UNC CHARLOTTE

The WILLIAM STATES LEE COLLEGE of ENGINEERING

# Introduction to ML
# Lecture 9: Support Vector Machine

Hamed Tabkhi
Department of Electrical and Computer Engineering,
University of North Carolina Charlotte (UNCC)
htabkhiv@uncc.edu

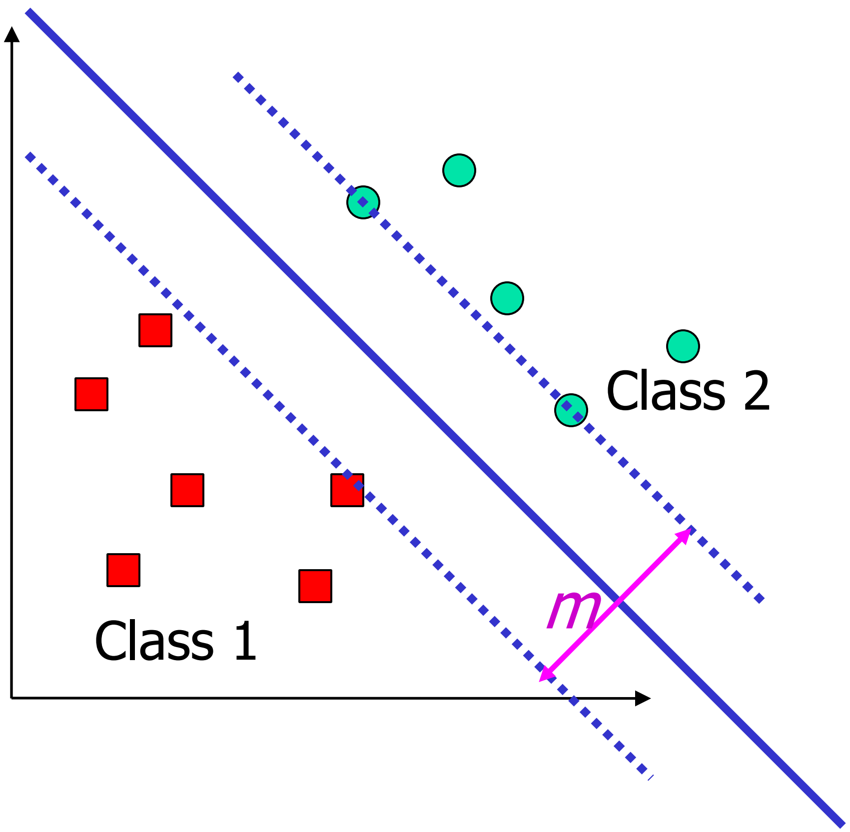
UNC CHARLOTTE

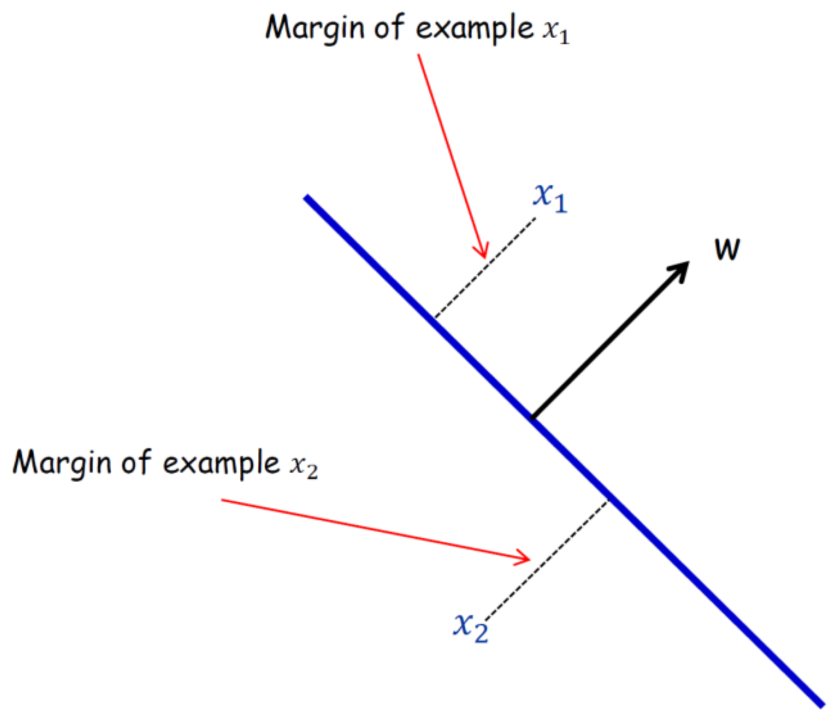
# Intuitive Definition

- A linear discriminative classifier would attempt to draw a straight line separating the two sets of data, and thereby create a model for classification.
- The intuition is this: rather than simply drawing a zero-width line between the classes, we can draw around each line a *margin* of some width, up to the nearest point.
- In support vector machines, the line that maximizes this margin is the one we will choose as the optimal model.
- Support vector machines are an example of such a *maximum margin* estimator.

# Good Decision Boundary: Margin Should Be Large

- The decision boundary should be as far away from the data of both classes as possible



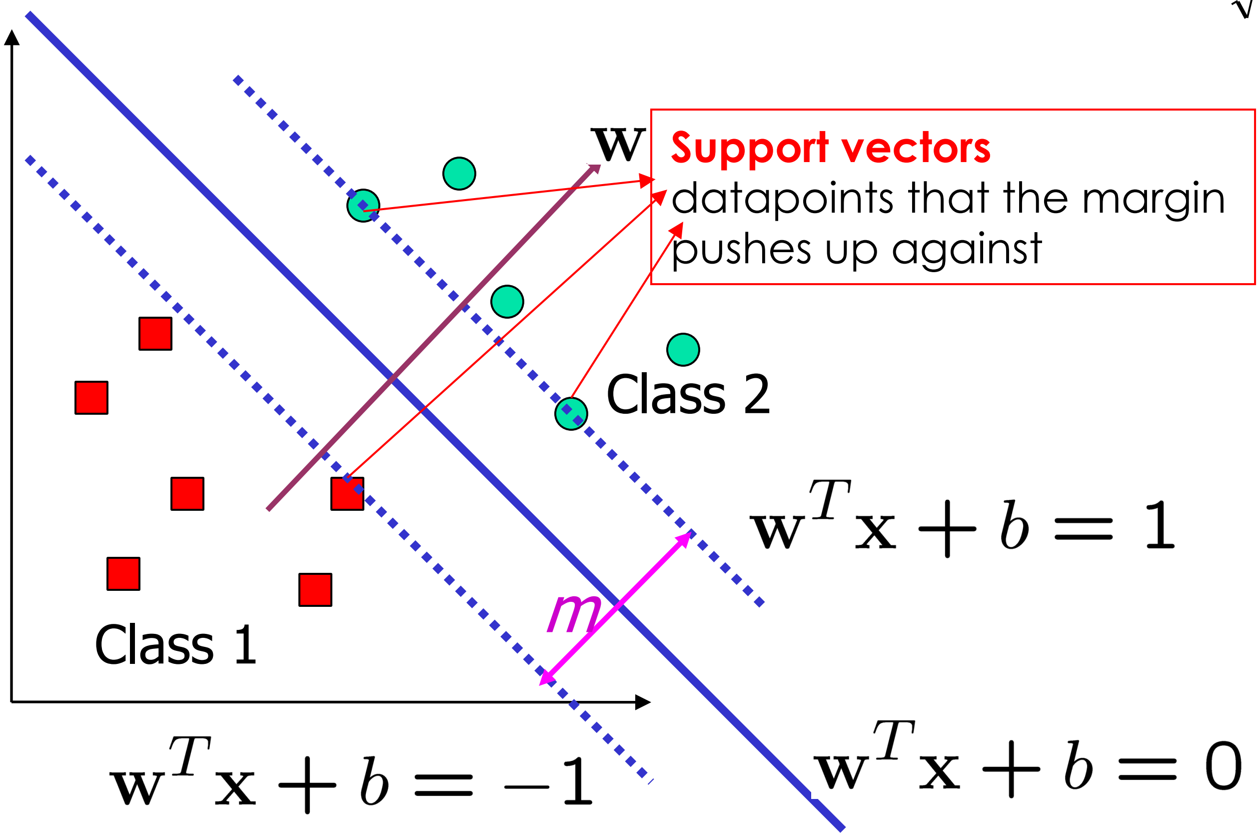




UNC CHARLOTTE

# Good Decision Boundary: Margin Should Be Large

- The decision boundary should be as far away from the data of both classes as possible
  - We should maximize the margin, $m$

$$m = \frac{2}{\sqrt{w.w}} \qquad m = \frac{2}{||\mathbf{w}||}$$




UNC CHARLOTTE

# The Optimization Problem

- Let $\{x_1, ..., x_n\}$ be our data set and let $y_i \in \{1,-1\}$ be the class label of $x_i$
- The decision boundary should classify all points correctly $\Rightarrow$

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \qquad \forall i$$

- A constrained optimization problem

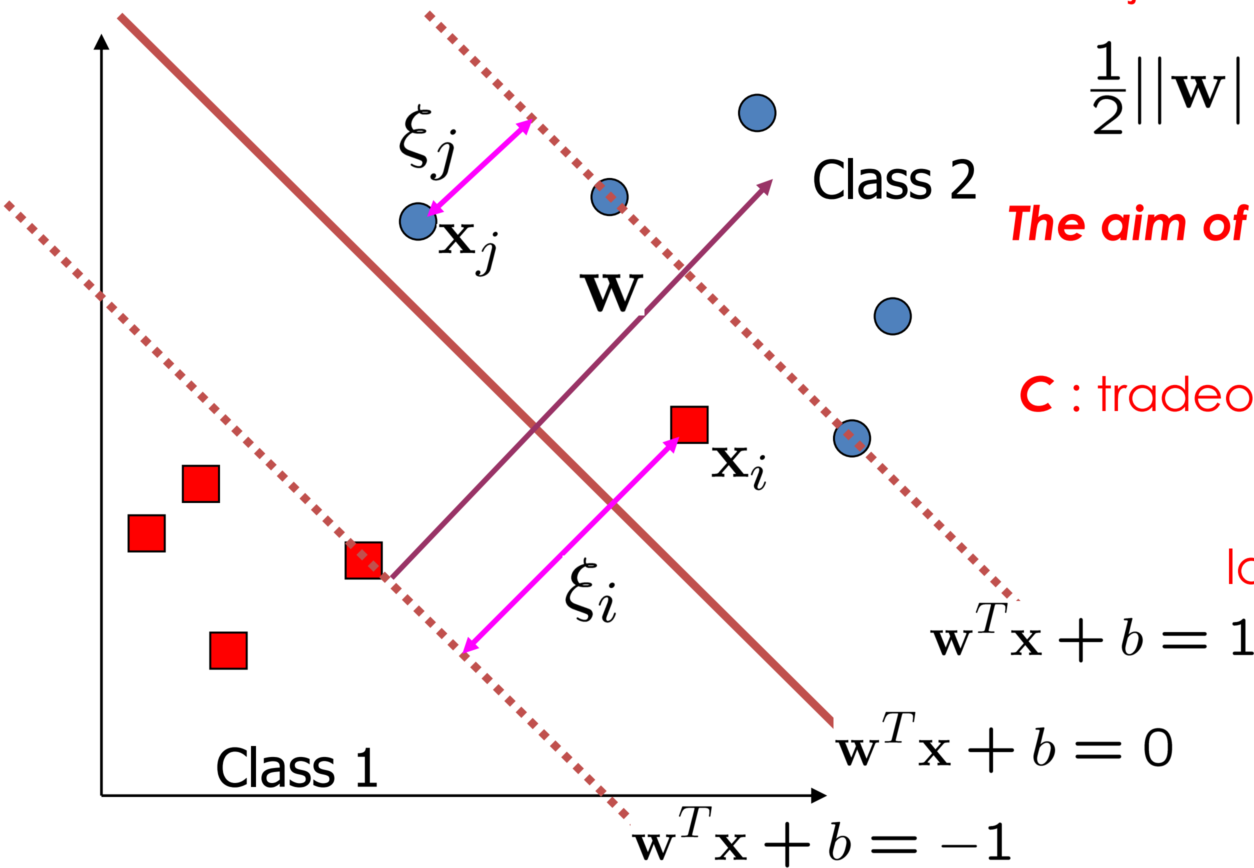
$$\text{Minimize } \frac{1}{2}||\mathbf{w}||^2$$

- $||\mathbf{w}||^2 = \mathbf{w}^T\mathbf{w}$

UNC CHARLOTTE

# Non-linearly Separable Problems

- We allow "error" $\xi_i$ in classification; it is based on the output of the discriminant function $\mathbf{w}^T\mathbf{x}+b$
- $\xi_i$ approximates the number of misclassified samples




New objective function:

$$\frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i$$

***The aim of SVM is to minimize this objective function.***

**C** : tradeoff parameter between error and margin; chosen by the user; large C means a higher penalty to errors

UNC CHARLOTTE

# The effect of C

- A small C will give a wider margin, at the cost of some misclassifications.

- A huge C will give the hard margin classifier and tolerates zero constraint violation -> Less Generalization Opportunities

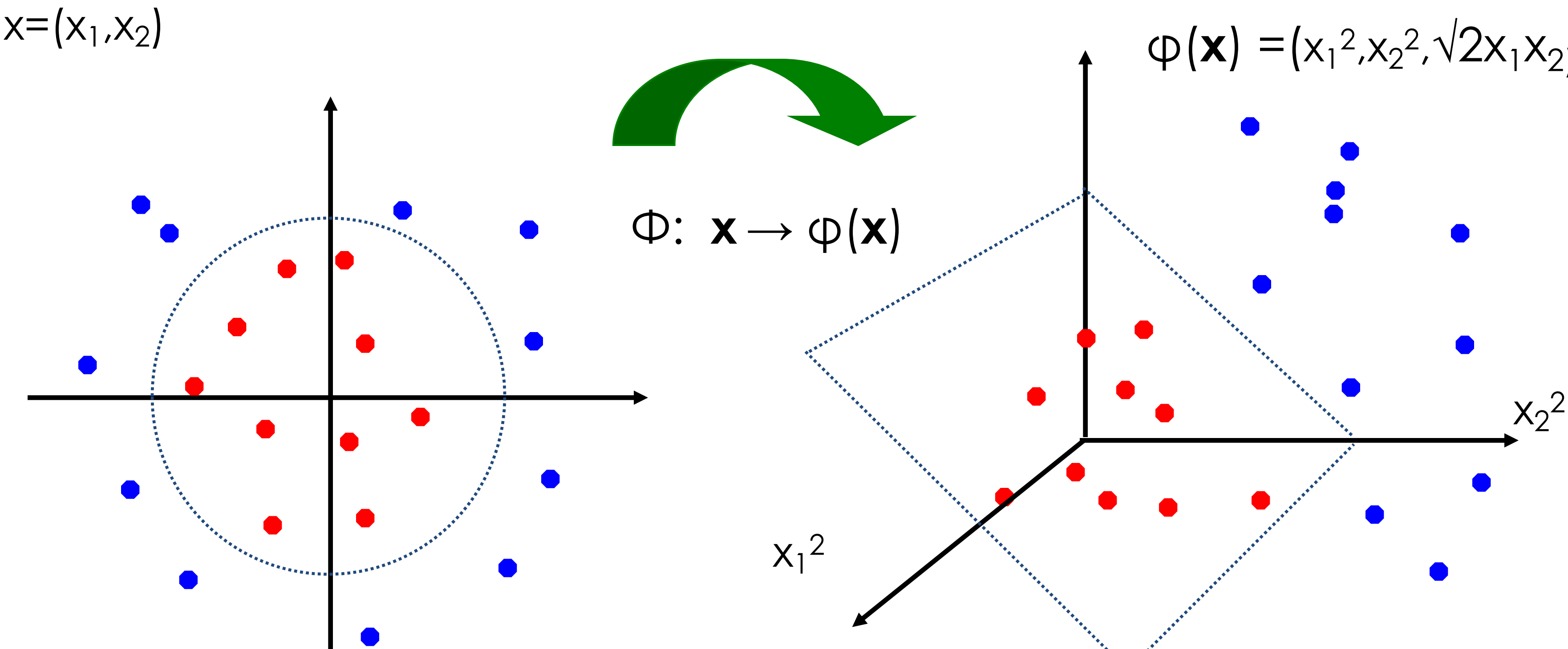
- The key is to find the value of such that noisy data does not impact the solution too much.

UNC CHARLOTTE

# Extension to Non-linear SVMs (Kernel Machines)

# Non-linear SVMs: Feature Space

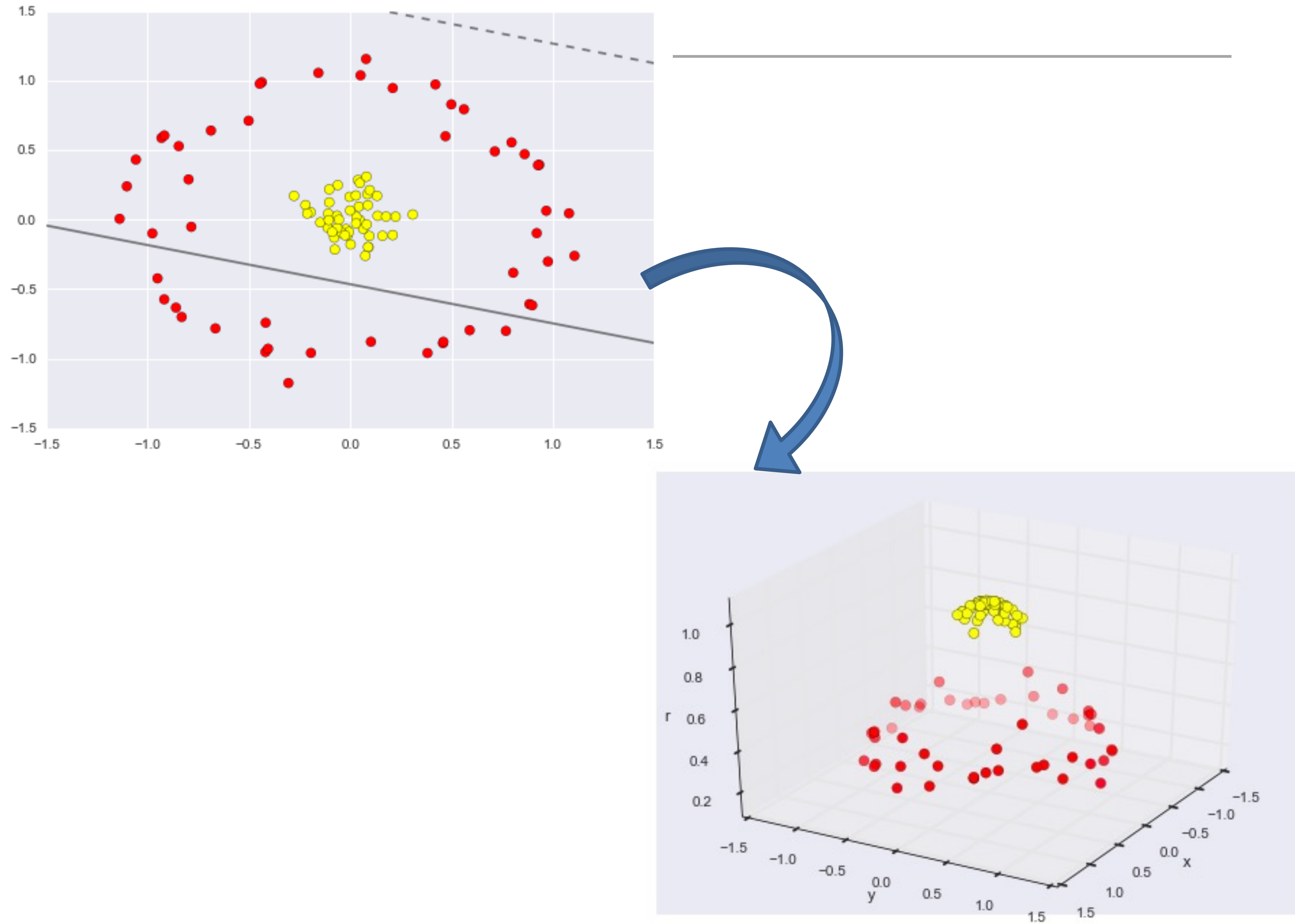
General idea: the original input space (x) can be mapped to some higher-dimensional feature space ($\varphi(\mathbf{x})$ )where the training set is separable:




If data are mapped into higher a space of sufficiently high dimension, then they will in general be linearly separable;
N data points are in general separable in a space of N-1 dimensions or more!!!

This slide is courtesy of *www.iro.umontreal.ca/~pift6080/documents/papers/**svm**_tutorial.ppt*

UNC CHARLOTTE



1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
r
1.0
0.8
0.6
0.4
0.2
y
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
x
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

# Examples of Kernel Functions

- Polynomial kernel with degree *d*

$$K(\mathbf{x},\mathbf{y}) = (\mathbf{x}^T\mathbf{y}+1)^d$$

- Radial basis function kernel with width $\sigma$

$$K(\mathbf{x},\mathbf{y}) = \exp(-||\mathbf{x}-\mathbf{y}||^2/(2\sigma^2)) = \exp\left(-\gamma\|x-y\|^2\right)$$

- Hyperbolic Tangent (Sigmoid) kernel with parameter $\kappa$ and $\theta$

$$K(\mathbf{x},\mathbf{y}) = \tanh(\kappa\mathbf{x}^T\mathbf{y}+\theta)$$

- Research on different kernel functions in different applications is very active

UNC CHARLOTTE

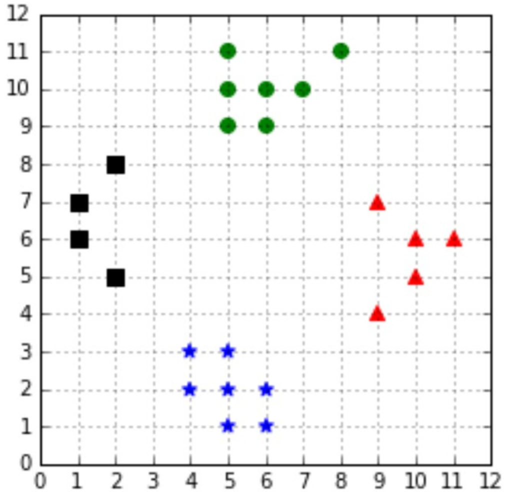
# Recap of Steps in SVM

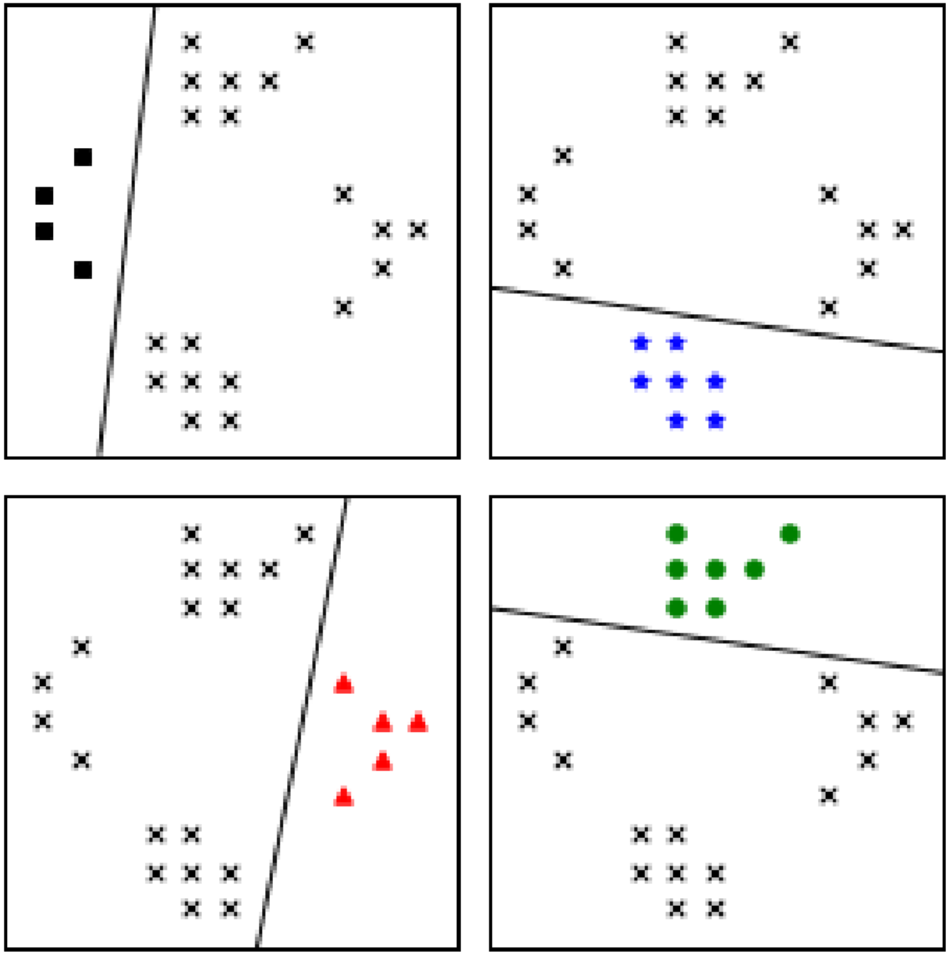
- Prepare data matrix $\{(x_i,y_i)\}$
- Select a Kernel function
- Select the error parameter *C*
- "Train" the system (to find all $\alpha_i$)
- New data can be classified using $\alpha_i$ and Support Vectors

# Extension to Non-linear SVMs (Kernel Machines)

# Multi-class SVM

**One-against-all** (one-versus-the-rest)



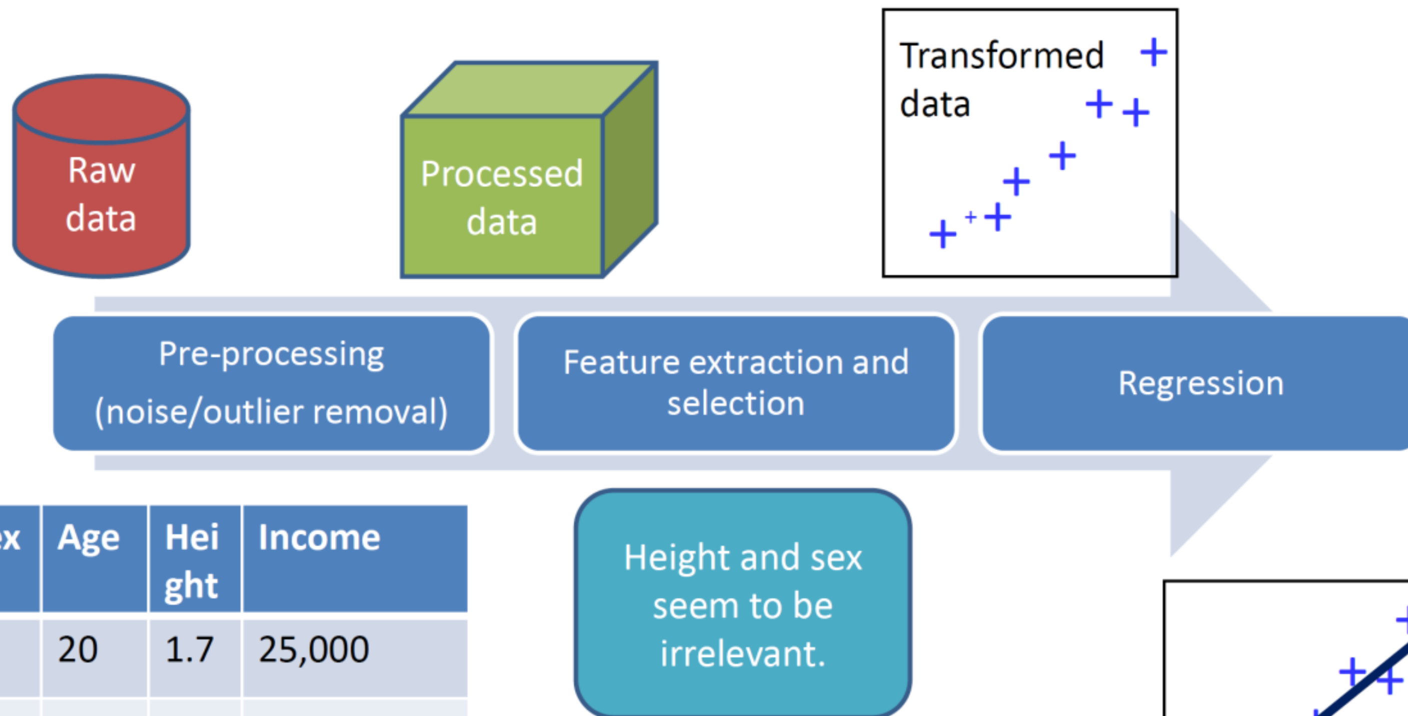



# Pros and Cons of SVM

- **Pros**
  - Good at dealing with high dimensional data
  - Works well on small data sets
- **Cons**
  - Picking the right kernel and parameters can be computationally intensive

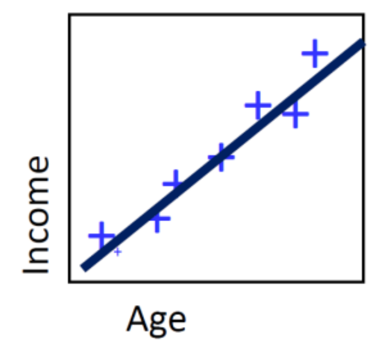
# Support Vector Regression

# Regression Processing Flow




| Sex | Age | Height | Income |
|---|---|---|---|
| M | 20 | 1.7 | 25,000 |
| F | 30 | 1.6 | 55,000 |
| M | **125** | 1.8 | 30,000 |
| … | | | |

Mis-entry (should have been 25!!!)

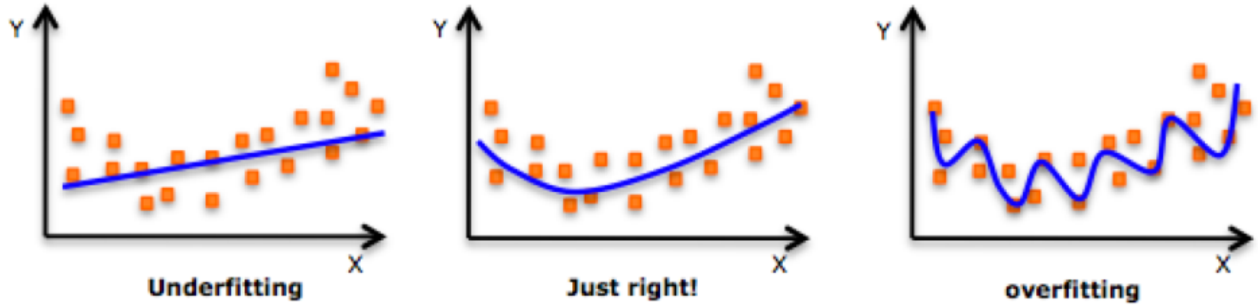





UNC CHARLOTTE

# Linear Regression

$$\min \sum_{i=1}^{m} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{m} (y_i - (\hat{\mathbf{w}} \cdot \mathbf{x}_i + \hat{b}))^2$$




- To ovoid over-fitting, a regularization term can be introduced (minimize a magnitude of w)
  - LASSO: $\min \sum_{i=1}^{m} (y_i - \mathbf{w} \cdot \mathbf{x}_i - b)^2 + C \sum_{j=1}^{n} | w_j |$
  - Ridge regression: $\min \sum_{i=1}^{m} (y_i - \mathbf{w} \cdot \mathbf{x}_i - b)^2 + C \sum_{j=1}^{n} | \mathbf{w}_j^2 |$



UNC CHARLOTTE

# Support Vector Regression

- Find a function, f(x), with at most ε-deviation from the target y

The problem can be written as a convex optimization problem

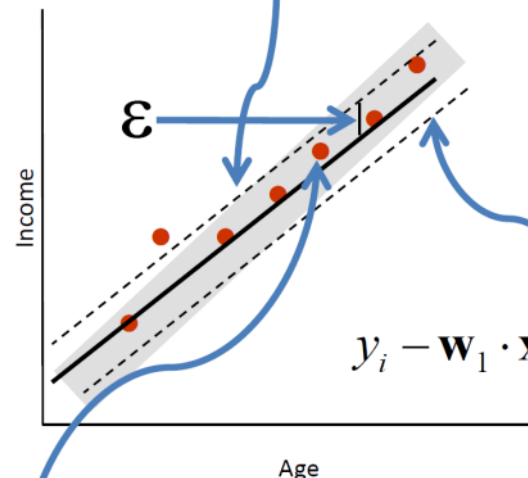
$$\min \frac{1}{2} \| \mathbf{w} \|^2$$
$$s.t.\ y_i - \mathbf{w}_1 \cdot \mathbf{x}_i - b \le \varepsilon;$$
$$\mathbf{w}_1 \cdot \mathbf{x}_i + b - y_i \le \varepsilon;$$

C: trade off the complexity

What if the problem is not feasible?

We can introduce slack variables (similar to soft margin loss function).

$\mathbf{w} \cdot \mathbf{x}_i + b - y_i \le \varepsilon$

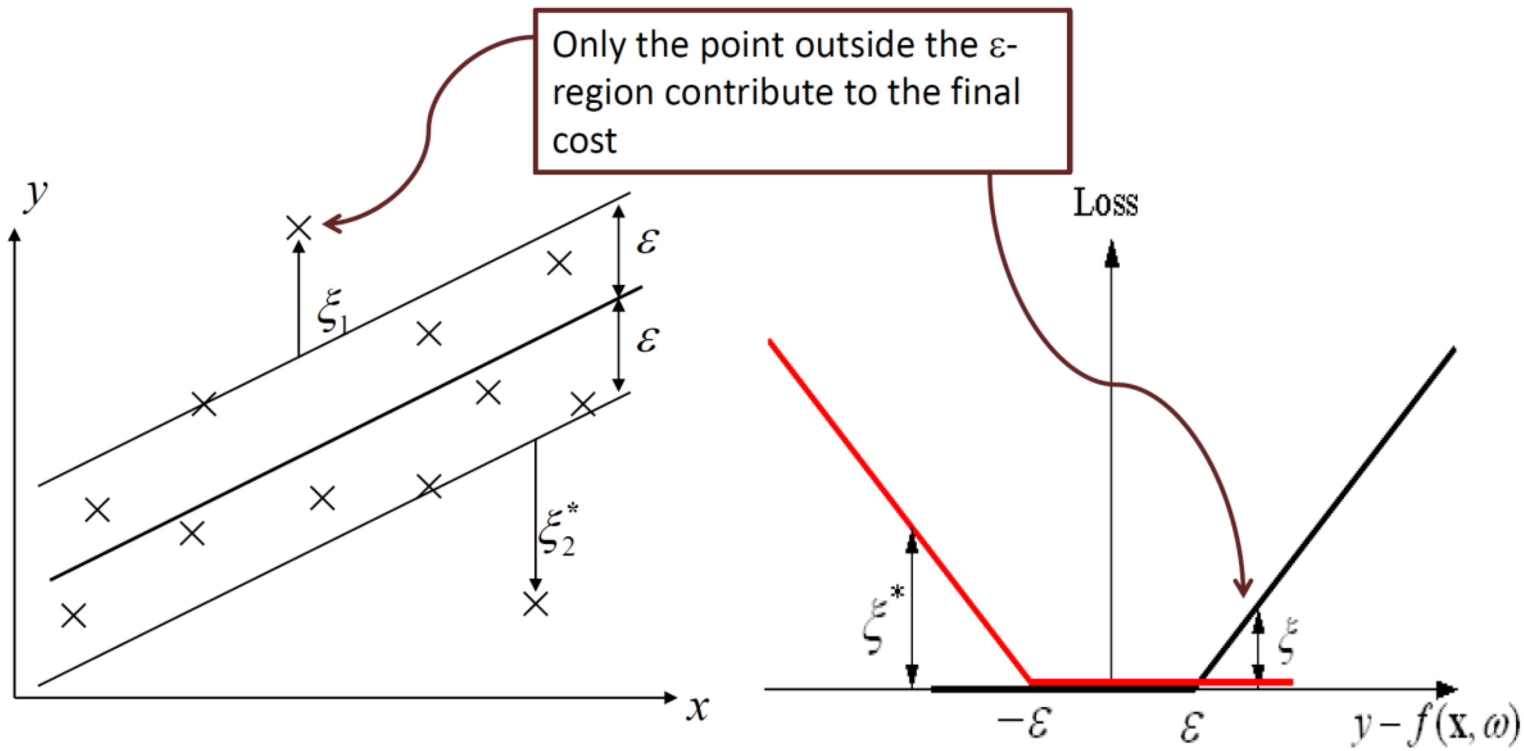



$y_i - \mathbf{w}_1 \cdot \mathbf{x}_i - b \le \varepsilon$

We do not care about errors as long as they are less than ε

# SVR Loss

Assume linear parameterization $f(\mathbf{x},\omega) = \mathbf{w} \cdot \mathbf{x} + b$






UNC CHARLOTTE

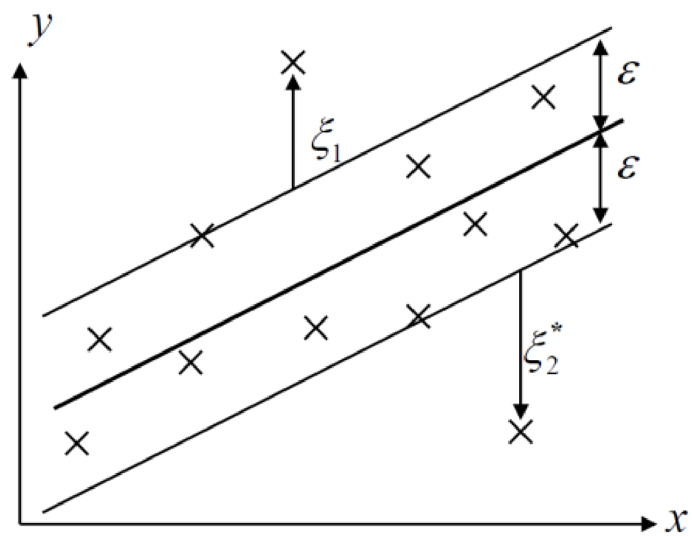
# Soft Margin SVR

Given training data

$$(\mathbf{x}_i, y_i) \qquad i = 1, \ldots, m$$

Minimize

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i^*)$$



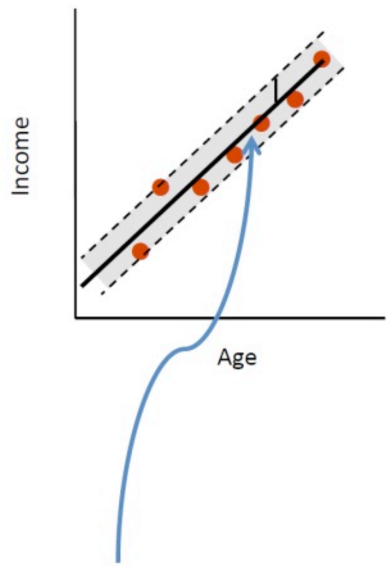



UNC CHARLOTTE

# Linear vs Non-linear SVR

- **Linear case**

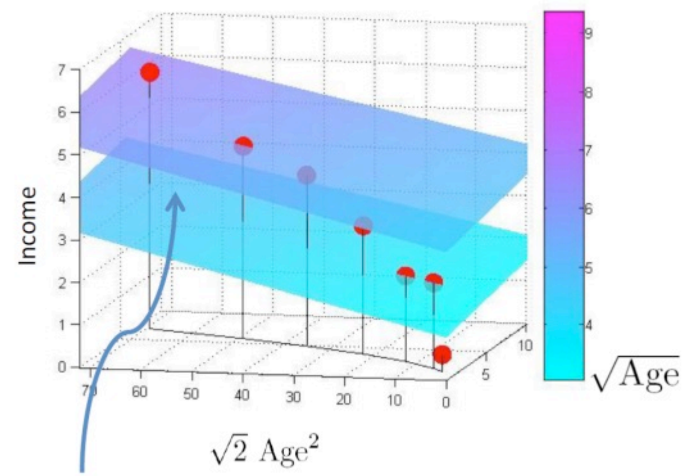
$$f: age \rightarrow income$$




$$y_i = \mathbf{w}_1 \cdot \mathbf{x}_i + b$$

- **Non-linear case**
  - Map data into a higher dimensional space, e.g.,

$$f: (\sqrt{age}, \sqrt{2}age^2) \rightarrow income$$




$$y_i = \mathbf{w}_1 \sqrt{\mathbf{x}_i} + \mathbf{w}_2 \sqrt{2}\mathbf{x}_i^2 + b$$

# Kernel Trick

- Linear: $\langle x, y \rangle$
- Non-linear: $\langle \varphi(x), \varphi(y) \rangle = K(x, y)$

Note: No need to compute the mapping function, φ(.), explicitly. Instead, we use the kernel function.

**Commonly used kernels:**

- Polynomial kernels: $K(x, y) = (x^T y + 1)^d$

- Radial basis function (RBF) kernels:

$$K(x, y) = \exp(-\frac{1}{2\sigma^2} \| x - y \|^2)$$



UNC CHARLOTTE

# SVR Applications

Stock price prediction


UNC CHARLOTTE