

Support Vector Machines

Nazerfard, Ehsan
nazerfard@aut.ac.ir

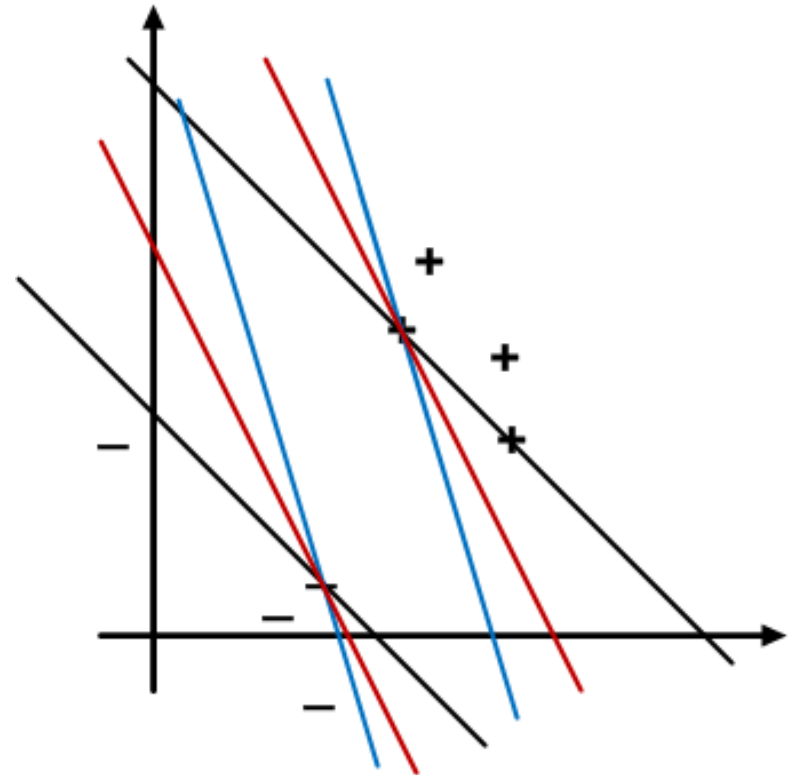
Support Vector Machines (SVMs)

□ Vladimir Vapnik

$$h: X \rightarrow \{-1, +1\}$$

- Widest street approach
- Maximum margin classifier

□ Which street is the best classifier: blue, red or black?



Support Vector Machines (cont.)

- Vladimir Vapnik

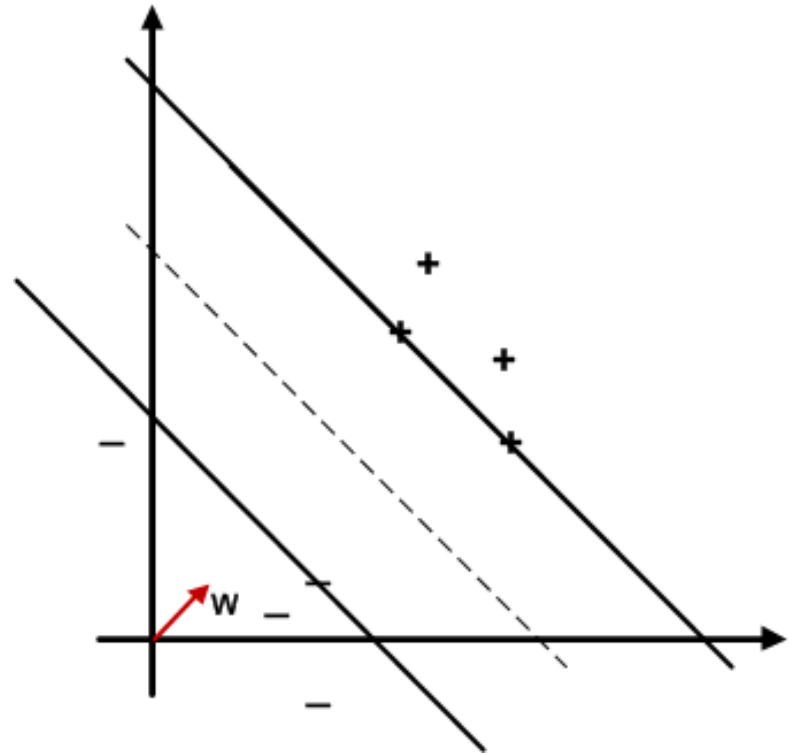
$$h: X \rightarrow \{-1, +1\}$$

- The separating hyperplane can be described as follows:

$$\mathbf{w} \cdot \mathbf{x} + b = 0^{**}$$

- Vector \mathbf{w} needs to be perpendicular to the street (why?)

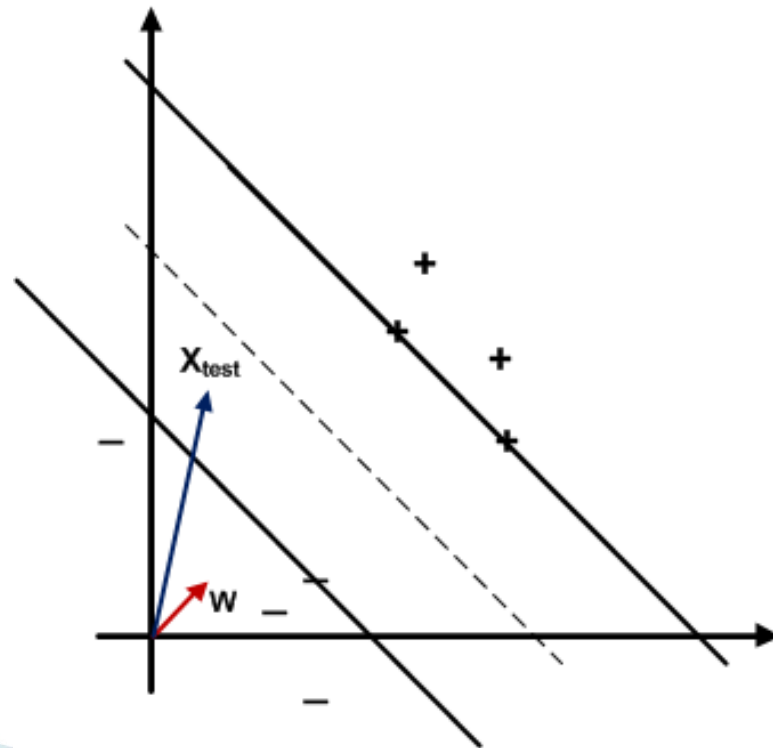
$$^{**} \mathbf{w}^T \mathbf{x} + b = 0$$



SVM Classifier

If $(\mathbf{w} \cdot \mathbf{x}_{test} \geq c)$ Then Class is +

If $(\mathbf{w} \cdot \mathbf{x}_{test} + b \geq 0)$ Then Class is +, s.t. $b = -c$

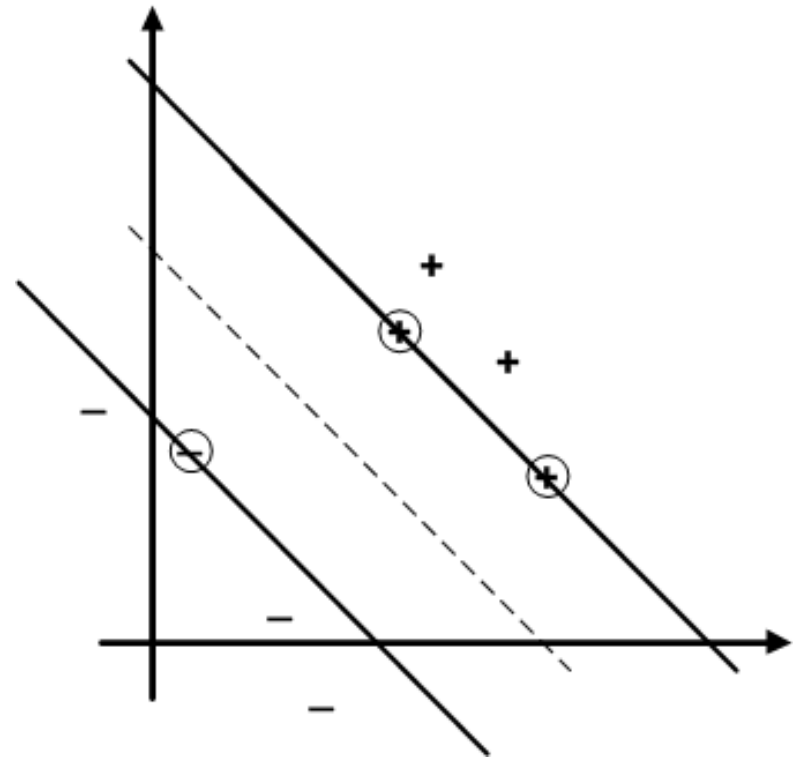


Constraints

□ Let assume the following constraints:

$$\begin{cases} \mathbf{w} \cdot \mathbf{x}_+ + b \geq 1 & (y = +1) \\ \mathbf{w} \cdot \mathbf{x}_- + b \leq -1 & (y = -1) \end{cases}$$

$$\Rightarrow y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq 0, \forall i$$



Constraints (cont.)

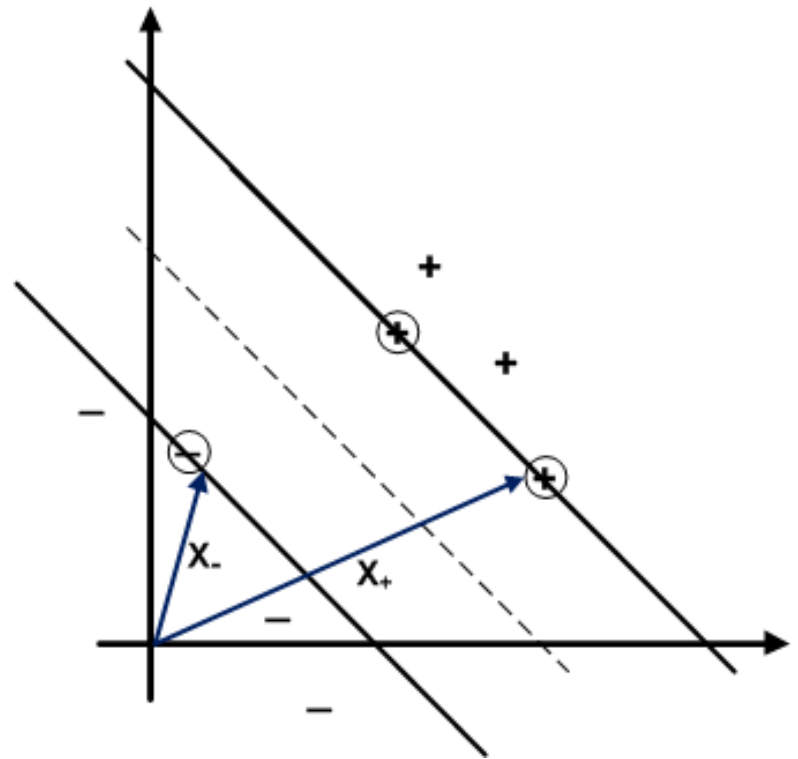
□ Let assume the following constraints:

$$\begin{cases} \mathbf{w} \cdot \mathbf{x}_+ + b \geq 1 & (y = +1) \\ \mathbf{w} \cdot \mathbf{x}_- + b \leq -1 & (y = -1) \end{cases}$$

$$\Rightarrow y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq 0, \forall i$$

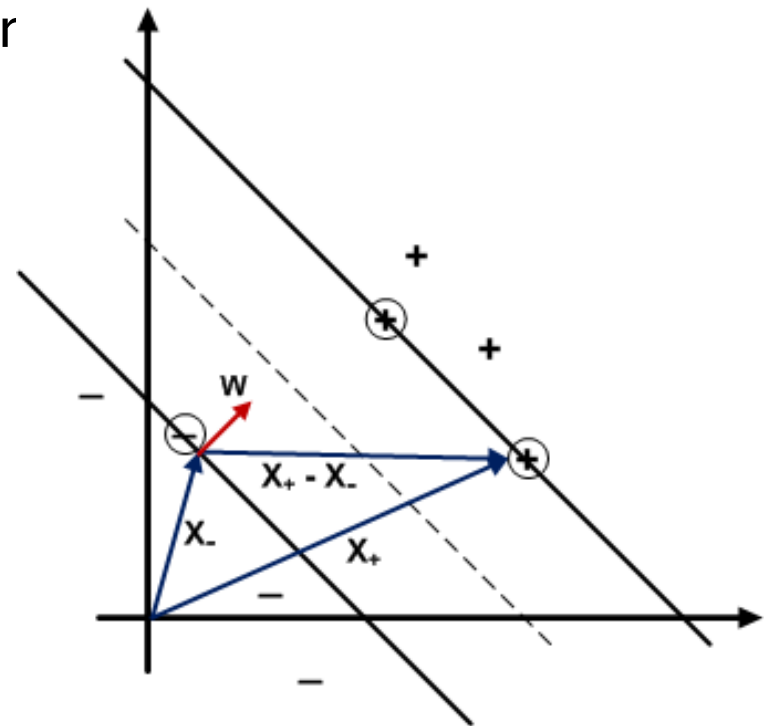
$$\boxed{y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 = 0}$$

IFF \mathbf{x}_i is a support vector



Width of the Street

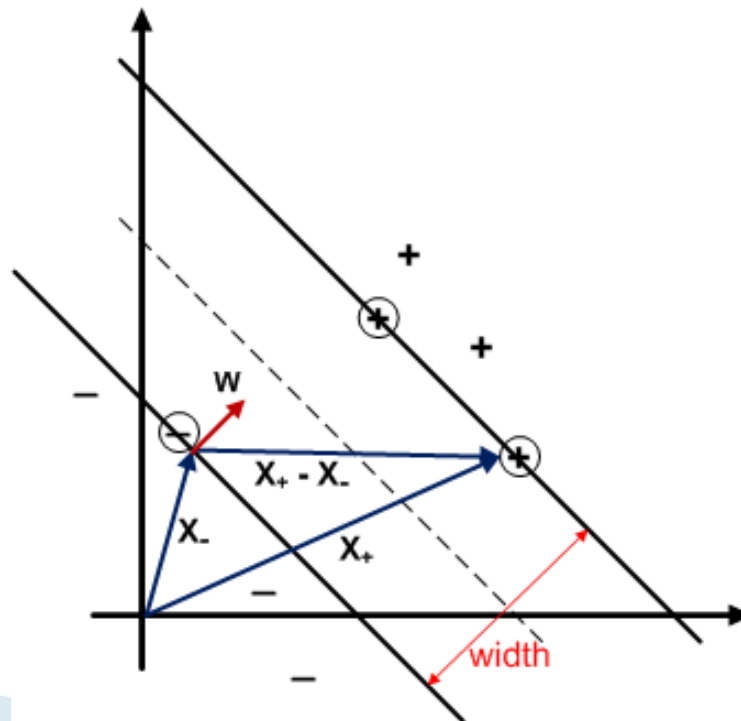
- Recall that vector w is perpendicular to the street.
- If we project vector $(x_+ - x_-)$ to vector w , the width of the street is obtained.
 - Vector w has to be a unit vector



Width of the Street (cont.)

$$Width = (x_+ - x_-) \cdot \frac{w}{||w||} = (w \cdot x_+ - w \cdot x_-) \cdot \frac{1}{||w||} = \frac{2}{||w||}$$

- Note that $w \cdot x_+ + b = 1$ and $w \cdot x_- + b = -1$



Optimization

$$Width = (x_+ - x_-) \cdot \frac{w}{||w||} = (w \cdot x_+ - w \cdot x_-) \cdot \frac{1}{||w||} = \frac{2}{||w||}$$

- The goal is to maximize $\frac{2}{||w||}$, or to minimize $||w||$,
subject to: $y_i(w \cdot x_i + b) - 1 \geq 0$

↓

minimize $\frac{1}{2} ||w||^2$

Lagrange Multipliers

- The goal is to minimize L , w.r.t. \mathbf{w}, b & maximize L , w.r.t. each α_i
 - Constrained optimization

$$L(\mathbf{w}, b) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_i \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1]$$

subject to: $\alpha_i \geq 0, \forall i$

- This quadratic optimization problem is known as the primal problem.

Primal Problem

- The goal is to minimize L , w.r.t. \mathbf{w}, b and maximize L , w.r.t. each α_i

$$L(\mathbf{w}, b) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_i \alpha_i [y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1]$$

$$\begin{cases} \frac{\partial L(\mathbf{w}, b)}{\partial \mathbf{w}} = \mathbf{w} - \sum_i \alpha_i y_i \mathbf{x}_i = 0 \rightarrow \boxed{\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i} \\ \frac{\partial L(\mathbf{w}, b)}{\partial b} = -\sum_i \alpha_i y_i = 0 \rightarrow \boxed{\sum_i \alpha_i y_i = 0} \end{cases}$$

- The Representer Theorem states that the solution \mathbf{w} can always be written as a linear combination of the training data.

Dual Problem

- If we substitute \mathbf{w} and b into the Lagrange multipliers formula:

$$L(\alpha) = \frac{1}{2} (\sum_i \alpha_i y_i \mathbf{x}_i) \cdot (\sum_j \alpha_j y_j \mathbf{x}_j) - (\sum_i \alpha_i y_i \mathbf{x}_i) \cdot (\sum_j \alpha_j y_j \mathbf{x}_j) - \sum_i \alpha_i y_i b + \sum_i \alpha_i$$

$$L(\alpha) = -\frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j + \sum_{i=1}^m \alpha_i, \forall i$$

subject to: $\alpha_i \geq 0, \forall i$

- This quadratic problem over α_i is known as the dual problem.
- It is shown that the problem space is convex, so it doesn't get stuck in local minimum/maximum.

Dual Problem (cont.)

- Quadratic optimization problem

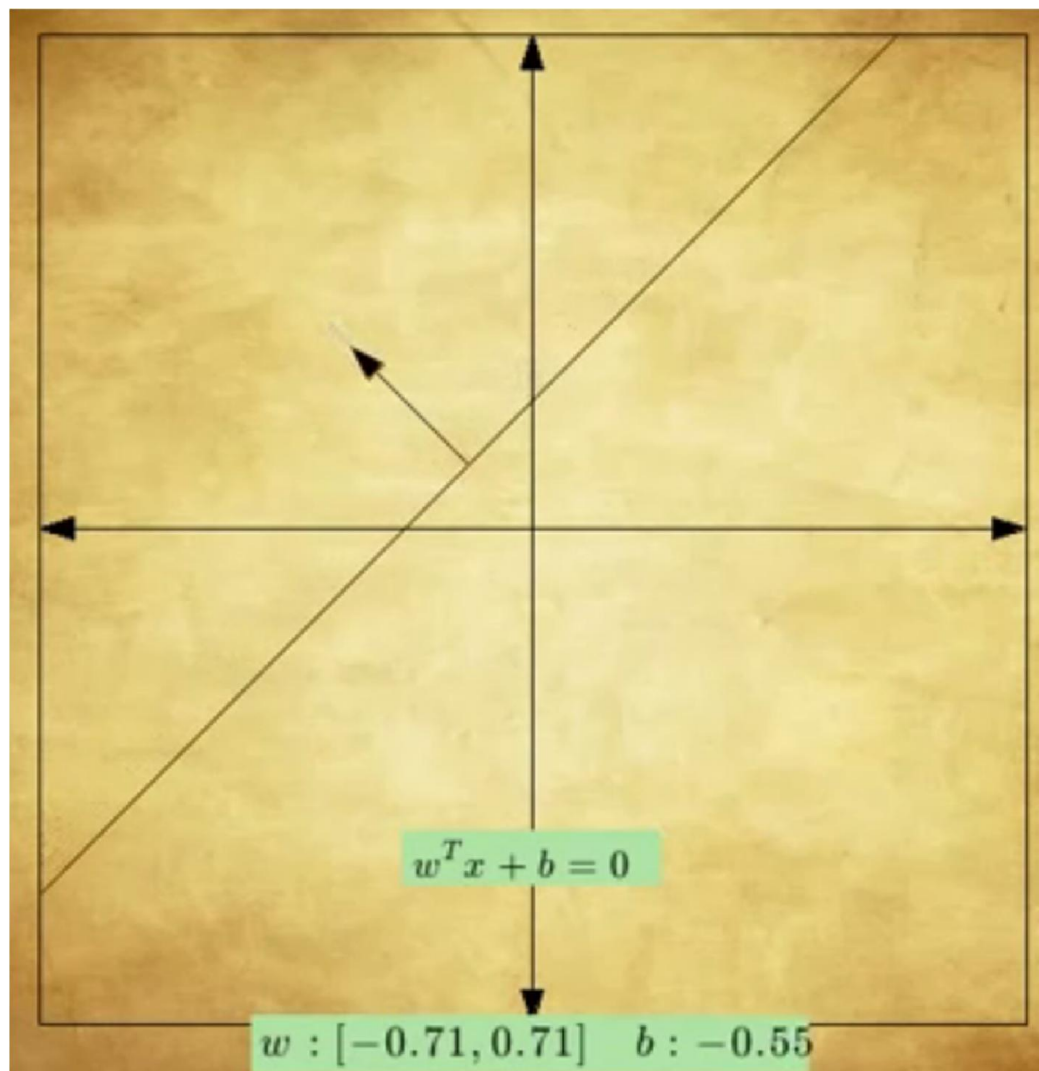
$$L(\boldsymbol{\alpha}) = -\frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j + \sum_{i=1}^m \alpha_i, \forall i$$

subject to: $\alpha_i \geq 0, \forall i$

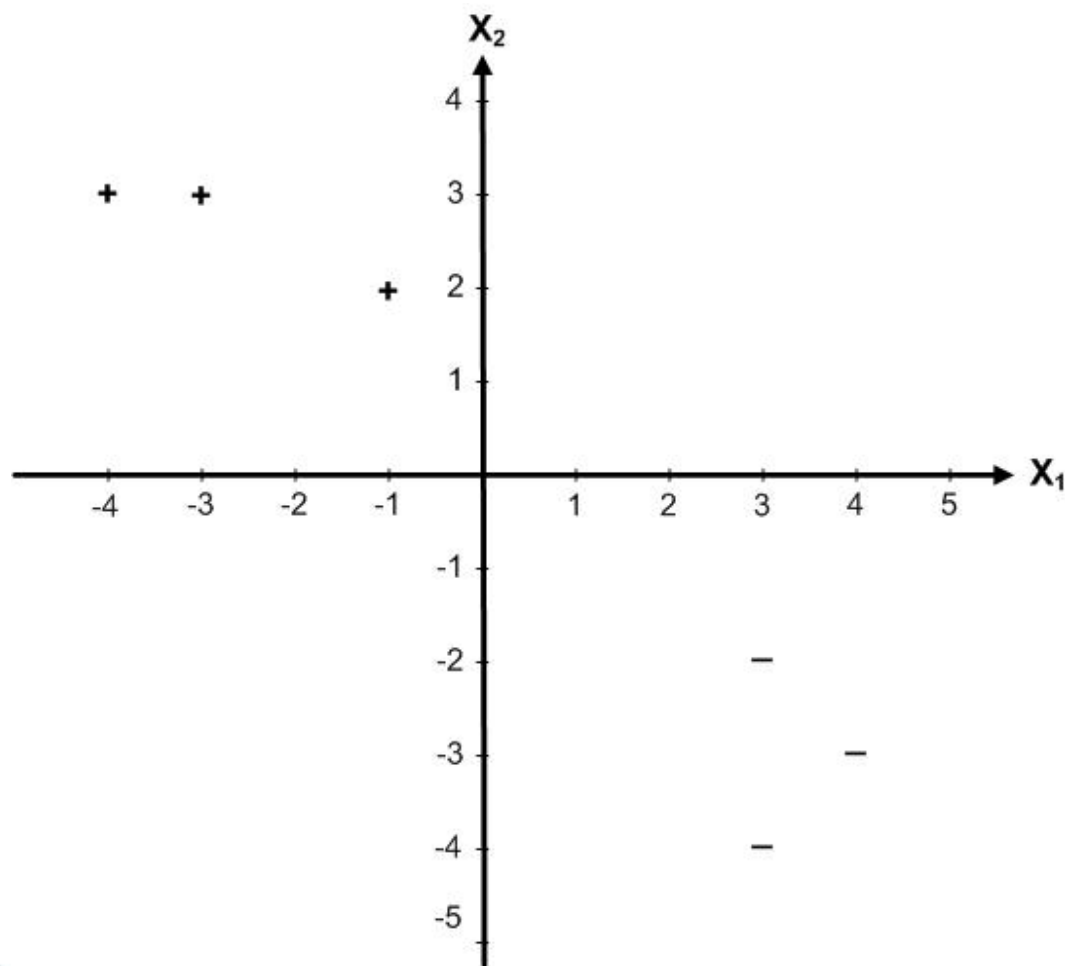
- Classification Rule

$$\sum_{i=1}^m \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x}_{test} + b \geq 0 \Rightarrow \text{Class is +}$$

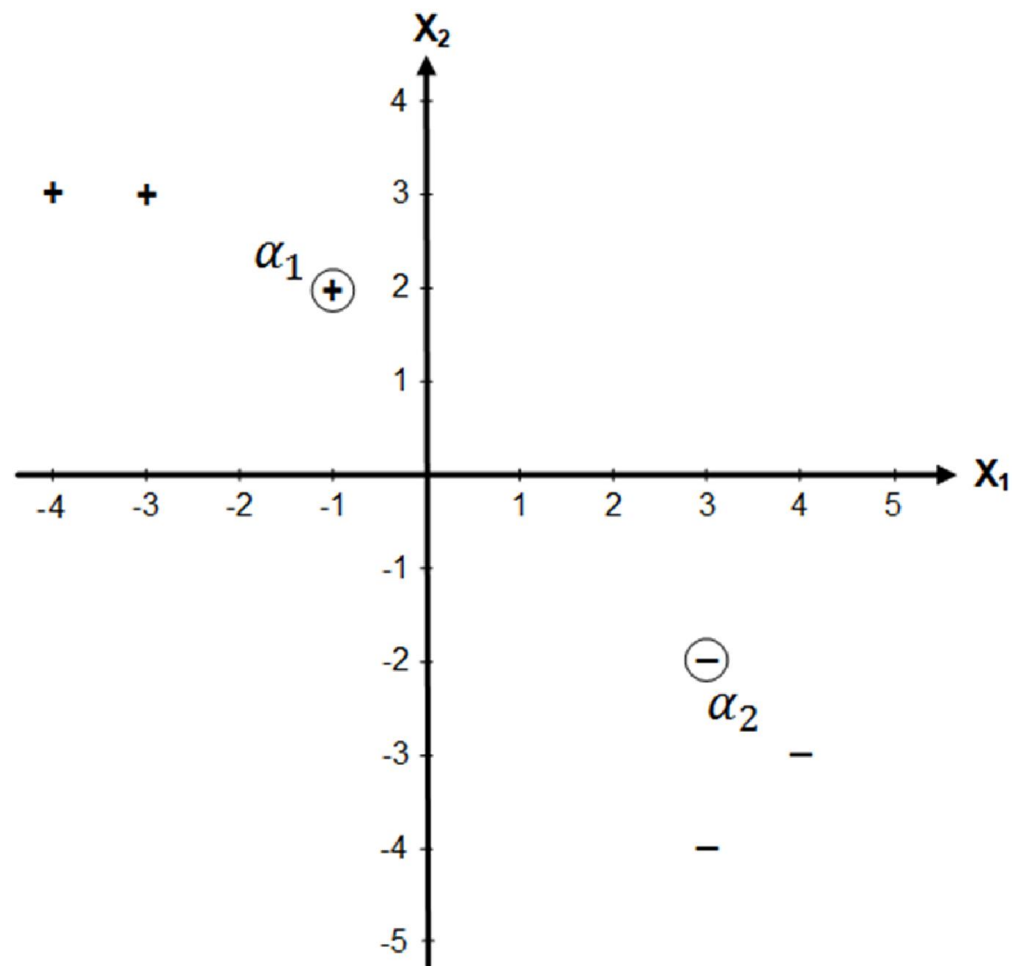
SVM Overview



SVM Example



Solution



Solution (cont.)

$$L(\boldsymbol{\alpha}) = -\frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j + \sum_i \alpha_i, (\mathbf{x}_i, \mathbf{x}_j \text{ are SV \& } \alpha_i, \alpha_j > 0)$$

$$\begin{aligned} L(\boldsymbol{\alpha}) &= -\frac{1}{2} (\alpha_1 \begin{bmatrix} -1 \\ 2 \end{bmatrix} - \alpha_2 \begin{bmatrix} 3 \\ -2 \end{bmatrix}) \cdot (\alpha_1 \begin{bmatrix} -1 \\ 2 \end{bmatrix} - \alpha_2 \begin{bmatrix} 3 \\ -2 \end{bmatrix}) + \alpha_1 + \alpha_2 = -\frac{1}{2} (\alpha_1^2 \\ &+ 6\alpha_1\alpha_2 + 9\alpha_2^2 + 4\alpha_1^2 + 8\alpha_1\alpha_2 + 4\alpha_2^2) + \alpha_1 + \alpha_2 = -\frac{1}{2} (5\alpha_1^2 + 14\alpha_1\alpha_2 \\ &+ 13\alpha_2^2 - 2\alpha_1 - 2\alpha_2) \end{aligned}$$

Solution (cont.)

$$L(\alpha) = -\frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j + \sum_i \alpha_i, (\mathbf{x}_i, \mathbf{x}_j \text{ are SV \& } \alpha_i, \alpha_j > 0)$$

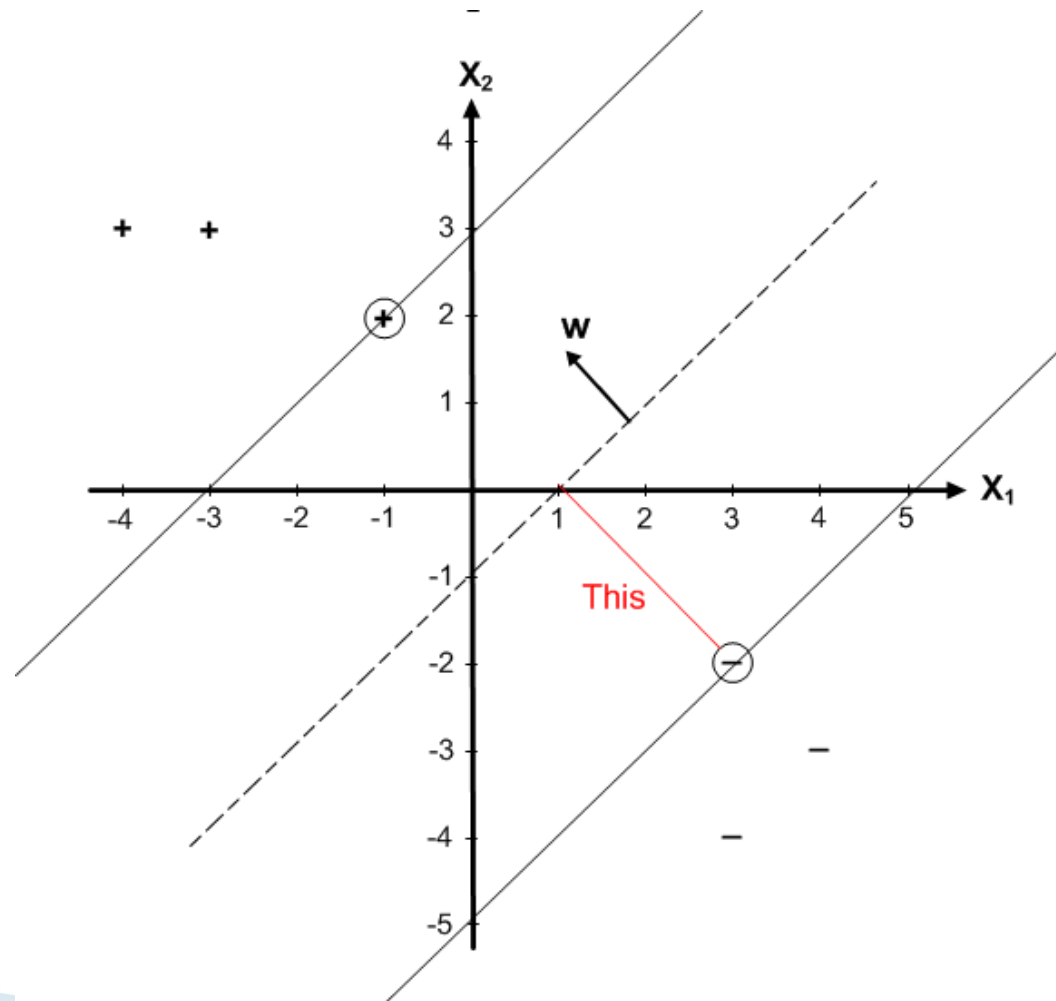
$$\begin{aligned} L(\alpha) &= -\frac{1}{2} (\alpha_1 \begin{bmatrix} -1 \\ 2 \end{bmatrix} - \alpha_2 \begin{bmatrix} 3 \\ -2 \end{bmatrix}) \cdot (\alpha_1 \begin{bmatrix} -1 \\ 2 \end{bmatrix} - \alpha_2 \begin{bmatrix} 3 \\ -2 \end{bmatrix}) + \alpha_1 + \alpha_2 = -\frac{1}{2} (\alpha_1^2 \\ &+ 6\alpha_1\alpha_2 + 9\alpha_2^2 + 4\alpha_1^2 + 8\alpha_1\alpha_2 + 4\alpha_2^2) + \alpha_1 + \alpha_2 = -\frac{1}{2} (5\alpha_1^2 + 14\alpha_1\alpha_2 \\ &+ 13\alpha_2^2 - 2\alpha_1 - 2\alpha_2) \end{aligned}$$

$$\Rightarrow \begin{cases} \frac{\partial L(\alpha)}{\partial \alpha_1} = 10\alpha_1 + 14\alpha_2 - 2 = 0 \\ \frac{\partial L(\alpha)}{\partial \alpha_2} = 14\alpha_1 + 26\alpha_2 - 2 = 0 \\ \sum_i \alpha_i y_i = 0 \end{cases} \Rightarrow \begin{cases} 6\alpha_1 + 10\alpha_2 = 1 \\ \alpha_1 = \alpha_2 \end{cases} \Rightarrow \boxed{\alpha_1 = \alpha_2 = \frac{1}{16}}$$

Solution (cont.)

$$\left\{ \begin{array}{l} \mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i \rightarrow \mathbf{w} = \frac{1}{16} \begin{bmatrix} -1 \\ 2 \end{bmatrix} - \frac{1}{16} \begin{bmatrix} 3 \\ -2 \end{bmatrix} \rightarrow \boxed{\mathbf{w} = \begin{bmatrix} -1/4 \\ 1/4 \end{bmatrix}} \\ y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 = 0 \rightarrow \begin{bmatrix} -\frac{1}{4} & \frac{1}{4} \end{bmatrix} \begin{bmatrix} -1 \\ 2 \end{bmatrix} + b - 1 = 0 \rightarrow \boxed{b = 1/4} \end{array} \right.$$

Verification



Verification (cont.)

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \rightarrow w_1 x_1 + w_2 x_2 + b = 0$$

Standard equation of a line: $x_2 = -\frac{w_1}{w_2} x_1 - \frac{b}{w_2}$

The separator line equation: $x_2 = x_1 - 1$

$$\begin{cases} x_2 = -\frac{w_1}{w_2} x_1 - \frac{b}{w_2} \\ x_2 = x_1 - 1 \end{cases} \Rightarrow \frac{w_1}{w_2} = -1, \frac{b}{w_2} = 1$$

$$\text{This} = 2\sqrt{2} = \frac{1}{\|\mathbf{w}\|} \Rightarrow \|\mathbf{w}\| = \frac{\sqrt{2}}{4} = \sqrt{w_1^2 + w_2^2}$$

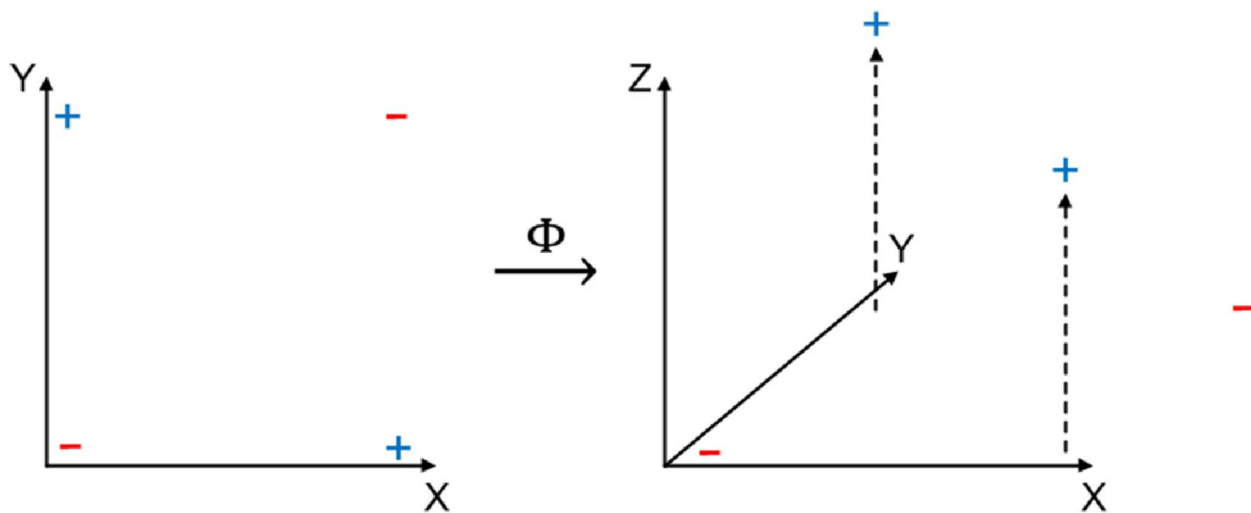
Verification (cont.)

$$\begin{cases} w_1 = -k \\ w_2 = k \\ b = k \end{cases} \quad (k > 0) \Rightarrow \begin{cases} \sqrt{w_1^2 + w_2^2} = \frac{\sqrt{2}}{4} \\ \sqrt{w_1^2 + w_2^2} = \sqrt{2} k \end{cases} \Rightarrow \begin{cases} w_1 = -\frac{1}{4} \\ w_2 = \frac{1}{4} \\ b = \frac{1}{4} \end{cases}$$

$$\begin{cases} \alpha_+ = \alpha_- \\ \begin{bmatrix} -1/4 \\ 1/4 \end{bmatrix} = \alpha_+ \begin{bmatrix} -1 \\ 2 \end{bmatrix} - \alpha_- \begin{bmatrix} 3 \\ -2 \end{bmatrix} \end{cases} \Rightarrow \boxed{\alpha_+ = \alpha_- = \frac{1}{16}}$$

Kernel Trick Intuition

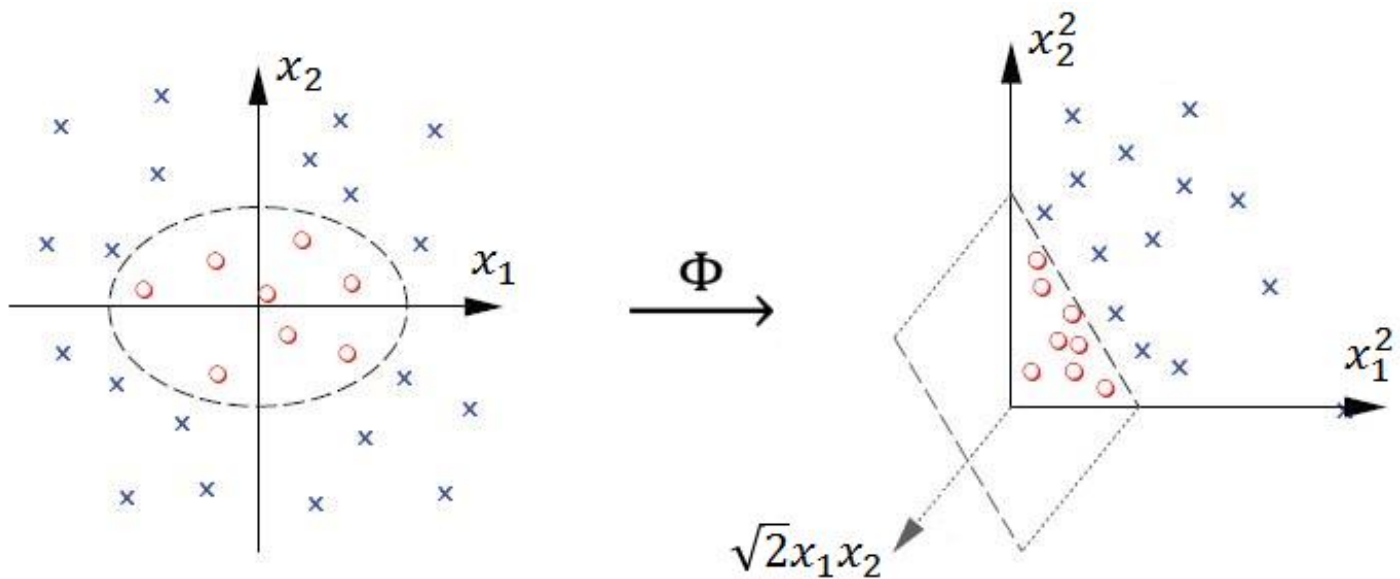
- SVM Solution for linearly inseparable problems, such as XOR
 - Kernel Trick: using a linear classifier to solve a non-linear problem.



Kernel Trick (cont.)

- Higher dimensional feature space – example:

$$(x_1, x_2) \rightarrow (x_1^2, x_2^2, \sqrt{2}x_1x_2)$$



Kernel Trick – Formal

$$\left\{ \begin{array}{l} L(\boldsymbol{\alpha}) = -\frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j + \sum_i \alpha_i \\ h(\mathbf{x}_{test}) = \text{sgn}(\sum_i \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x}_{test} + b) \end{array} \right.$$

1


□ SVM Transformation:

$$L(\boldsymbol{\alpha}) = -\frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \boldsymbol{\Phi}(\mathbf{x}_i) \cdot \boldsymbol{\Phi}(\mathbf{x}_j) + \sum_i \alpha_i$$


2

Kernel Trick – Formal (cont.)

□ SVM Transformation:

$$L(\alpha) = -\frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) + \sum_i \alpha_i$$


□ Kernel Trick (to avoid expensive data transformations)

$$L(\alpha) = -\frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \boxed{K(\mathbf{x}_i, \mathbf{x}_j)} + \sum_i \alpha_i$$


Kernel Trick – Formal (cont.)

$$\Phi: X \rightarrow \mathbb{Z}$$

$$L(\boldsymbol{\alpha}) = -\frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) + \sum_i \alpha_i$$

- If $K(\mathbf{x}_i, \mathbf{x}_j)$ is an inner product in some space, we are good!

Kernel Function Properties

- Function $K(\mathbf{x}, \mathbf{x}')$ is a valid kernel if:
 - It computes an inner product in some space \mathbb{Z} .
 - We just need to know that space \mathbb{Z} exists!
 - It is symmetric / commutative, i.e. $K(\mathbf{x}, \mathbf{x}') = K(\mathbf{x}', \mathbf{x})$.
 - It should (preferably) be positive semi-definite, i.e. satisfy Mercer's theorem.

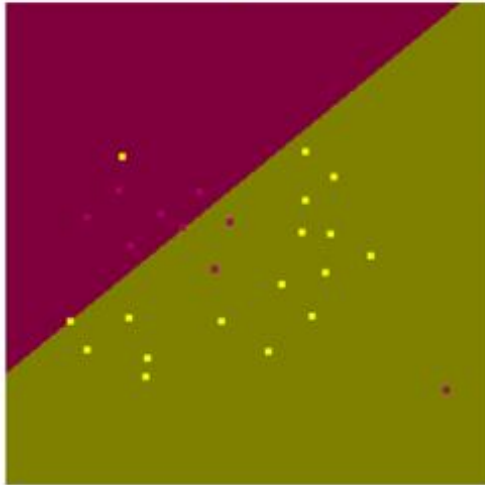
Kernel Types

□ The most frequently used kernel types:

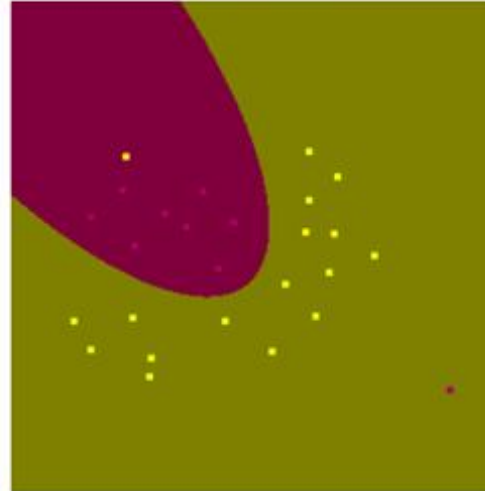
- Linear: $(\mathbf{x}_i \cdot \mathbf{x}_j + c)$
 - If $c = 0$, it is homogenous.
- Polynomial: $(\alpha \mathbf{x}_i \cdot \mathbf{x}_j + c)^d$, subject to: $d > 1$
- Gaussian RBF: $\exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right) = \exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2\right)$,
subject to: $\gamma = \frac{1}{2\sigma^2}$

Kernel Types

Linear kernel



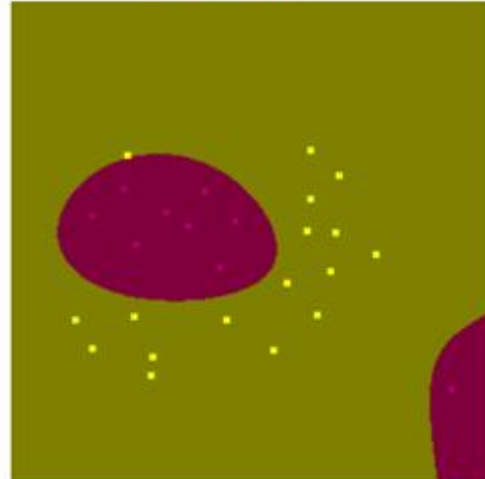
Poly degree=2



Poly degree=4



RBF kernel



Kernel Choice

- ❑ time of SVM learning: linear < poly < rbf
- ❑ ability to fit any data: linear < poly < rbf
- ❑ risk of overfitting: linear < poly < rbf
- ❑ risk of underfitting: rbf < poly < linear
- ❑ number of hyper-parameters: linear < rbf < poly

- ❑ So which one to choose? [1]
 - Occam's razor

Other Kernels

□ List of other well-known kernel functions:

- Exponential Kernel
- Laplacian Kernel
- ANOVA Kernel
- Hyperbolic Tangent (Sigmoid) Kernel
- Rational Quadratic Kernel
- Multiquadric Kernel
- Inverse Multiquadric Kernel
- Circular Kernel
- Spherical Kernel
- Power Kernel
- Log Kernel
- Spline Kernel
- B-Spline Kernel
- Bessel Kernel
- Cauchy Kernel
- Chi-Square Kernel
- Histogram Intersection Kernel
- T-Student Kernel
- Bayesian Kernel
- Wavelet Kernel

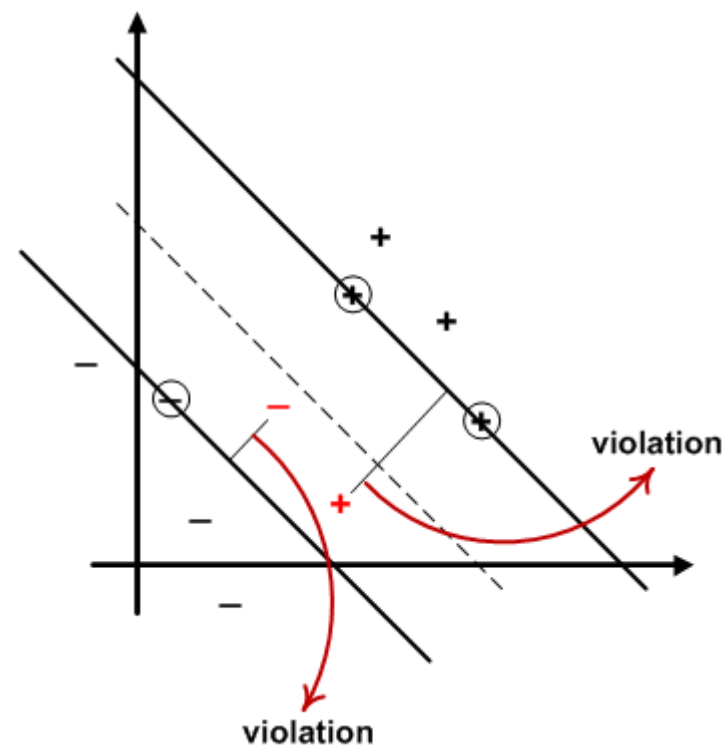
Soft Margin SVMs

□ What if some violations occur?

- Hard Margin SVM: no violation at all
- **Soft Margin SVM**: a slack variable, ξ_i , is defined for each data point, which indicates its violation amount.

$$\begin{cases} \xi_i = 0: \text{data is classified correctly.} \\ 0 < \xi_i \leq 1: \text{data lies between margin} \\ \quad \text{\& correct side of the plane.} \\ \xi_i > 1: \text{data is misclassified.} \end{cases}$$

- $y_i(\mathbf{w} \cdot \mathbf{x} + b) \geq 1 - \xi_i \quad (\xi_i \geq 0)$
- $\text{total violation} = \sum_{i=1}^m \xi_i \geq 0$



New Optimization – Primal

$$\left\{ \begin{array}{l} \text{minimize } \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^m \xi_i \\ \text{subject to: } y_i(\mathbf{w} \cdot \mathbf{x} + b) \geq 1 - \xi_i \text{ \& } \xi_i \geq 0 \\ \text{(for } i = 1, 2, \dots, m) \end{array} \right.$$

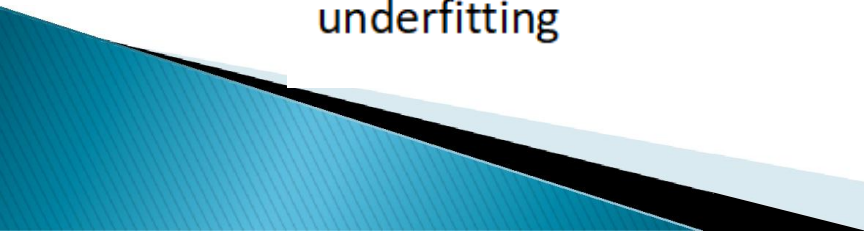
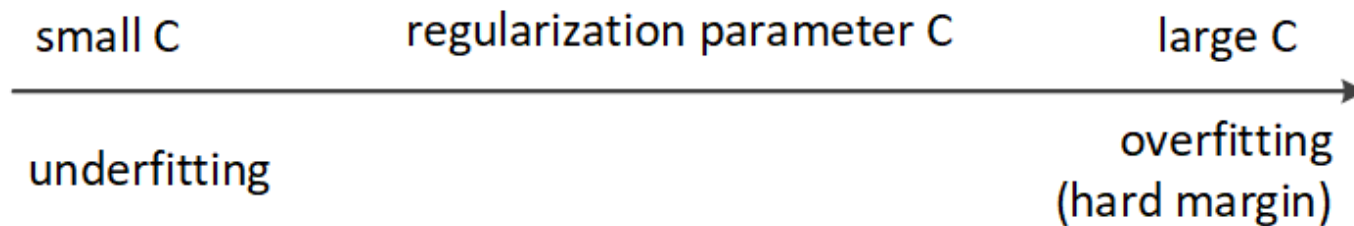
□ Two goals:

1. a hyperplane with the largest margin
2. a hyperplane that correctly separates as many instances as possible

New Optimization – Primal (cont.)

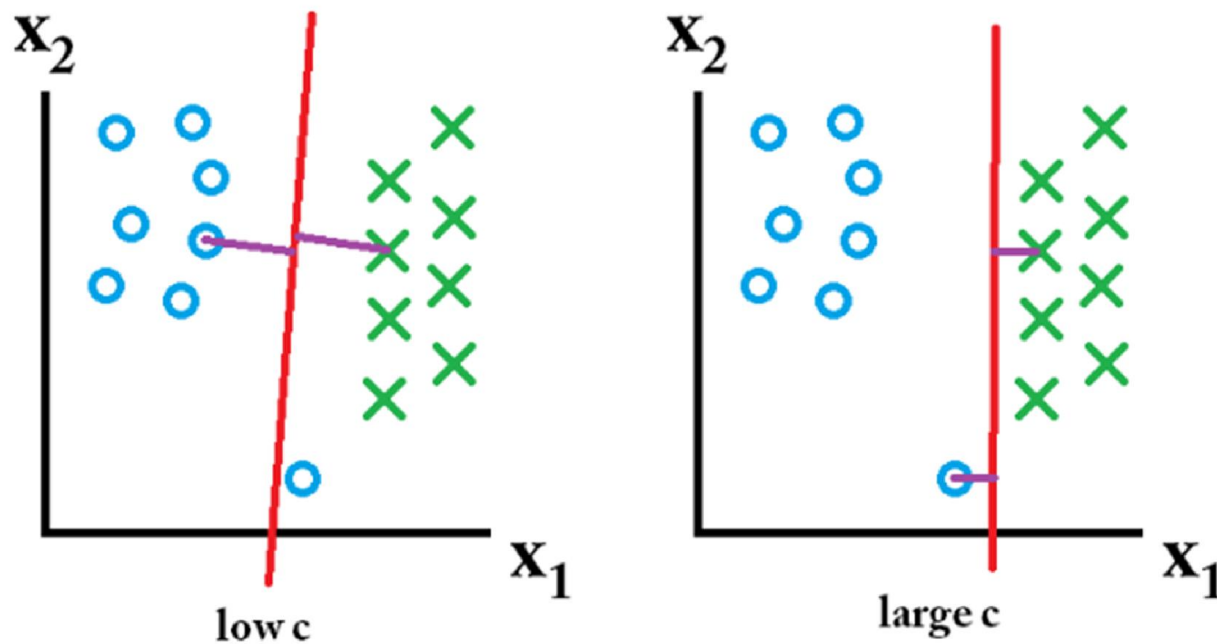
$$\begin{cases} \text{minimize } \frac{1}{2} ||\mathbf{w}'||^2 + C \sum_{i=1}^m \xi_i \\ \text{subject to: } y_i(\mathbf{w} \cdot \mathbf{x} + b) \geq 1 - \xi_i \quad \& \quad \xi_i \geq 0 \\ \quad \quad \quad (for \ i = 1, 2, \dots, m) \end{cases}$$

- The C parameter tells the SVM optimization how much you want to avoid misclassifying each training example.



Soft Margin SVM Decision Boundary

- The effect of the C parameter on the margin (purple).



Lagrange Multipliers

$$L(\mathbf{w}, b, \xi, \alpha, \beta) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^m \xi_i - \sum_{i=1}^m \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^m \beta_i \xi_i$$

- The goal is to minimize L w.r.t. \mathbf{w}, b, ξ & maximize L w.r.t. each $\alpha_i \geq 0$ and $\beta_i \geq 0$

Lagrange Multipliers (cont.)

$$L(\mathbf{w}, b, \xi, \alpha, \beta) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^m \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1] - \sum_{i=1}^m \xi_i (C - \alpha_i - \beta_i)$$

- The goal is to minimize L w.r.t. \mathbf{w}, b, ξ & maximize L w.r.t. each $\alpha_i \geq 0$ and $\beta_i \geq 0$

$$\left\{ \begin{array}{l} \frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^m \alpha_i y_i \mathbf{x}_i = 0 \\ \frac{\partial L}{\partial b} = - \sum_{i=1}^m \alpha_i y_i = 0 \\ \frac{\partial L}{\partial \xi_i} = C - \alpha_i - \beta_i = 0 \end{array} \right.$$

Dual Problem – Soft Margin SVM

$$\begin{cases} C - \alpha_i = \beta_i \\ \beta_i \geq 0 \end{cases} \rightarrow \alpha_i \leq C \rightarrow \boxed{0 \leq \alpha_i \leq C}$$

$$L(\boldsymbol{\alpha}) = -\frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j + \sum_{i=1}^m \alpha_i, \forall i$$

subject to: $0 \leq \alpha_i \leq C, \forall i$

Support Vector (SV) Types

$$\left\{ \begin{array}{l} \text{non - SV: } \alpha_i = 0 \rightarrow y_i(\mathbf{w} \cdot \mathbf{x}_i + b) > 1 \\ \text{margin SV: } 0 < \alpha_i < C \rightarrow y_i(\mathbf{w} \cdot \mathbf{x}_i + b) = 1 \\ \text{non - margin SV: } \alpha_i = C \rightarrow y_i(\mathbf{w} \cdot \mathbf{x}_i + b) < 1 \end{array} \right.$$

□ KKT conditions

Further Reading

- ❑ Mercer's theorem – positive semi-definite kernels
- ❑ Karush–Kuhn–Tucker (KKT) conditions
- ❑ Kernel Clustering
- ❑ Support Vector Regression

References

1. Support-vector networks, Corinna Cortes & Vladimir Vapnik, Machine Learning 20, 273–297, 1995.
2. Lecture on Support Vector Machines, Patrick Winston, Massachusetts Institute of Technology, 2010.
3. Learning from data, Yaser Abu-Mostafa, Malik Magdon-Ismail, Hsuan-Tien Lin, 2012.