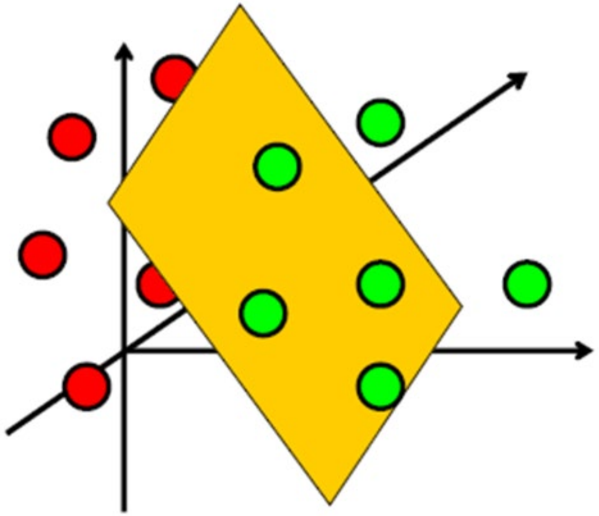
- A nonlinear transformation of the input variables into a high-dimensional feature space
- The maximum-margin hyperplane is constructed in the high-dimensional feature space

Solution: A Kernel Trick

Input space 2-D



Feature space 3-D



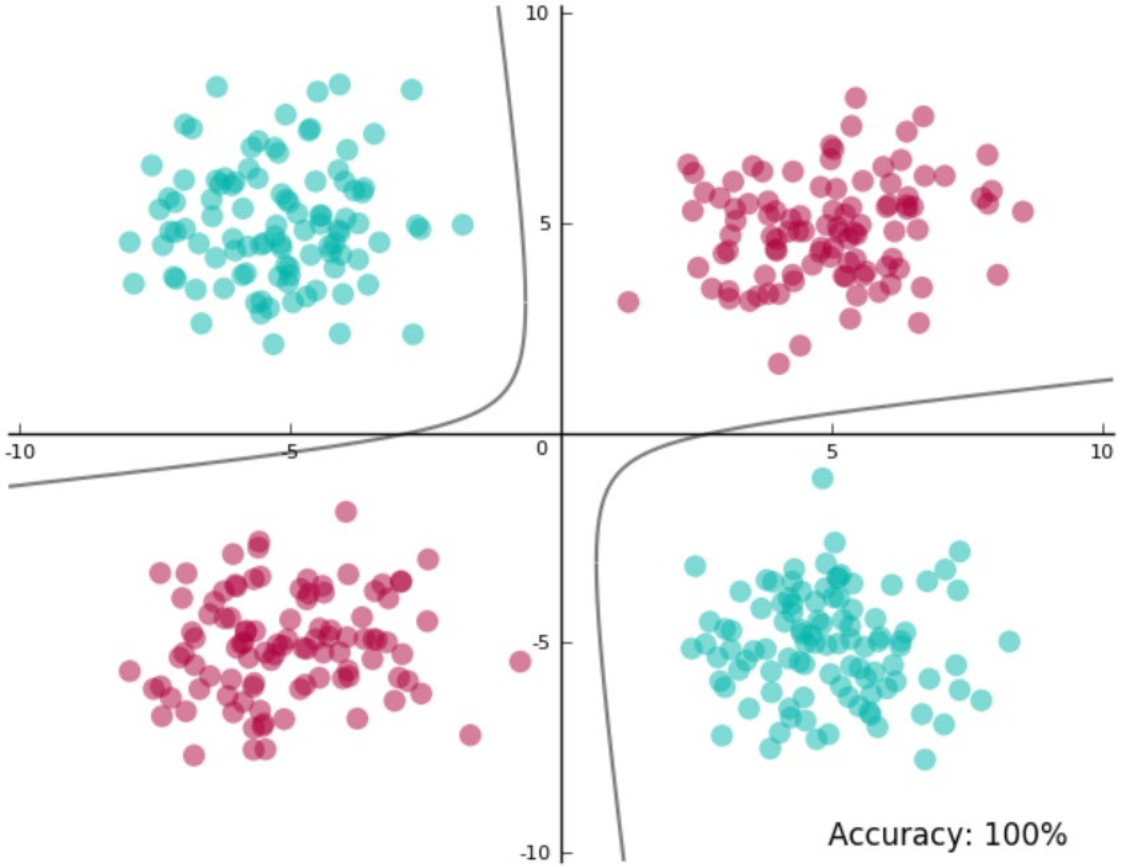
We start with the original dataset, and project it into three-dimensional space where the new coordinates could be (e.g.,):

$X_1 = x_1^2$

$X_2 = x_2^2$

$X_3 = \sqrt{2\,(x_1 x_2)}$

SVM in the Feature Space to Original Two-Dimensional Space



The shape of the separating boundary in the original space depends on the projection

Analysis

- How do you know what space to project data into? – Difficult to know
  - Data is more likely to be linearly separable when projected into higher dimensions (try out a few high dimensional projections)
- Ask the SVM to do the projection: SVMs use something called *kernels* to do these projections and they are computationally very fast

So Far…

1. For linearly separable data SVMs work amazingly well
2. For data that's almost linearly separable, SVMs can still be made to work pretty well by using the right value of C
3. For data that's not linearly separable, we can project data to a space where it is perfectly/almost linearly separable, which reduces the problem to 1 or 2

The Kernel Trick

- We don’t have to worry about exact projections
- We could write number of dimensions as dot products between various data points (represented as vectors)
- For *p*-dimensional vectors *i* and *j* where the first subscript on a dimension identifies the point and the second indicates the dimension number:

$$\overrightarrow{x_i} = (x_{i1}, x_{i2}, \dots, x_{ip})$$

$$\overrightarrow{x_j} = (x_{j1}, x_{j2}, \dots, x_{jp})$$

- The dot product is:

$$\overrightarrow{x_i} \cdot \overrightarrow{x_j} = (x_{i1}x_{j1}, x_{i2}x_{j2}, \dots, x_{ip}x_{jp})$$

The Kernel Trick

- A kernel, short for *kernel function*, takes as input two observations in the original space, and directly gives us the dot product in the projected space
- Revisiting last Projection, for observation i on two variables x_1 and x_2:

$$\overrightarrow{x_i} = (x_{i1}, x_{i2})$$

- Corresponding projected point was: $\overrightarrow{X_i} = (x_{i1}^2, x_{i1}^2, \sqrt{2\,(xi_1 x_{i2})}\ \ )$

Important Observation

- Dual optimization problem

$$W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle$$

- Classification function

$$f(x_{new}) = sign\left(\sum_{i=1}^{n} \alpha_i y_i \langle x_i, x_{new} \rangle + b \right)$$

Dot product

The Kernel Trick

- We use a kernel function, living in the input space, but behaving as a dot product in the feature space

- Kernels take the data as inputs and transform into the required form
- Trick: we do not have to know $the\ feature\ space\ looks$ explicitly!

Examples of Kernel Functions

- Linear

$$\mathcal{K}(x_i, x_j) = \langle x_i, x_j \rangle$$

- Polynomial

$$\mathcal{K}(x_i, x_j) = (\gamma\langle x_i, x_j \rangle + k)^d$$

- Radial-Basis-Function

$$\mathcal{K}(x_i, x_j) = \exp(-\gamma\left\|x_i - x_j\right\|^2)$$

- Sigmoid

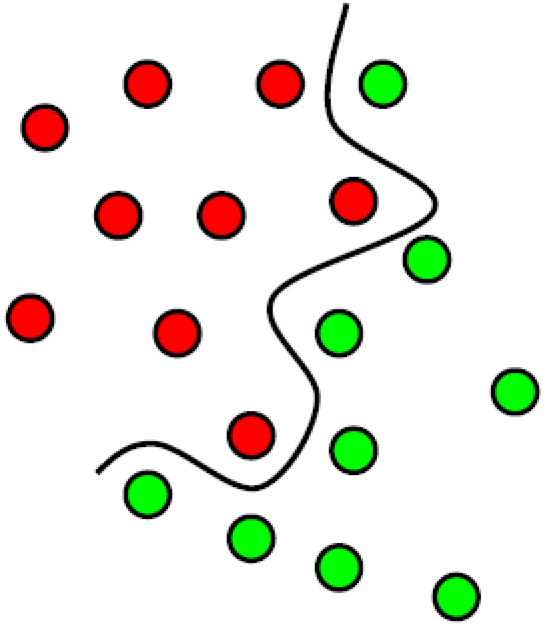
$$\mathcal{K}(x_i, x_j) = \tanh(\gamma \langle x_i, x_j \rangle\text{-b})$$

d= degree of polynomial

γ represents similarity measure and is sometimes parameterized as = $\frac{1}{2\sigma}$

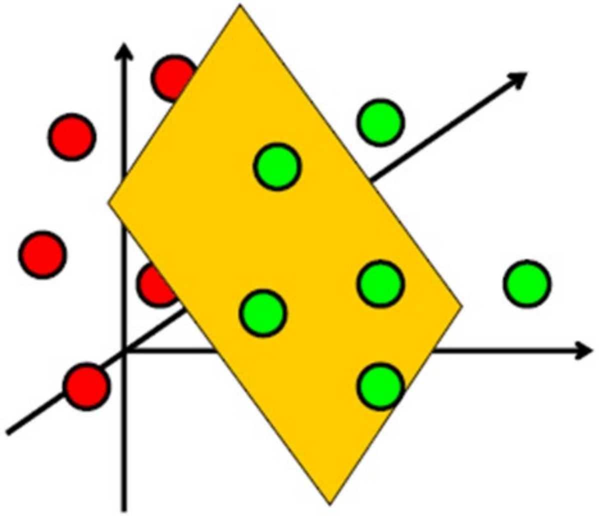
Kernel Trick

Input space 2-D



Nonlinear separation with kernel function

Feature space 3-D



Linear separation with dot product

Summary of SVM

A SVM is a hyperplane with a maximum-margin in a feature space, constructed by use of a kernel function in the input space

A kernel helps to find a hyperplane in the higher dimensional space without increasing computational cost if we are required to move to higher dimension

Parameters to tune:

- The penalty C (regularization term) for data that is not completely linearly separable
- The kernel function and its parameters

SVM APPLICATIONS

Applications: Mostly in Classification

- Cancer Diagnosis and Prognosis
- Text Classification: Emails into spam/good; news articles into topis, etc.
- Sales Forecasting and Customer Attrition Models
- Credit Scoring and Fraud Detection
- Facial Expression Classification
- …

Advantages and Disadvantages of SVMs

Advantages

- Finds a global unique minimum error
- The kernel trick
- Simple geometric interpretation
- Strong ability to generalize
- The complexity of calculation do not depend on the dimension of the input space: avoids curse of dimensionality

Disadvantages

- Which kernel function to apply?
- How to select the parameters of the kernel function?

SVM FOR REGRESSION

Support Vector Machines for Regression

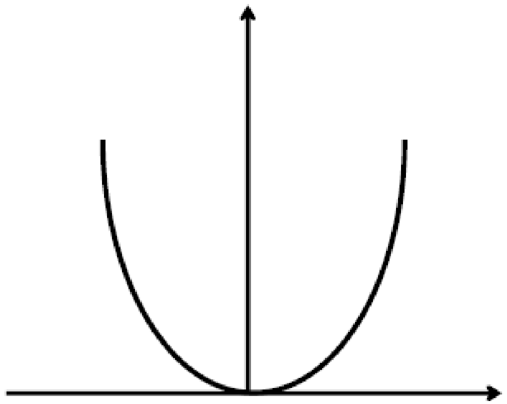
Classical Linear Regression

Estimate the Function

$f(x) = \langle w, x \rangle = b$

by minimizing

$$L\big(y, f(x)\big) = (f(x) - y)^2$$



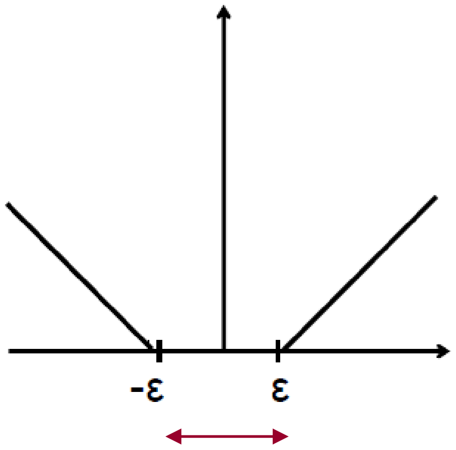
Support Vector Regression

Estimate the Function

$f(x) = \langle w, x \rangle = b$

by minimizing

$$L_g\big(y, f(x)\big) = |f(x) - y|_{\varepsilon}$$



ε-insensitive zone

Support Vector Regression (SVR)

- Our objective, is to consider the points that are within the decision boundary line
- Suppose these lines (decision boundary) are at a distance ‘a’ from the hyperplane (a is ϵ_i)

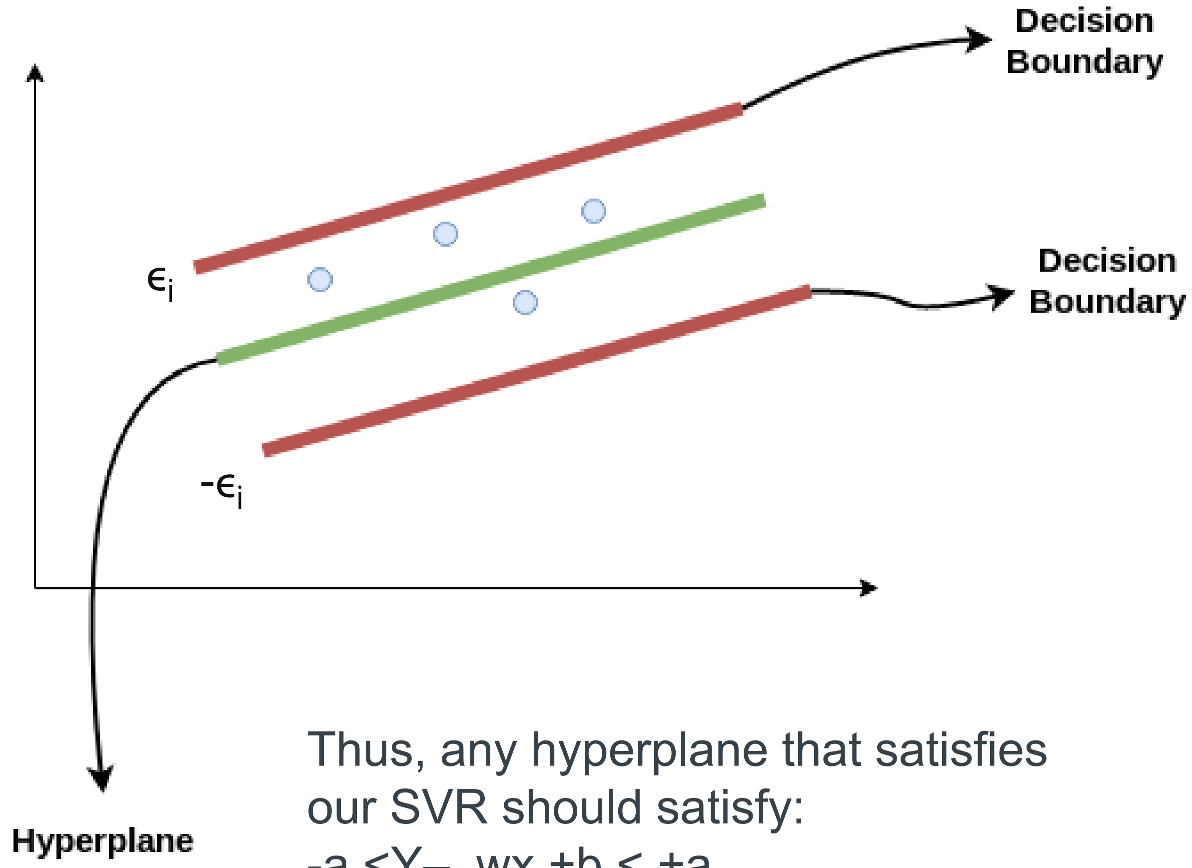
If equation to hyperplane:

H = wx + b

Equation for decision boundary:

wx+b = +a

wx+b = -a



Thus, any hyperplane that satisfies our SVR should satisfy:

-a <Y– wx +b < +a

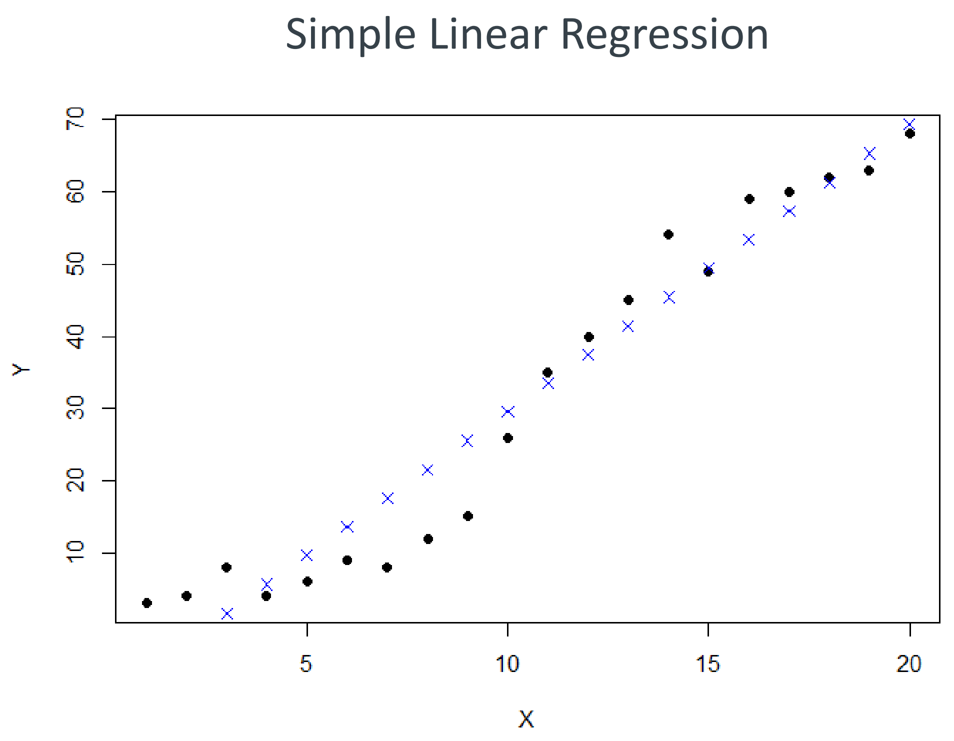
Support Vector Regression

- *Our main aim here is to decide a decision boundary at 'a' distance from the original hyperplane such that data points closest to the hyperplane or the support vectors are within that boundary line*

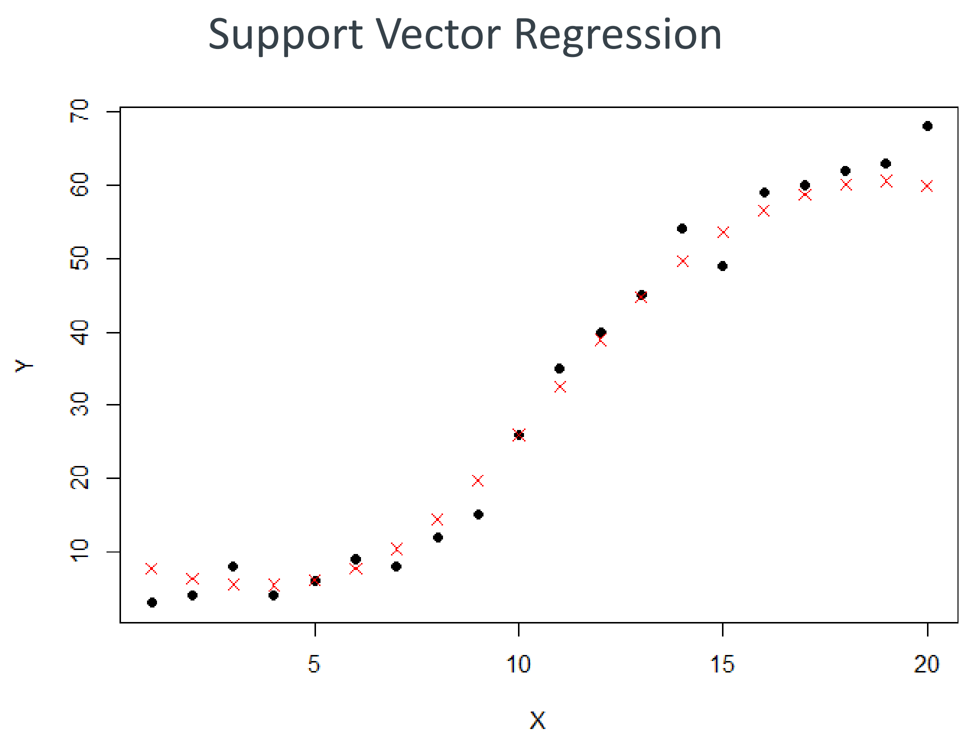
- Hence, take only those points that are within the decision boundary and have the least error rate, or are within the Margin of Tolerance for a better fitting model

Example of SLR vs. SVR for Illustration

```
X,Y
1,3
2,4
3,8
4,4
5,6
6,9
7,8
8,12
9,15
10,26
11,35
12,40
13,45
14,54
15,49
16,59
17,60
18,62
19,63
20,68
```



RMSE = 5.70



RMSE = 3.157