CSE 176 Introduction to Machine Learning
Lecture 16: Support Vector Machines

Materials from Olga Vekler and Miguel Carreira-Perpiñán

Support Vector Machine

- Start in 1979 with Vladimir Vapnik's paper
- Major developments throughout 1990's
- SVM v.s. Neural Network
  - SVM is better with limited data
  - SVM is more interpretable
  - SVM is more robust to outlier
  - SVM has elegant theory

UCMERCED

Topics today

- ❑ Intuition
- ❑ Formulation
- ❑ Optimization
- ❑ From linear to non-linear SVM (Kernel SVM)

UCMERCED

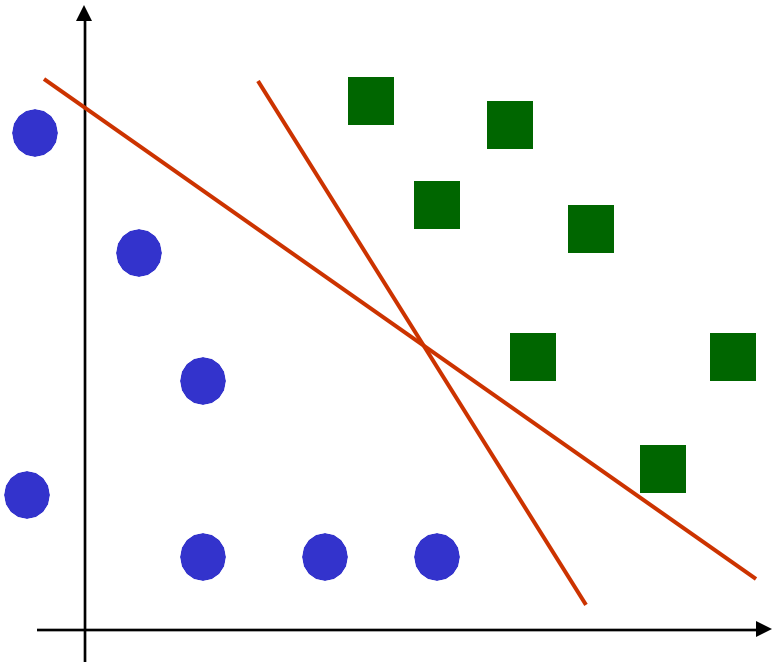
Intuition of SVM

Linear Discriminant Function

- Which separating hyperplane should we choose?

$$g(\mathbf{x}) = \mathbf{w}^t\mathbf{x} + w_0$$

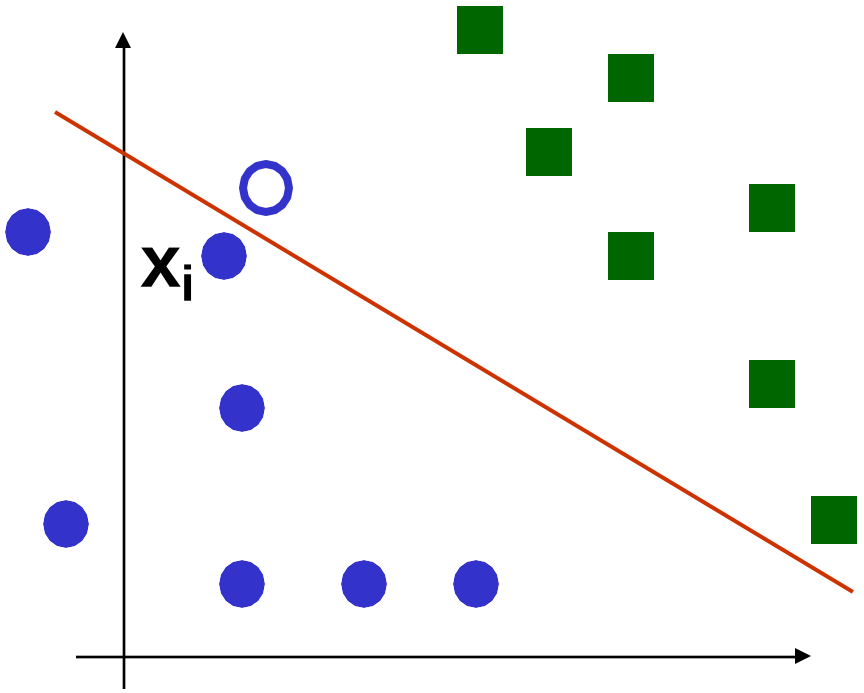
$$g(\mathbf{x}) > 0 \Rightarrow \mathbf{x} \in \text{class } 1$$
$$g(\mathbf{x}) < 0 \Rightarrow \mathbf{x} \in \text{class } 2$$



UCMERCED

Margin Intuition

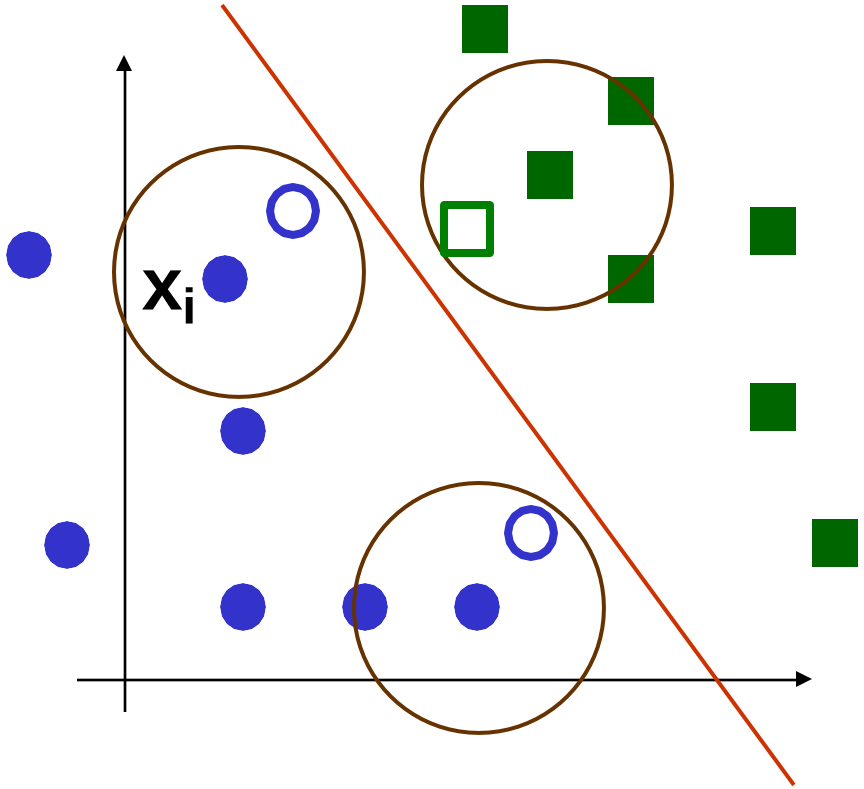
- If sample is close to sample $\mathbf{x_i}$, it is likely to be on the wrong side



- Poor generalization

Margin Intuition

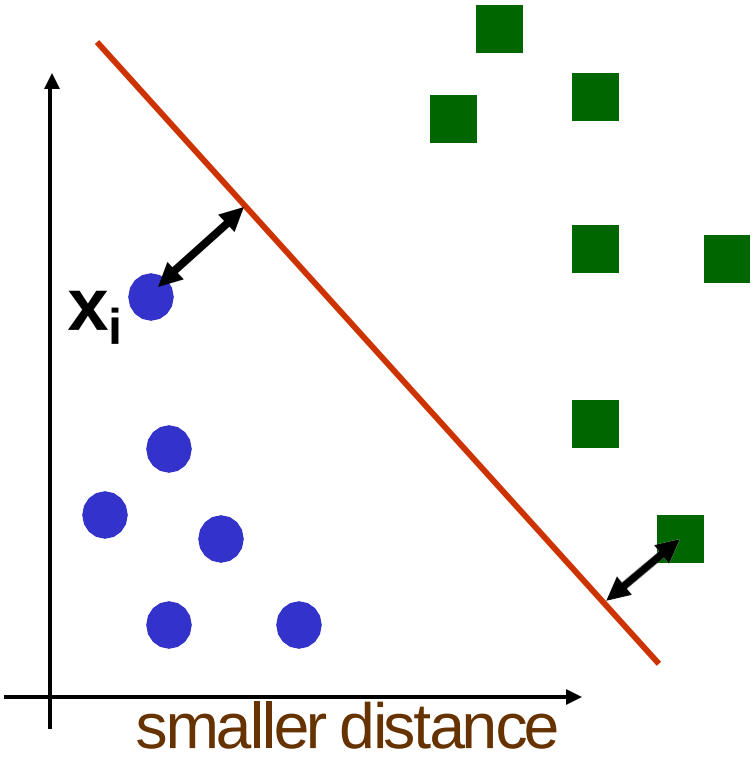
- Hyperplane as far as possible from any sample

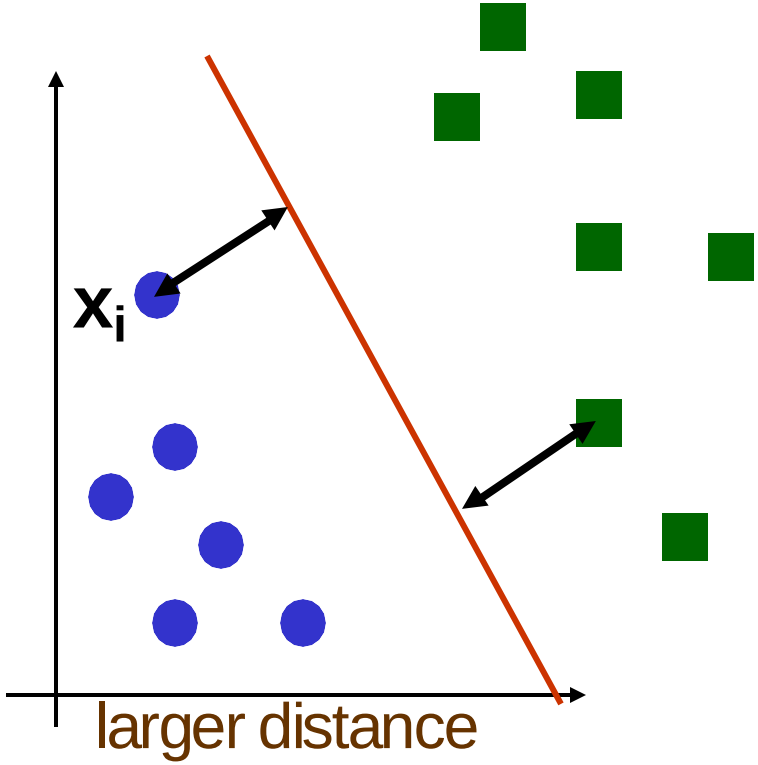


- Good generalization

SVM

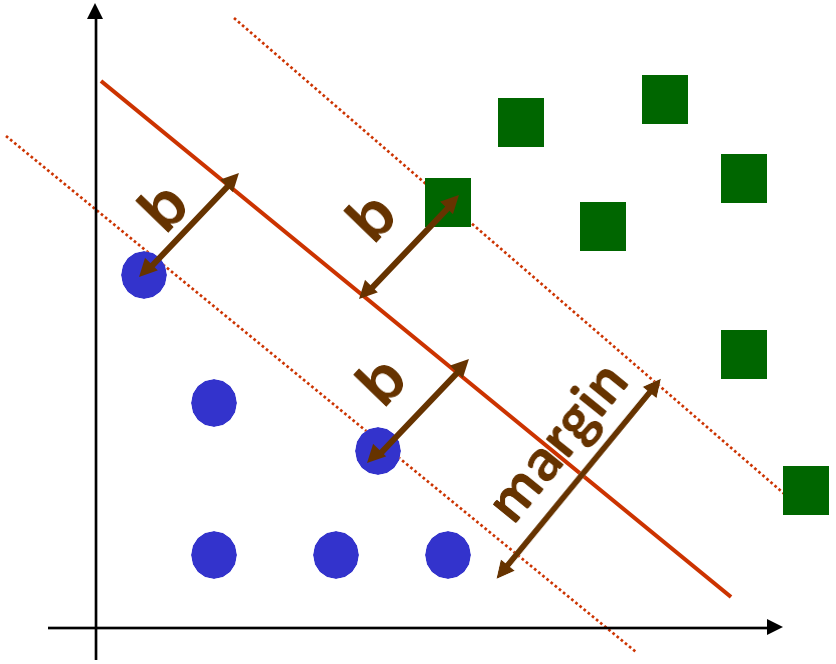
- Idea: maximize distance to the closest example





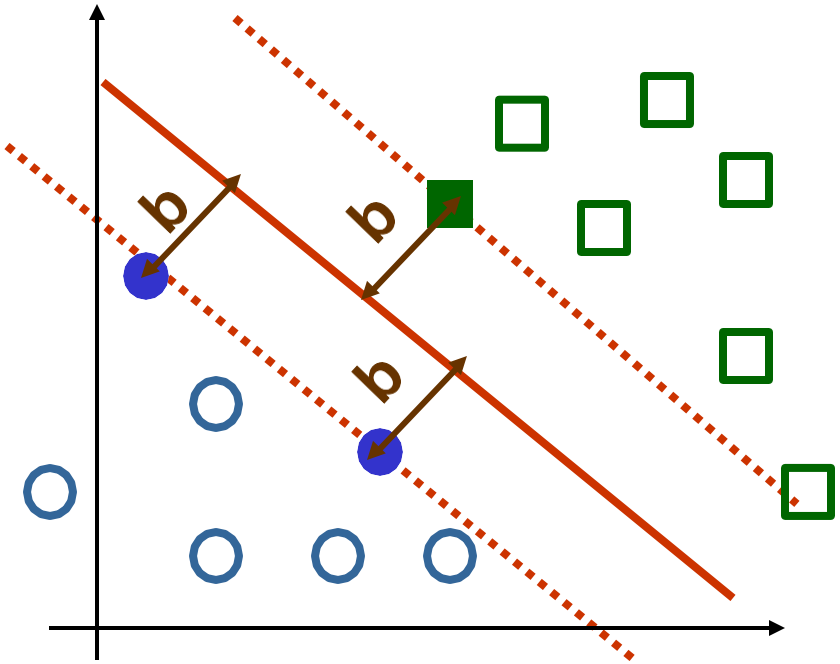
SVM: Linearly Separable Case

- SVM: maximize the ***margin***



- ***margin*** is twice the absolute value of distance ***b*** of the closest example to the separating hyperplane

SVM: Linearly Separable Case



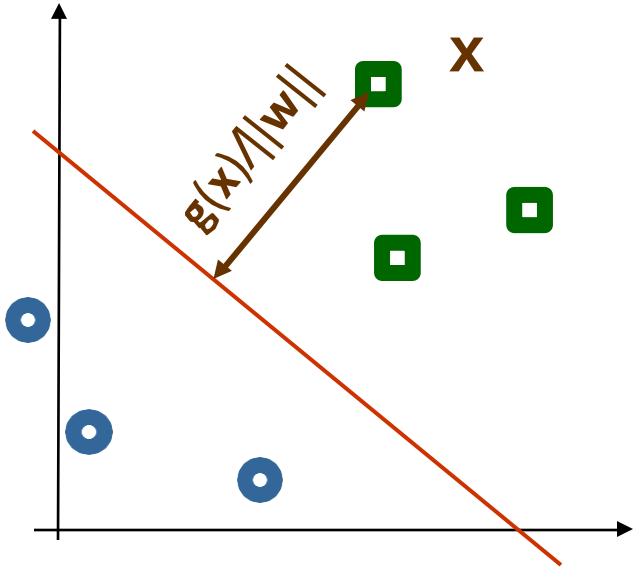
- ***Support vectors*** are samples closest to separating hyperplane

Linear SVM

SVM: Formula for the Margin

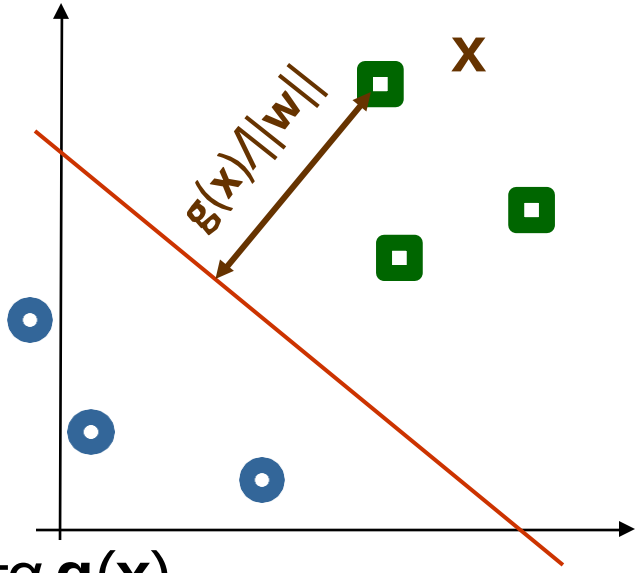
- $\mathbf{g(x) = w^t x + w_0}$
- absolute distance between **x** and the boundary **g(x) = 0**

$$\frac{|w^t x + w_0|}{\|w\|}$$



SVM: Formula for the Margin

- $\mathbf{g(x) = w^t x + w_0}$
- absolute distance between **x** and the boundary $\mathbf{g(x) = 0}$



$$\frac{|w^t x + w_0|}{\|w\|}$$

- distance is unchanged for hyperplane $\mathbf{g_1(x) = \alpha\, g(x)}$

$$\frac{|\alpha\, w^t x + \alpha\, w_0|}{\|\alpha\, w\|} = \frac{|w^t x + w_0|}{\|w\|}$$

- Let $\mathbf{x_i}$ be an example closest to the boundary. Set

$$|w^t x_i + w_0| = 1$$

- Now the largest margin hyperplane is unique

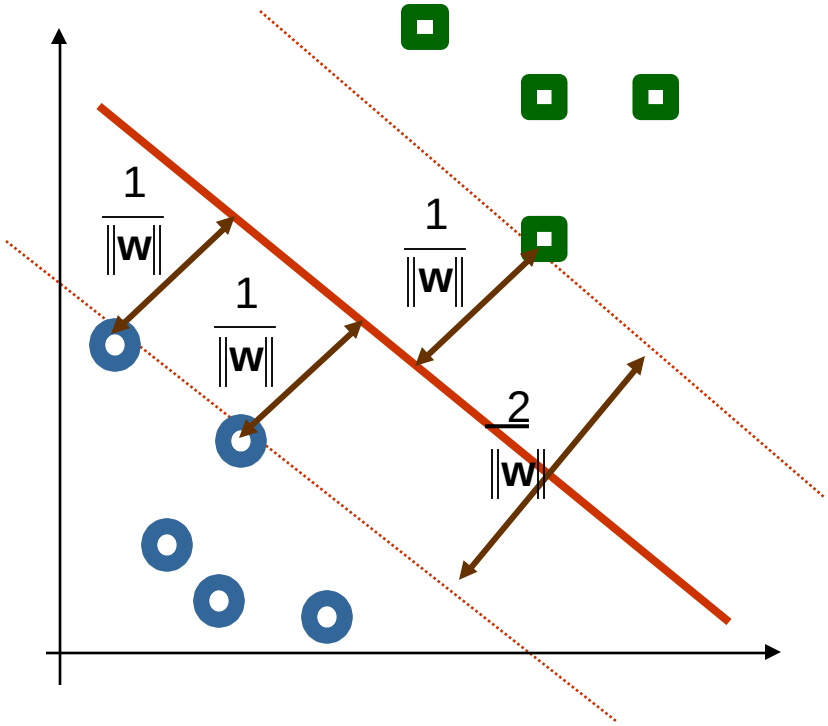
SVM: Formula for the Margin

- For uniqueness, set $|\mathbf{w}^t\mathbf{x}_i + \mathbf{w}_0| = 1$ for any example $\mathbf{x_i}$ closest to the boundary
- now distance from closest sample $\mathbf{x_i}$ to $\mathbf{g(x)} = 0$ is

$$\frac{|\mathbf{w}^t\mathbf{x}_i + \mathbf{w}_0|}{\|\mathbf{w}\|} = \frac{1}{\|\mathbf{w}\|}$$

- Thus the margin is

$$\mathbf{m} = \frac{2}{\|\mathbf{w}\|}$$



SVM: Optimal Hyperplane

- Maximize margin $\mathbf{m} = \dfrac{2}{\|\mathbf{w}\|}$
  - subject to constraints

$$\begin{cases} \mathbf{w}^t\mathbf{x}_i + \mathbf{w}_0 \geq 1 & \textbf{if } \mathbf{x}_i \textbf{ is positive example} \\ \mathbf{w}^t\mathbf{x}_i + \mathbf{w}_0 \leq -1 & \textbf{if } \mathbf{x}_i \textbf{ is negative example} \end{cases}$$

- Let

$$\begin{cases} \mathbf{z}_i = 1 & \textbf{if } \mathbf{x}_i \textbf{ is positive example} \\ \mathbf{z}_i = -1 & \textbf{if } \mathbf{x}_i \textbf{ is negative example} \end{cases}$$

- Convert our problem to

$$\begin{aligned} &\text{minimize} && J(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 \\ &\text{constrained to} && \mathbf{z}^i\left(\mathbf{w}^t\mathbf{x}_i + \mathbf{w}_0\right) \geq 1 \quad \forall \mathbf{i} \end{aligned}$$

- $J(\mathbf{w})$ is a convex function, thus it has a single global minimum

SVM: Optimal Hyperplane

- Use Kuhn-Tucker theorem to convert our problem to:

maximize	$\mathbf{L_D}(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \mathbf{z_i z_j x_i^t x_j}$
constrained to	$\alpha_i \geq 0 \quad \forall \mathbf{i}$ **and** $\sum_{i=1}^{n} \alpha_i \mathbf{z_i} = 0$

- $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_n\}$ are new variables, one for each sample
- $\mathbf{L_D}(\boldsymbol{\alpha})$ can be optimized by quadratic programming
- $\mathbf{L_D}(\boldsymbol{\alpha})$ formulated in terms of $\boldsymbol{\alpha}$
  - depends on $\mathbf{w}$ and $\mathbf{w_0}$

SVM: Optimal Hyperplane

- After finding the optimal $\alpha = \{\alpha_1, \ldots, \alpha_n\}$
  - for every sample **i**, one of the following must hold
    - $\alpha_i = 0$ (sample ***i*** is not a support vector)
    - $\alpha_i \neq 0$ and $z_i(\mathbf{w}^t\mathbf{x}_i + w_0 - 1) = 0$ (sample **i** is support vector)
  - compute $\mathbf{w} = \sum_{i=1}^{n} \alpha_i z_i \mathbf{x}_i$
  - solve for w_0 using any $\alpha_i > 0$ and $\alpha_i \left[z_i \left(\mathbf{w}^t \mathbf{x}_i + w_0 \right) - 1 \right] = 0$

$$w_0 = \frac{1}{z_i} - \mathbf{w}^t \mathbf{x}_i$$

- Final discriminant function:

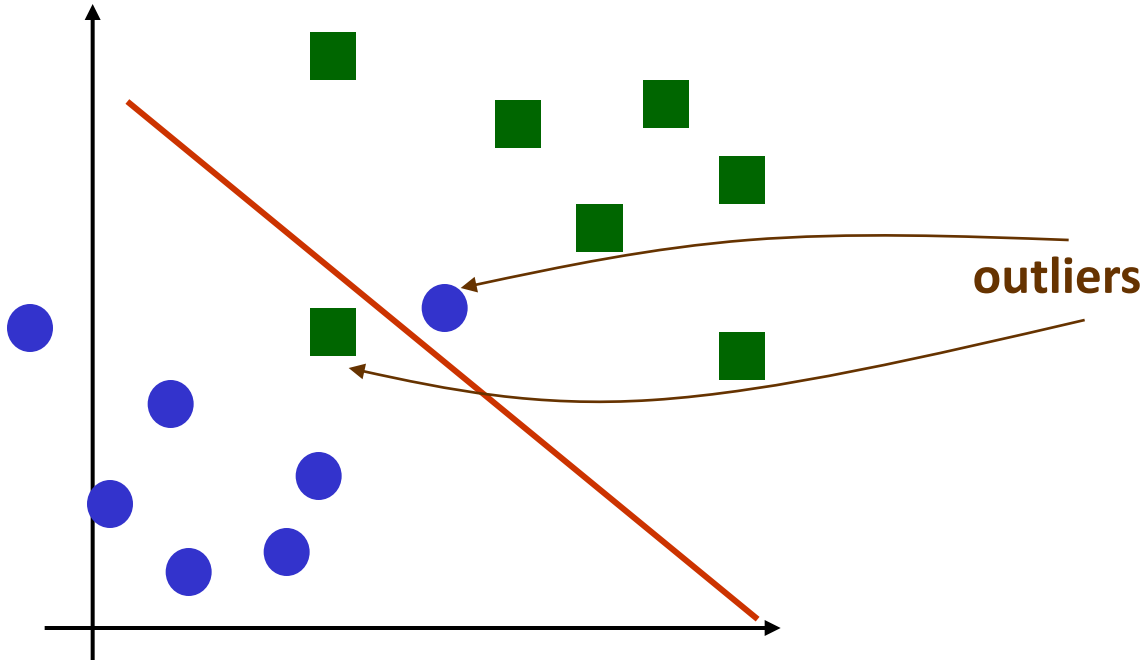
$$g(\mathbf{x}) = \left(\sum_{\mathbf{x}_i \in S} \alpha_i z_i \mathbf{x}_i \right)^t \mathbf{x} + w_0$$

  - where ***S*** is the set of support vectors

$$S = \{\mathbf{x}_i \mid \alpha_i \neq 0\}$$

SVM: Non Separable Case

- Linear classifier still be appropriate when data is not linearly separable, but almost linearly separable



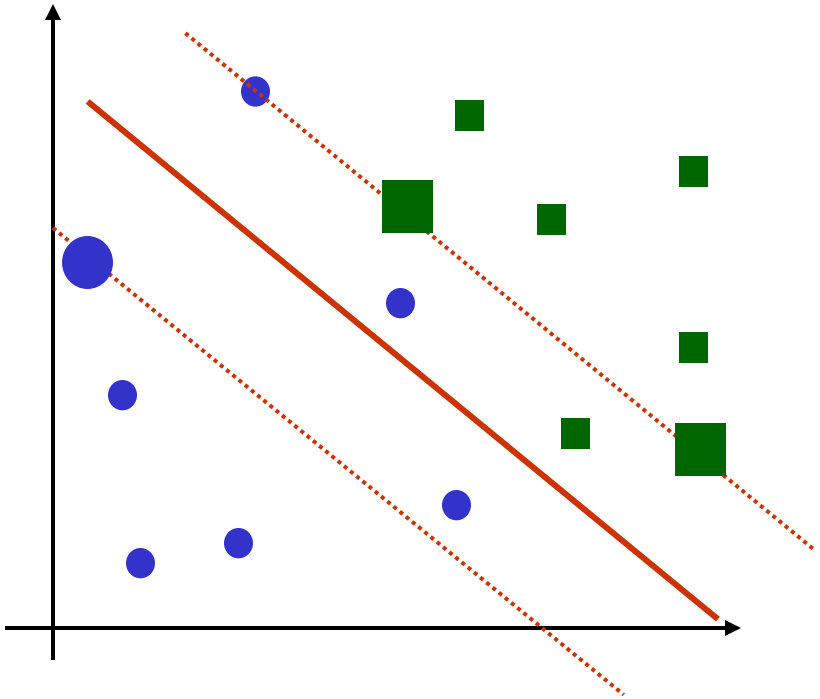
- Can adapt SVM to almost linearly separable case

SVM as Unconstrained Minimization

- SVM objective can be rewritten as unconstrained optimization

$$J(\mathbf{w}) = \underbrace{\frac{1}{2}\|\mathbf{w}\|^2}_{\text{weights regularization}} + \beta \underbrace{\sum_{i=1}^{n} \max(0, 1 - z_i f(\mathbf{x}_i))}_{\text{loss function}}$$

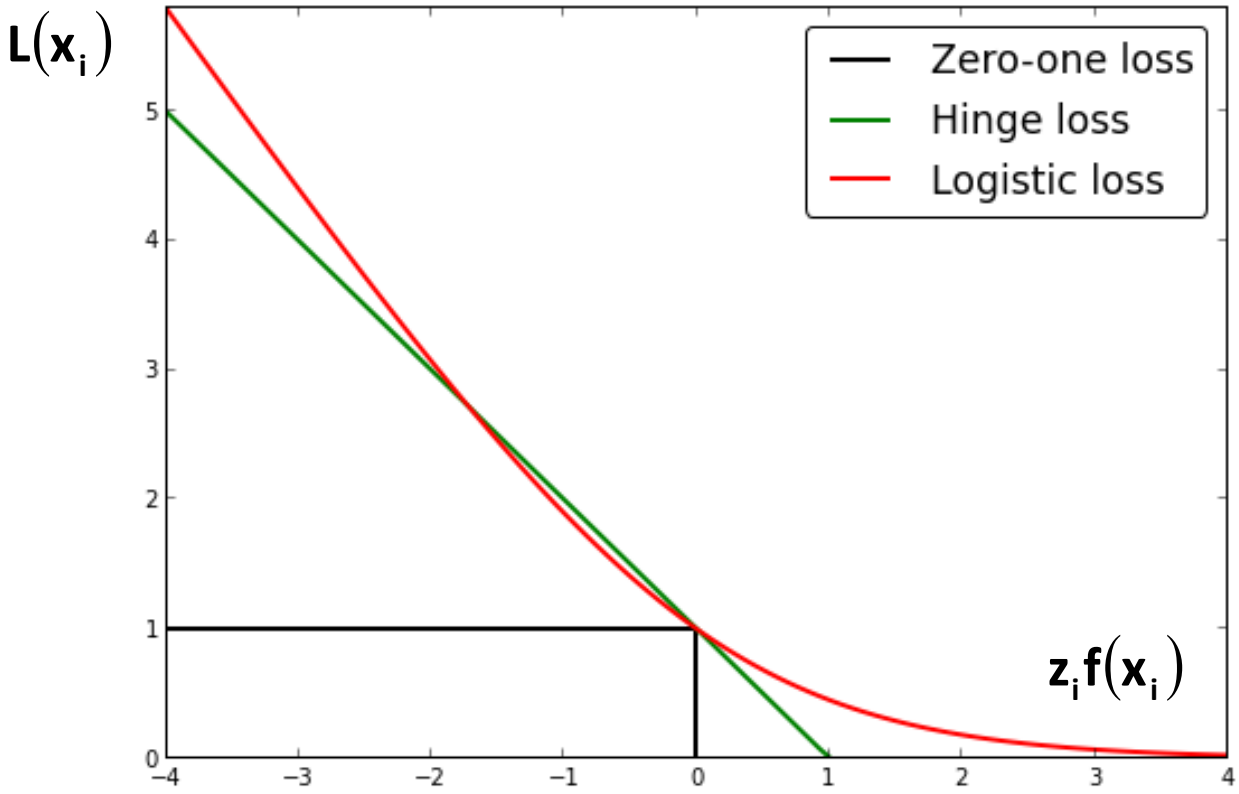
- $z_i f(\mathbf{x}_i) > 1$: $\mathbf{x}_i$ is on the right side of the hyperplane and outside margin, no loss
- $z_i f(\mathbf{x}_i) = 1$: $\mathbf{x}_i$ on the margin, no loss
- $z_i f(\mathbf{x}_i) < 1$: $\mathbf{x}_i$ is inside margin, or on the wrong side of the hyperplane, contributes to loss



SVM: Hinge Loss

- SVM uses Hinge loss per sample $\mathbf{x}_i$

$$\mathbf{L}_i(\mathbf{x}_i) = \max(0, 1 - \mathbf{z}_i \mathbf{f}(\mathbf{x}_i))$$



- Hinge loss encourages classification with a margin of 1

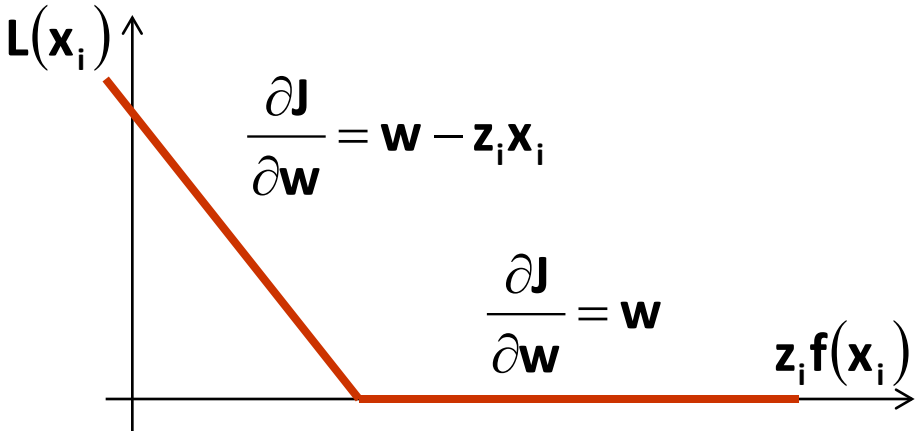
SVM: Hinge Loss

- Can optimize with gradient descent, convex function

$$J(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + \beta \sum_{i=1}^{n} \max(0, 1 - z_i f(\mathbf{x}_i))$$

$$f(\mathbf{x}_i) = \mathbf{w}^t \mathbf{x}_i + w_0$$

- Gradient



- Gradient descent, single sample

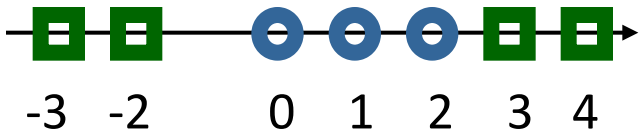
$$\mathbf{w} = \begin{cases} \mathbf{w} - \alpha(\mathbf{w} - \beta z_i \mathbf{x}_i) & \text{if } z_i f(\mathbf{x}_i) < 1 \\ \mathbf{w} - \alpha \mathbf{w} & \text{otherwise} \end{cases}$$

Non-linear SVM (Kernel SVM)

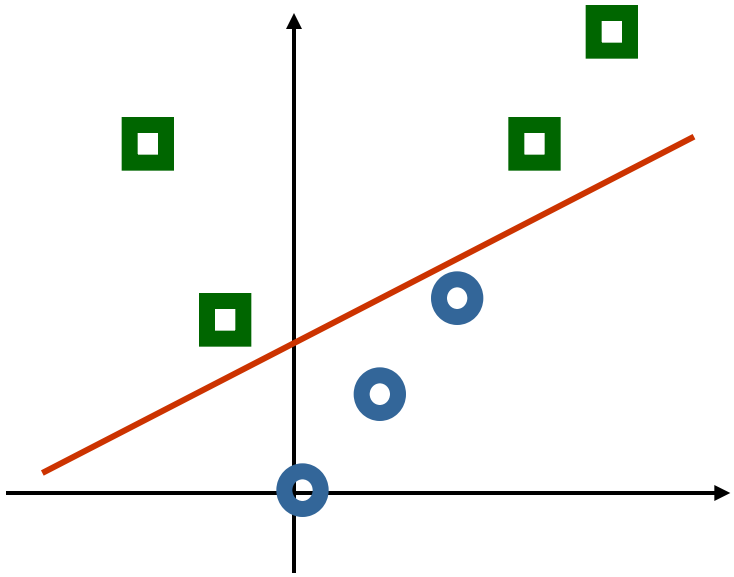
Non Linear Mapping

- Cover's theorem:
  - *"pattern-classification problem cast in a high dimensional space non-linearly is more likely to be linearly separable than in a low-dimensional space"*

- Not linearly separable in 1D

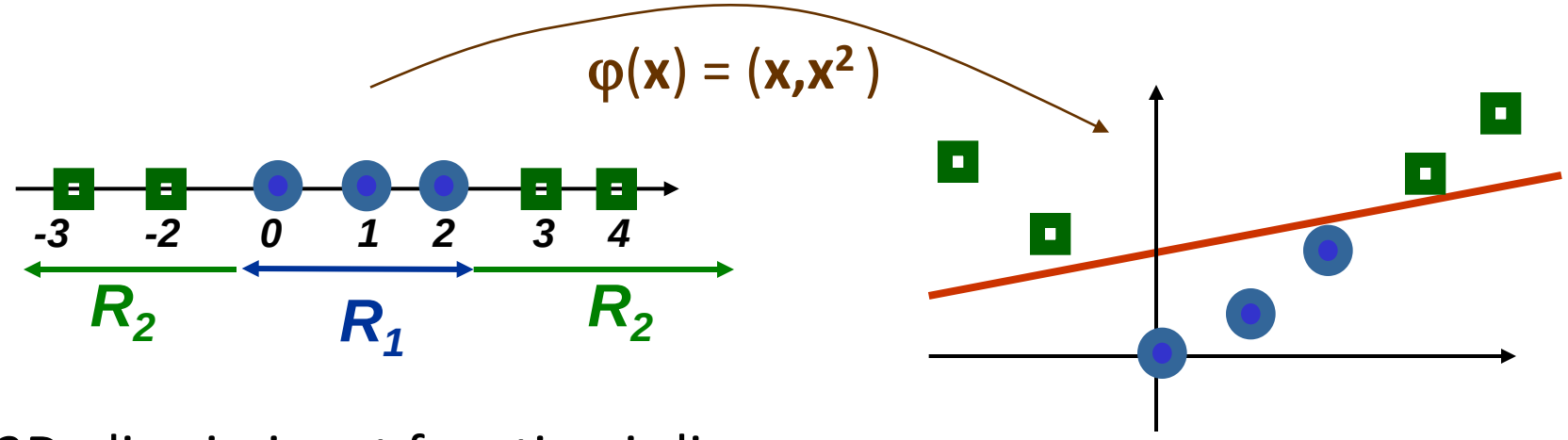


- Lift to 2D space with $\mathbf{h}(\mathbf{x}) = (\mathbf{x}, \mathbf{x}^2)$



Non Linear Mapping

- To solve a non linear problem with a linear classifier
  1. Project data ***x*** to high dimension using function $\varphi(\mathbf{x})$
  2. Find a linear discriminant function for transformed data $\varphi(\mathbf{x})$
  3. Final nonlinear discriminant function is $\mathbf{g(x)} = \mathbf{w}^t\,\varphi(\mathbf{x}) + \mathbf{w}_0$

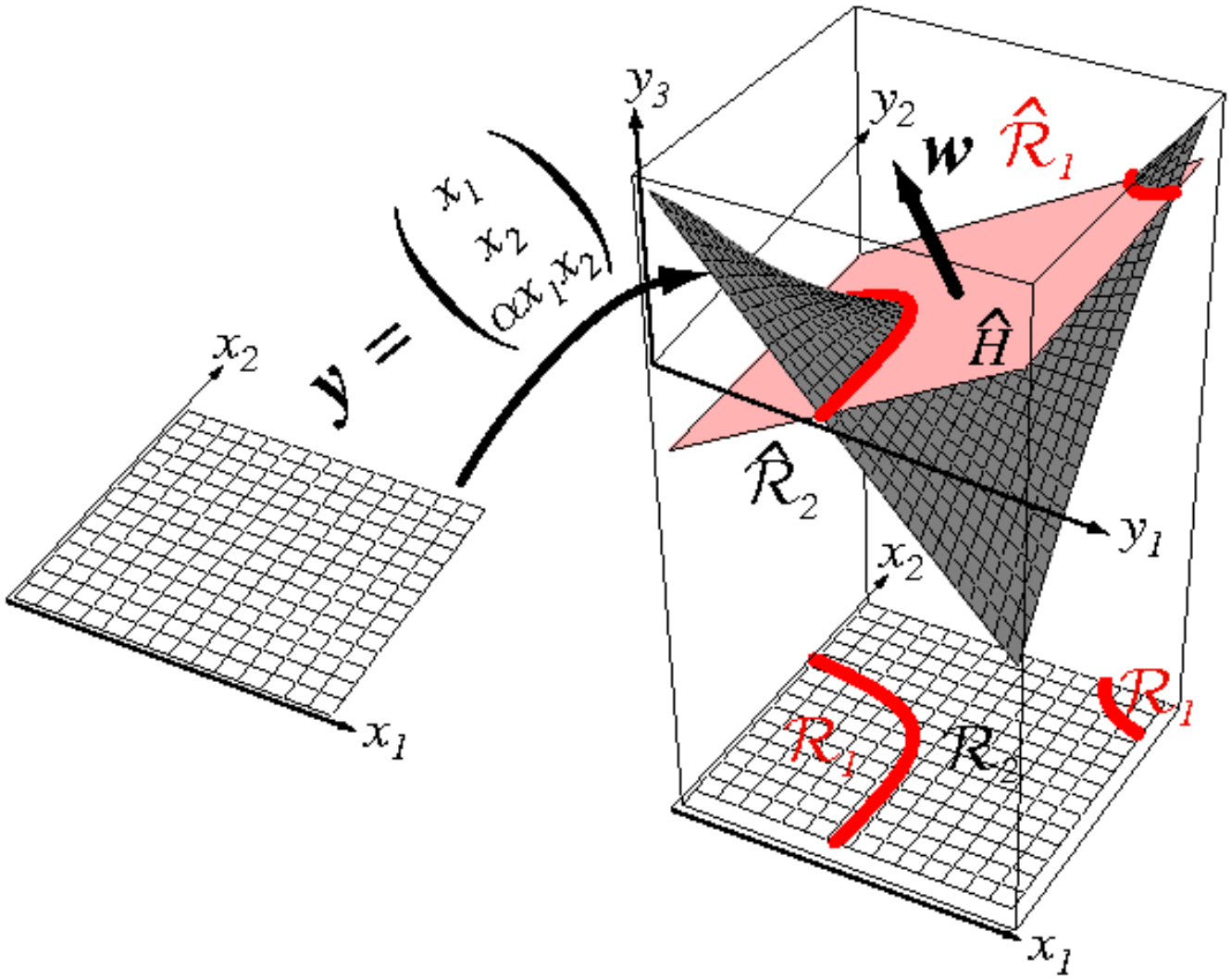


- In 2D, discriminant function is linear

$$\mathbf{g}\left(\begin{bmatrix}\mathbf{x}^{(1)}\\ \mathbf{x}^{(2)}\end{bmatrix}\right) = \begin{bmatrix}\mathbf{w}_1 & \mathbf{w}_2\end{bmatrix}\begin{bmatrix}\mathbf{x}^{(1)}\\ \mathbf{x}^{(2)}\end{bmatrix} + \mathbf{w}_0$$

- In 1D, discriminant function is not linear $\mathbf{g}(\mathbf{x}) = \mathbf{w}_1\mathbf{x} + \mathbf{w}_2\mathbf{x}^2 + \mathbf{w}_0$

Non Linear Mapping: Another Example



Non Linear SVM

- Can use any linear classifier after lifting data into a higher dimensional space
- However we will have to deal with the “curse of dimensionality”
  1. poor generalization to test data
  2. computationally expensive

- SVM avoids the “curse of dimensionality” by
  - enforcing largest margin permits good generalization
  - computation in the higher dimensional case is performed only implicitly through the use of ***kernel*** functions

Non Linear SVM: Kernels

- Recall SVM optimization

maximize $$L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_i z_i z_j x_i^t x_j$$

- Optimization depends on samples x_i only through the dot product $x_i^t x_j$

- If we lift x_i to high dimension using $\varphi(x)$, need to compute high dimensional product $\varphi(x_i)^t \varphi(x_j)$

maximize $$L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_i z_i z_j \underbrace{\varphi(x_i)^t \varphi(x_j)}_{K(x_i, x_j)}$$

- Idea: find ***kernel*** function $K(x_i, x_j)$ s.t. $K(x_i, x_j) = \varphi(x_i)^t \varphi(x_j)$

Non Linear SVM: Kernels

maximize $$L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_i z_i z_j \varphi(x_i)^t \varphi(x_j)$$

$K(x_i, x_j)$

- Kernel trick
  - only need to compute $K(x_i, x_j)$ instead of $\varphi(x_i)^t \varphi(x_j)$
  - no need to lift data in high dimension explicitely, computation is performed in the original dimension

Non Linear SVM: Kernels

- Suppose we have 2 features and $K(\mathbf{x},\mathbf{y}) = (\mathbf{x}^t\mathbf{y})^2$

- Which mapping $\varphi(\mathbf{x})$ does it correspond to?

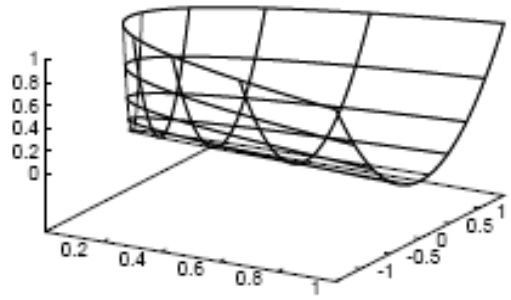
$$K(\mathbf{x},\mathbf{y}) = (\mathbf{x}^t\mathbf{y})^2 = \left(\begin{bmatrix} x^{(1)} & x^{(2)} \end{bmatrix} \begin{bmatrix} y^{(1)} \\ y^{(2)} \end{bmatrix} \right)^2 = \left(x^{(1)}y^{(1)} + x^{(2)}y^{(2)}\right)^2$$

$$= \left(x^{(1)}y^{(1)}\right)^2 + 2\left(x^{(1)}y^{(1)}\right)\left(x^{(2)}y^{(2)}\right) + \left(x^{(2)}y^{(2)}\right)^2$$

$$= \begin{bmatrix} \left(x^{(1)}\right)^2 & \sqrt{2}x^{(1)}x^{(1)}x^{(2)} & \left(x^{(2)}\right)^2 \end{bmatrix} \begin{bmatrix} \left(y^{(1)}\right)^2 & \sqrt{2}y^{(1)}y^{(1)}y^{(2)} & \left(y^{(2)}\right)^2 \end{bmatrix}^t$$

- Thus

$$\varphi(\mathbf{x}) = \begin{bmatrix} \left(x^{(1)}\right) & \sqrt{2}x^{(1)}x^{(2)} & \left(x^{(2)}\right) \end{bmatrix}$$



Non Linear SVM: Kernels

- How to choose kernel $\mathbf{K}(\mathbf{x}_i,\mathbf{x}_j)$?
  - $\mathbf{K}(\mathbf{x}_i,\mathbf{x}_j)$ should correspond to product $\boldsymbol{\varphi}(\boldsymbol{x}_i)^t\boldsymbol{\varphi}(\boldsymbol{x}_j)$ in a higher dimensional space
  - Mercer's condition states which kernel function can be expressed as dot product of two vectors
  - Kernel's not satisfying Mercer's condition can be sometimes used, but no geometrical interpretation
- Common choices satisfying Mercer's condition
  - Polynomial kernel $\mathbf{K}(\mathbf{x}_i,\mathbf{x}_j)=(\mathbf{x}_i^t\mathbf{x}_j+1)^p$
  - Gaussian radial Basis kernel (data is lifted in infinite dimensions)

$$\mathbf{K}(\mathbf{x}_i,\mathbf{x}_j)=\exp\left(-\frac{1}{2\sigma^2}\|\mathbf{x}_i-\mathbf{x}_j\|^2\right)$$

Non Linear SVM

- search for separating hyperplane in high dimension

$$\mathbf{w}\varphi(\mathbf{x}) + \mathbf{w}_0 = 0$$

- Choose $\varphi(\boldsymbol{x})$ so that the first ("0"th) dimension is the augmented dimension with feature value fixed to 1

$$\varphi(\mathbf{x}) = \begin{bmatrix} 1 & \mathbf{x}^{(1)} & \mathbf{x}^{(2)} & \mathbf{x}^{(1)}\mathbf{x}^{(2)} \end{bmatrix}^t$$

- Threshold $\mathbf{w_0}$ gets folded into vector $\mathbf{w}$

$$\begin{bmatrix} \mathbf{w}_0 & \mathbf{w} \end{bmatrix} \underbrace{\begin{bmatrix} 1 \\ * \end{bmatrix}}_{\varphi(\mathbf{x})} = 0$$

Non Linear SVM

- Thus seeking hyperplane

$$\mathbf{w}\varphi(\mathbf{x}) = 0$$

- Or, equivalently, a hyperplane that goes through the origin in high dimensions
  - removes only one degree of freedom
  - but we introduced many new degrees when lifted the data in high dimension

Non Linear SVM Recepie

- Start with $\mathbf{x}_1,\ldots,\mathbf{x}_n$ in original feature space of dimension **d**
- Choose kernel $\mathbf{K}(\mathbf{x}_i,\mathbf{x}_j)$
  - implicitly chooses function $\boldsymbol{\varphi}(\mathbf{x}_i)$ that takes $\mathbf{x}_i$ to a higher dimensional space
  - gives dot product in the high dimensional space
- Find largest margin linear classifier in the higher dimensional space by using quadratic programming package to solve

maximize $$\mathbf{L}_D(\alpha)=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_i\mathbf{z}_i\mathbf{z}_j\mathbf{K}(\mathbf{x}_i,\mathbf{x}_j)$$

constrained to $$0\le\alpha_i\le\beta \quad \forall\mathbf{i} \quad \textbf{and} \quad \sum_{i=1}^{n}\alpha_i\mathbf{z}_i=0$$

Non Linear SVM Recipe

- Weight vector ***w*** in the high dimensional space

$$\mathbf{w} = \sum_{\mathbf{x}_i \in \mathbf{S}} \alpha_i \mathbf{z}_i \varphi(\mathbf{x}_i)$$

  - where ***S*** is the set of support vectors

$$\mathbf{S} = \{\mathbf{x}_i \mid \alpha_i \neq 0\}$$

- Linear discriminant function in the high dimensional space

$$\mathbf{g}(\varphi(\mathbf{x})) = \mathbf{w}^t \varphi(\mathbf{x}) = \left(\sum_{\mathbf{x}_i \in \mathbf{S}} \alpha_i \mathbf{z}_i \varphi(\mathbf{x}_i) \right)^t \varphi(\mathbf{x})$$

- Non linear discriminant function in the original space:

$$\mathbf{g}(\mathbf{x}) = \left(\sum_{\mathbf{x}_i \in \mathbf{S}} \alpha_i \mathbf{z}_i \varphi(\mathbf{x}_i) \right)^t \varphi(\mathbf{x}) = \sum_{\mathbf{x}_i \in \mathbf{S}} \alpha_i \mathbf{z}_i \varphi^t(\mathbf{x}_i) \varphi(\mathbf{x}) = \sum_{\mathbf{x}_i \in \mathbf{S}} \alpha_i \mathbf{z}_i \mathbf{K}(\mathbf{x}_i, \mathbf{x})$$

- Decide class 1 if ***g***(***x***) > 0, otherwise decide class 2

Non Linear SVM

- Nonlinear discriminant function

$$g(\mathbf{x}) = \sum_{\mathbf{x}_i \in S} \alpha_i \; z_i \; K(\mathbf{x}_i, \mathbf{x})$$

$$g(\mathbf{x}) = \sum$$

weight of support vector $\mathbf{x}_i$

∓ 1

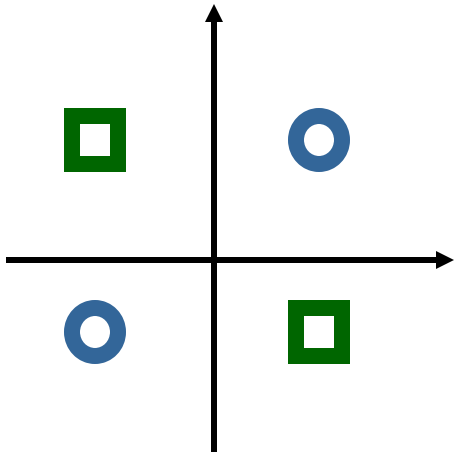
similarity between $\boldsymbol{x}$ and support vector $\mathbf{x}_i$

most important training samples, i.e. support vectors

$$K(\mathbf{x}_i, \mathbf{x}) = \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{x}_i - \mathbf{x}\|^2 \right)$$

SVM Example: XOR Problem

- Class 1: $\mathbf{x_1} = [1,-1]$, $\mathbf{x_2} = [-1,1]$
- Class 2: $\mathbf{x_3} = [1,1]$, $\mathbf{x_4} = [-1,-1]$
- Use polynomial kernel of degree 2
  - $\mathbf{K}(\mathbf{x_i},\mathbf{x_j}) = (\mathbf{x_i}^t \mathbf{x_j} + 1)^2$
  - Kernel corresponds to mapping



$$(\mathbf{x}) = \left[1 \quad \sqrt{2}\mathbf{x}^{(1)} \quad \sqrt{2}\mathbf{x}^{(2)} \quad \sqrt{2}\mathbf{x}^{(1)}\mathbf{x}^{(2)} \quad \left(\mathbf{x}^{(1)}\right)^2 \quad \left(\mathbf{x}^{(2)}\right)^2\right]^t$$

- Need to maximize $\mathbf{L_D}(\alpha) = \sum_{i=1}^{4} \alpha_i \quad \frac{1}{2}\sum_{i=1}^{4}\sum_{j=1}^{4} \alpha_i \alpha_i \mathbf{z_i z_j} \left(\mathbf{x_i^t x_j} + 1\right)^2$

constrained to $0 \le \alpha_i \quad \forall \mathbf{i}$ **and** $\alpha_1 + \alpha_2 - \alpha_3 - \alpha_4 = 0$

SVM Example: XOR Problem

- Rewrite $\mathbf{L_D}(\alpha) = \sum_{i=1}^{4} \alpha_i - \frac{1}{2}\alpha^t \mathbf{H} \alpha$
  - where $\alpha = [\alpha_1 \quad \alpha_2 \quad \alpha_3 \quad \alpha_4]^t$ and $\mathbf{H} = \begin{bmatrix} 9 & 1 & -1 & -1 \\ 1 & 9 & -1 & -1 \\ -1 & -1 & 9 & 1 \\ -1 & -1 & 1 & 9 \end{bmatrix}$
- Take derivative with respect to α and set it to ***0***

$$\frac{\mathbf{d}}{\mathbf{da}}\mathbf{L_D}(\alpha) = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 9 & 1 & -1 & -1 \\ 1 & 9 & -1 & -1 \\ -1 & -1 & 9 & 1 \\ -1 & -1 & 1 & 9 \end{bmatrix} \alpha = 0$$

- Solution to the above is $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \mathbf{0.25}$
  - satisfies the constraints $\forall \mathbf{i},\ 0 \le \alpha_i$ **and** $\alpha_1 + \alpha_2 - \alpha_3 - \alpha_4 = 0$
  - all samples are support vectors

SVM Example: XOR Problem

$$\varphi(\mathbf{x}) = \begin{bmatrix} 1 & \sqrt{2}\mathbf{x}^{(1)} & \sqrt{2}\mathbf{x}^{(2)} & \sqrt{2}\mathbf{x}^{(1)}\mathbf{x}^{(2)} & \left(\mathbf{x}^{(1)}\right)^2 & \left(\mathbf{x}^{(2)}\right)^2 \end{bmatrix}^t$$

- Weight vector **w** is:

$$\mathbf{w} = \sum_{i=1}^{4} \alpha_i \mathbf{z}_i \varphi(\mathbf{x}_i) = 0.25\left(\varphi(\mathbf{x}_1) + \varphi(\mathbf{x}_2) - \varphi(\mathbf{x}_3) - \varphi(\mathbf{x}_4)\right)$$

$$= \begin{bmatrix} 0 & 0 & 0 & \sqrt{2} & 0 & 0 \end{bmatrix}$$

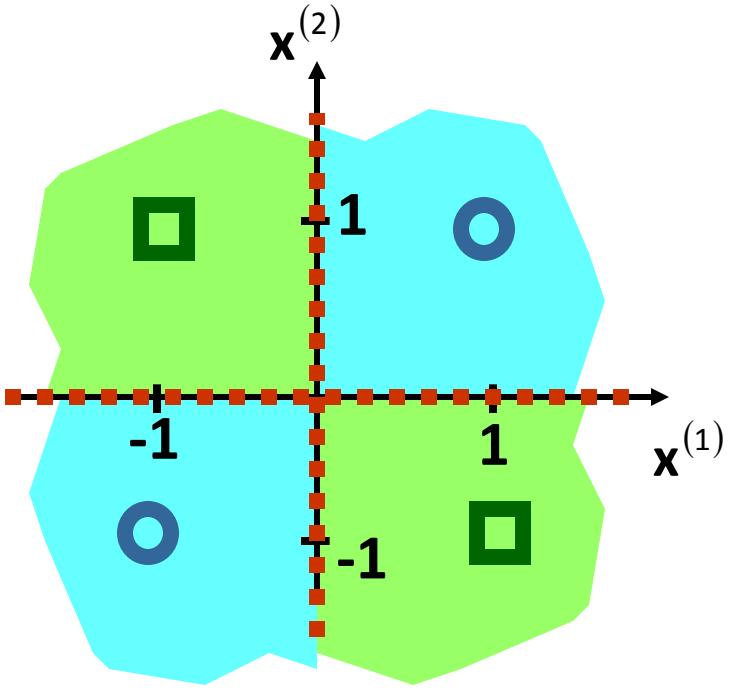
  - by plugging in $\mathbf{x}_1 = [1,-1]$, $\mathbf{x}_2 = [-1,1]$, $\mathbf{x}_3 = [1,1]$, $\mathbf{x}_4 = [-1,-1]$

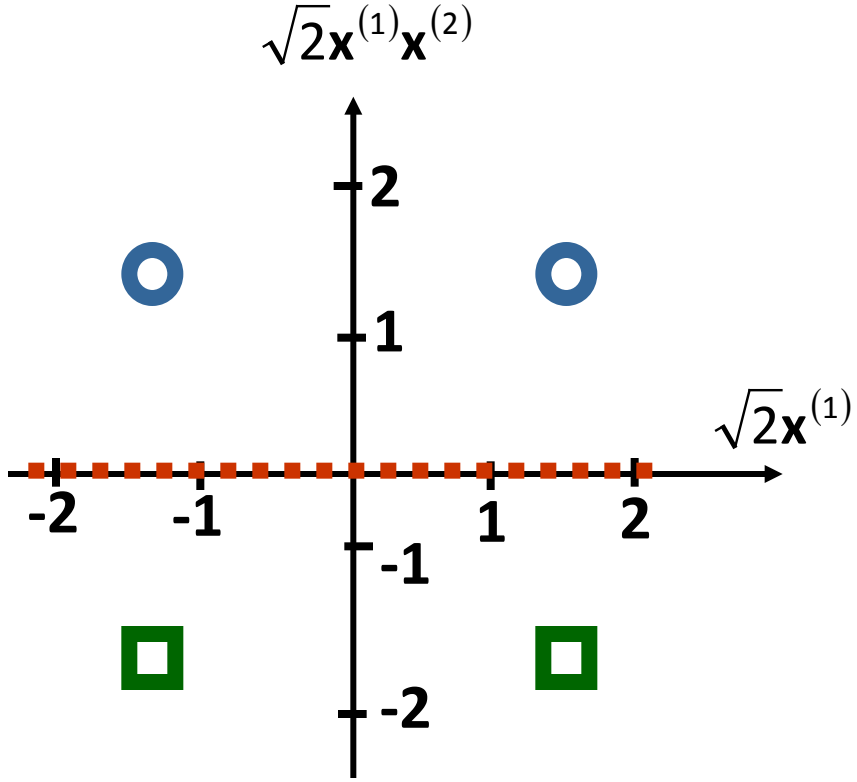
- Nonlinear discriminant function is

$$g(\mathbf{x}) = \mathbf{w}\varphi(\mathbf{x}) = \sum_{i=1}^{6} \mathbf{w}_i \varphi_i(\mathbf{x}) = \sqrt{2}\left(\sqrt{2}\mathbf{x}^{(1)}\mathbf{x}^{(2)}\right) = 2\mathbf{x}^{(1)}\mathbf{x}^{(2)}$$

SVM Example: XOR Problem

$$g(\mathbf{x}) = -2x^{(1)}x^{(2)}$$

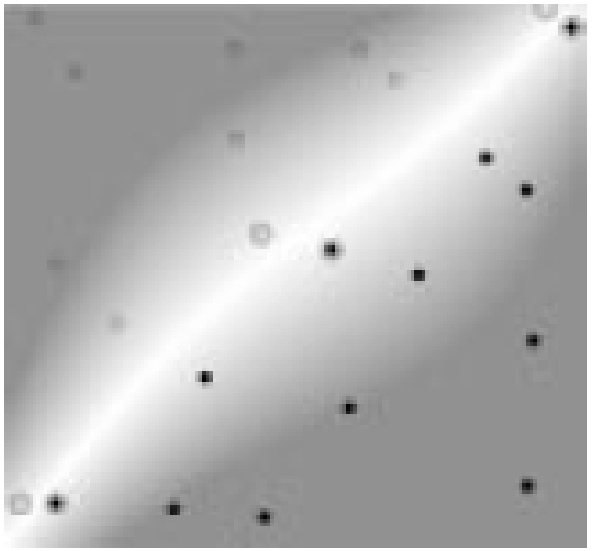


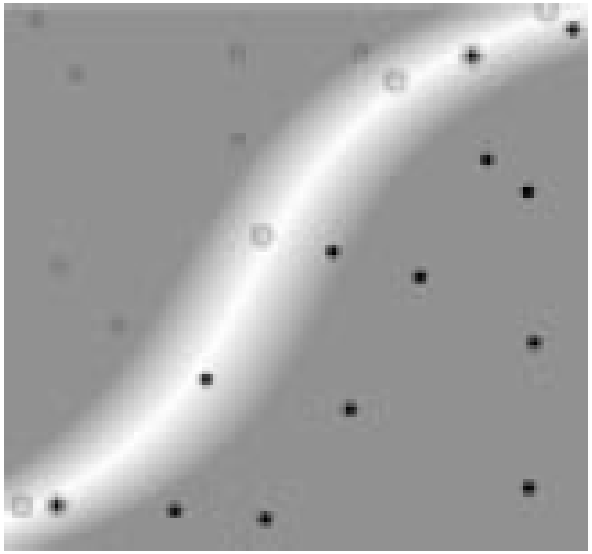


decision boundaries nonlinear

decision boundary is linear

Degree 3 Polynomial Kernel





- Left: In linearly separable case, decision boundary is roughly linear, indicating that dimensionality is controlled
- Right: nonseparable case is handled by a polynomial of degree 3

SVM Summary

- Advantages:
  - nice theory
  - good generalization properties
  - objective function has no local minima
  - can be used to find non linear discriminant functions
  - often works well in practice, even if not a lot of training data

- Disadvantages:
  - tends to be slower than other methods
  - quadratic programming is computationally expensive