Support Vector Machines

Nazerfard, Ehsan
nazerfard@aut.ac.ir

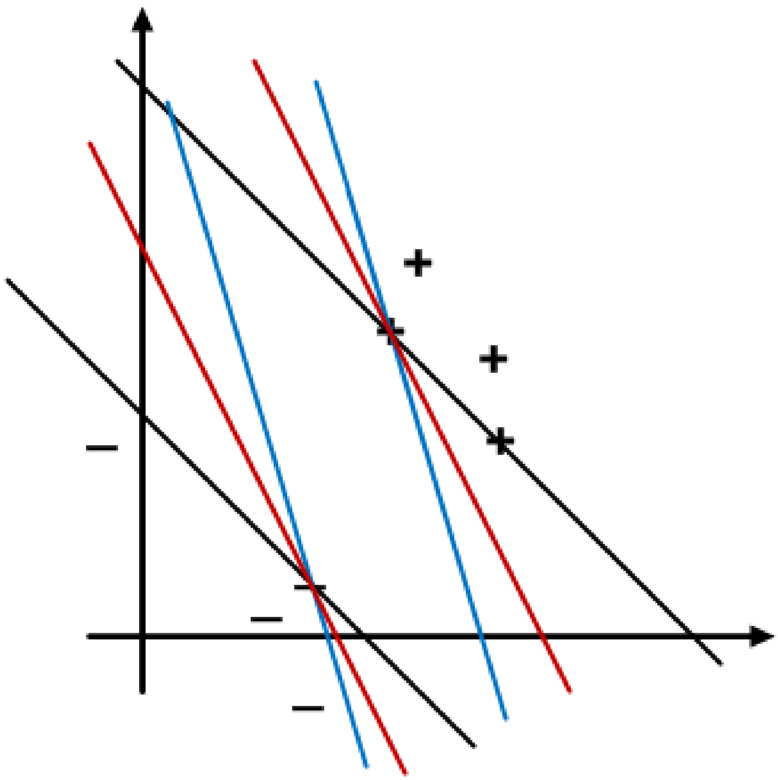
Support Vector Machines (SVMs)

- Vladimir Vapnik

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

  - Widest street approach
  - Maximum margin classifier

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



© Patric Winston, Lecture on SVMs, MIT

Support Vector Machines (cont.)

- Vladimir Vapnik

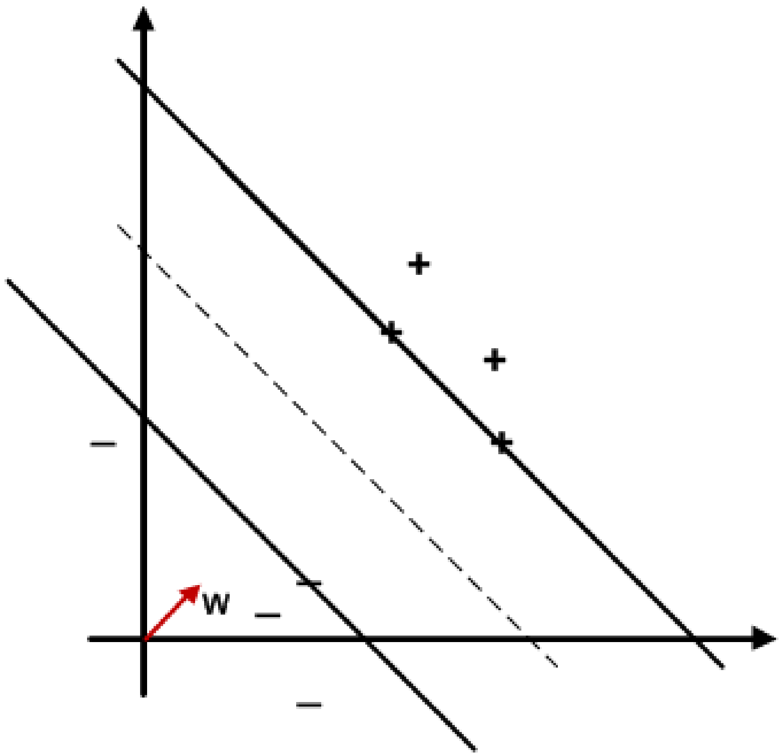
$$h: \mathrm{X} \rightarrow \{-1, +1\}$$

- The separating hyperplane can be described as follows:

$$\boldsymbol{w}.\boldsymbol{x} + b = 0^{**}$$

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

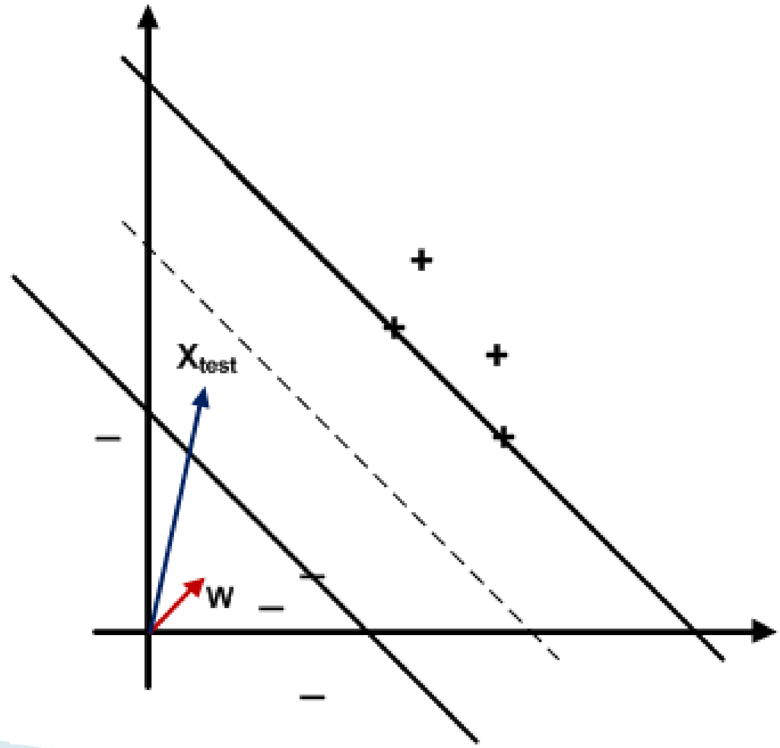
$$^{**}\ \boldsymbol{w}^T\boldsymbol{x} + b = 0$$



SVM Classifier

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

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

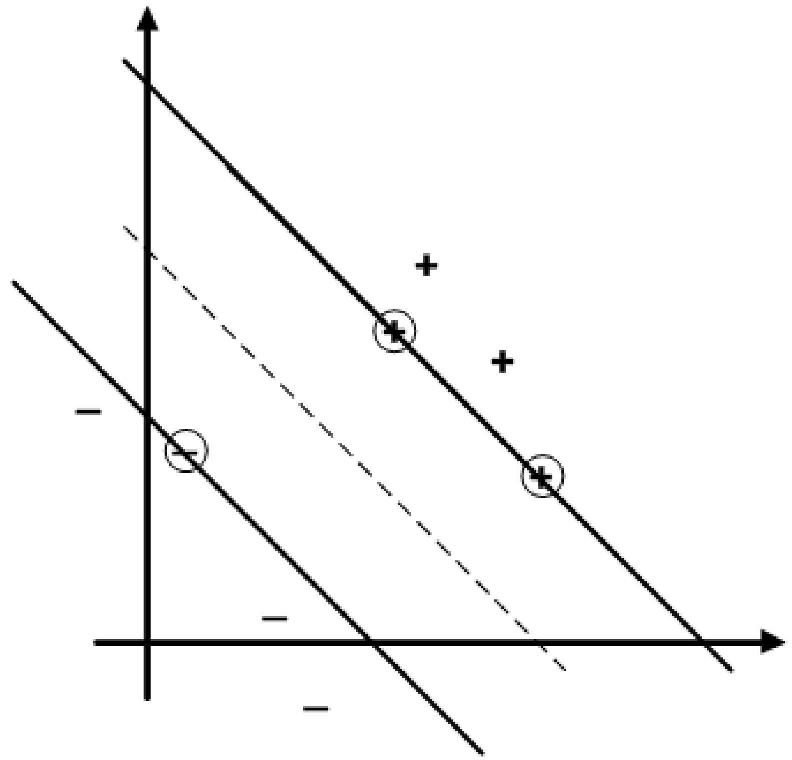


Constraints

- Let assume the following constraints:

$$\begin{cases} \boldsymbol{w}.\boldsymbol{x}_{+} + b \;\geq\; 1 \quad (y = +1) \\ \boldsymbol{w}.\boldsymbol{x}_{-} + b \;\leq\; -1 \quad (y = -1) \end{cases}$$

$$\Rightarrow y_i\ (\boldsymbol{w}.\boldsymbol{x_i} + b) - 1 \geq 0, \forall i$$



Constraints (cont.)

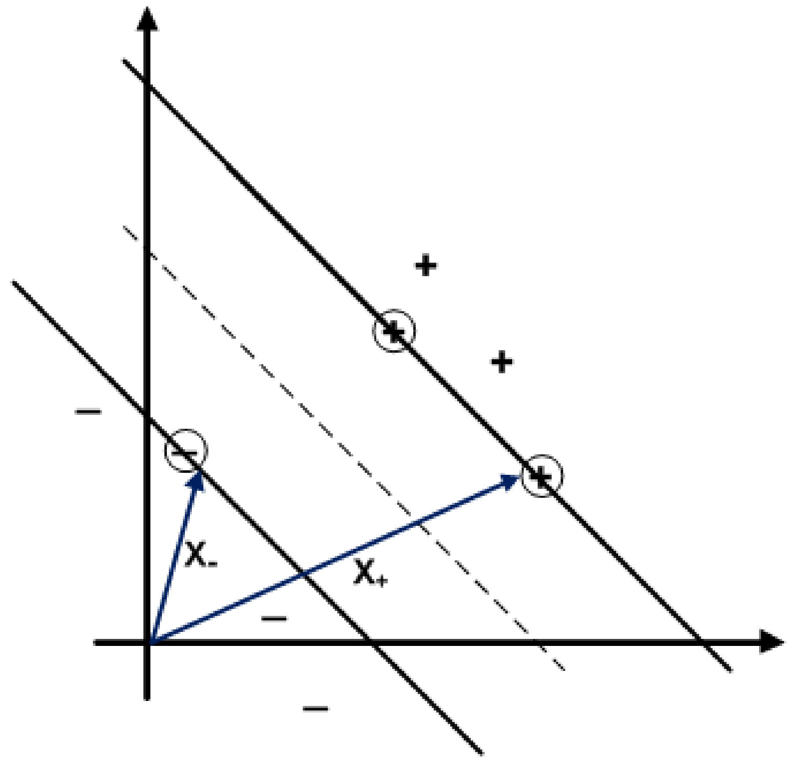
- Let assume the following constraints:

$$\begin{cases} \boldsymbol{w}.\boldsymbol{x}_+ + b \geq 1 \quad (y = +1) \\ \boldsymbol{w}.\boldsymbol{x}_- + b \leq -1 \quad (y = -1) \end{cases}$$

$$\Rightarrow \; y_i(\boldsymbol{w}.\boldsymbol{x_i} + b) - 1 \geq 0, \forall i$$

$$\boxed{y_i(\boldsymbol{w}.\boldsymbol{x_i} + b) - 1 = 0}$$

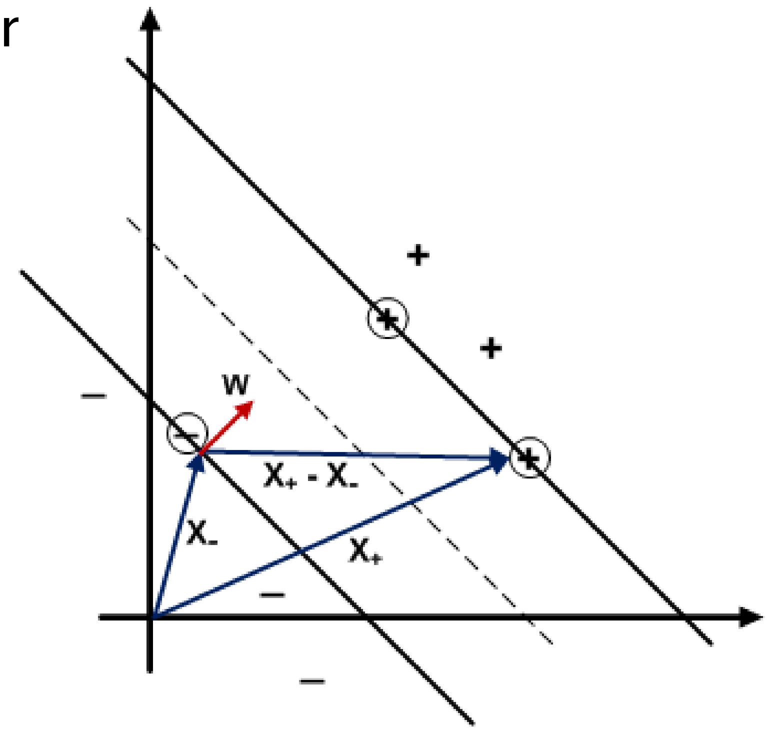
IFF $\boldsymbol{x_i}$ is a support vector



Width of the Street

- Recall that vector $\boldsymbol{w}$ is perpendicular to the street.

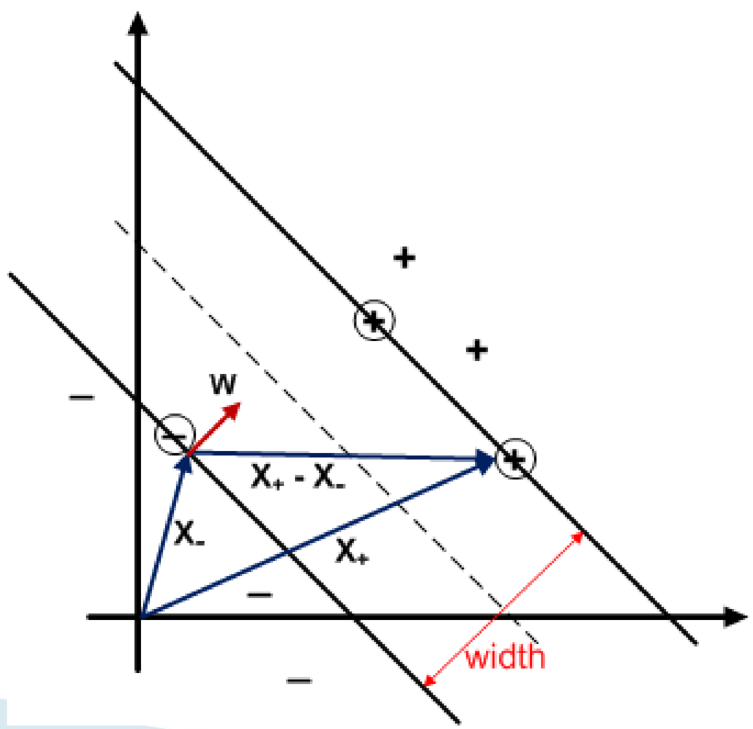
- If we project vector $(\boldsymbol{x}_{+} - \boldsymbol{x}_{-})$ to vector $\boldsymbol{w}$, the width of the street is obtained.
  - Vector $\boldsymbol{w}$ has to be a unit vector



Width of the Street (cont.)

$$Width = (\boldsymbol{x}_{+} - \boldsymbol{x}_{-}) \, . \, \frac{\boldsymbol{w}}{||\boldsymbol{w}||} = (\boldsymbol{w}.\boldsymbol{x}_{+} - \boldsymbol{w}.\boldsymbol{x}_{-}) \, . \, \frac{1}{||\boldsymbol{w}||} = \frac{2}{||\boldsymbol{w}||}$$

- Note that $\boldsymbol{w}.\boldsymbol{x}_{+} + b = 1$ and $\boldsymbol{w}.\boldsymbol{x}_{-} + b = -1$



Optimization

$$Width = (\boldsymbol{x}_+ - \boldsymbol{x}_-) \,.\, \frac{\boldsymbol{w}}{||\boldsymbol{w}||} = (\boldsymbol{w}.\boldsymbol{x}_+ - \boldsymbol{w}.\boldsymbol{x}_-) \,.\, \frac{1}{||\boldsymbol{w}||} = \frac{2}{||\boldsymbol{w}||}$$

- The goal is to maximize $\frac{2}{||\boldsymbol{w}||}$, or to minimize $||\boldsymbol{w}||$,

  subject to: $y_i(\boldsymbol{w}.\boldsymbol{x_i} + b) - 1 \geq 0$

$$\Downarrow$$

$$\textbf{minimize}\,\frac{\mathbf{1}}{\mathbf{2}}||\boldsymbol{w}||^{\mathbf{2}}$$

Lagrange Multipliers

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

$$\boxed{\begin{gathered} L(\boldsymbol{w}, b) = \frac{1}{2}||\boldsymbol{w}||^2 - \sum_i \alpha_i \left[y_i(\boldsymbol{w}.\boldsymbol{x_i} + b) - 1\right] \\ \textit{subject to}\text{:}\ \alpha_i \geq 0, \forall \mathrm{i} \end{gathered}}$$

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

Primal Problem

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

$$L(\boldsymbol{w}, b) = \frac{1}{2}||\boldsymbol{w}||^2 - \sum_i \alpha_i \left[y_i(\boldsymbol{w}.\boldsymbol{x_i} + b) - 1\right]$$

$$\begin{cases} \frac{\partial L(\boldsymbol{w},b)}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{\boldsymbol{i}} \alpha_i y_i \boldsymbol{x_i} = 0 \;\rightarrow \boxed{\boldsymbol{w} = \sum_{\boldsymbol{i}} \alpha_i y_i \boldsymbol{x_i}} \\ \frac{\partial L(\boldsymbol{w},b)}{\partial b} = -\sum_i \alpha_i y_i = 0 \;\rightarrow \boxed{\sum_i \alpha_i y_i = 0} \end{cases}$$

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

Dual Problem

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

$$L(\boldsymbol{\alpha}) = \frac{1}{2}(\sum_i \alpha_i y_i \boldsymbol{x_i}).(\sum_j \alpha_j y_j \boldsymbol{x_j}) - (\sum_i \alpha_i y_i \boldsymbol{x_i}).(\sum_j \alpha_j y_j \boldsymbol{x_j}) - \sum_i \alpha_i y_i\, b + \sum_i \alpha_i$$

$$\boxed{\begin{gathered} L(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \boldsymbol{x_i}.\boldsymbol{x_j} + \sum_{i=1}^{m} \alpha_i\,, \forall i \\ subject\ to\colon\ \alpha_i \geq 0, \forall i \end{gathered}}$$

- 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.)

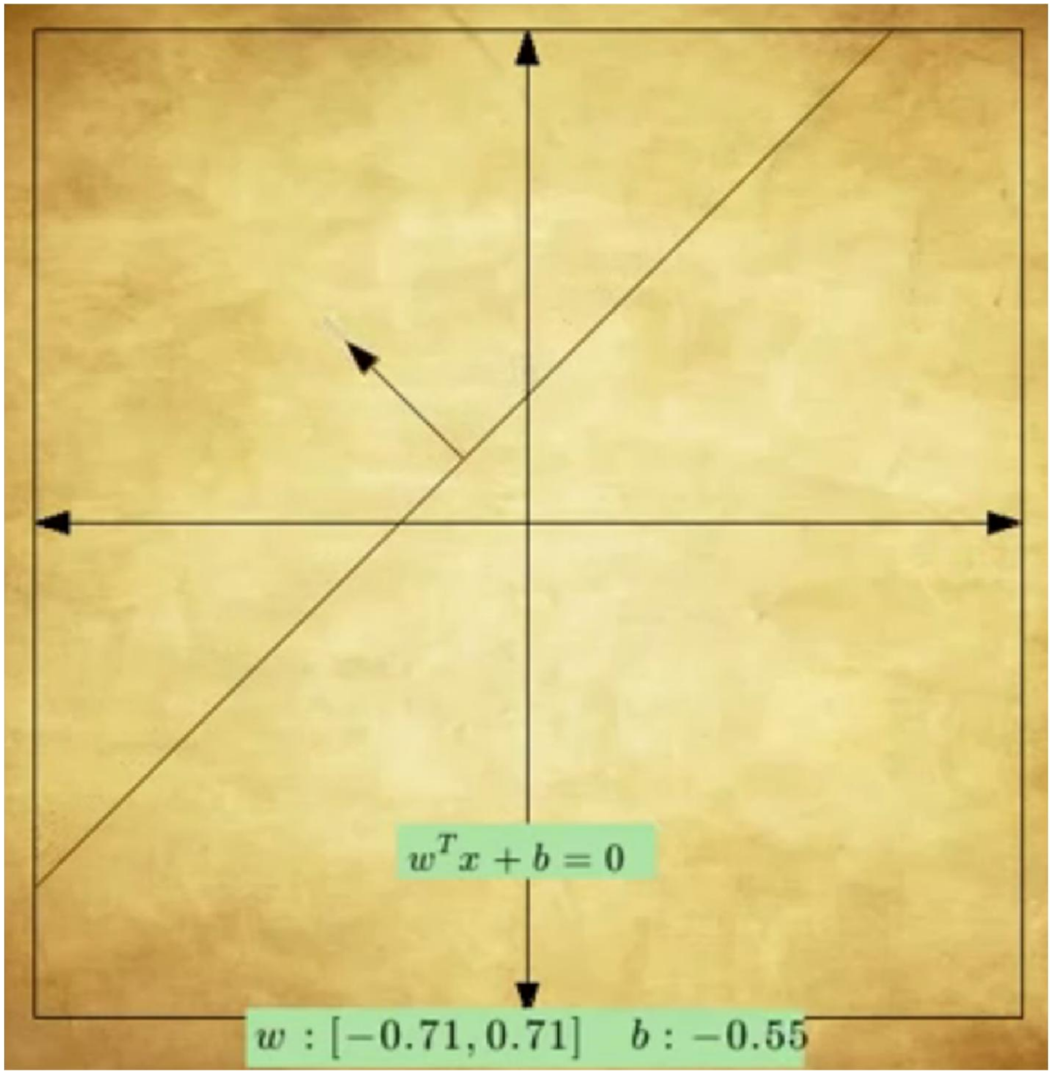
- Quadratic optimization problem

$$\boxed{\begin{gathered} L(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \boldsymbol{x_i}.\boldsymbol{x_j} + \sum_{i=1}^{m}\alpha_i \,, \forall i \\ subject\ to{:}\ \alpha_i \geq 0, \forall i \end{gathered}}$$

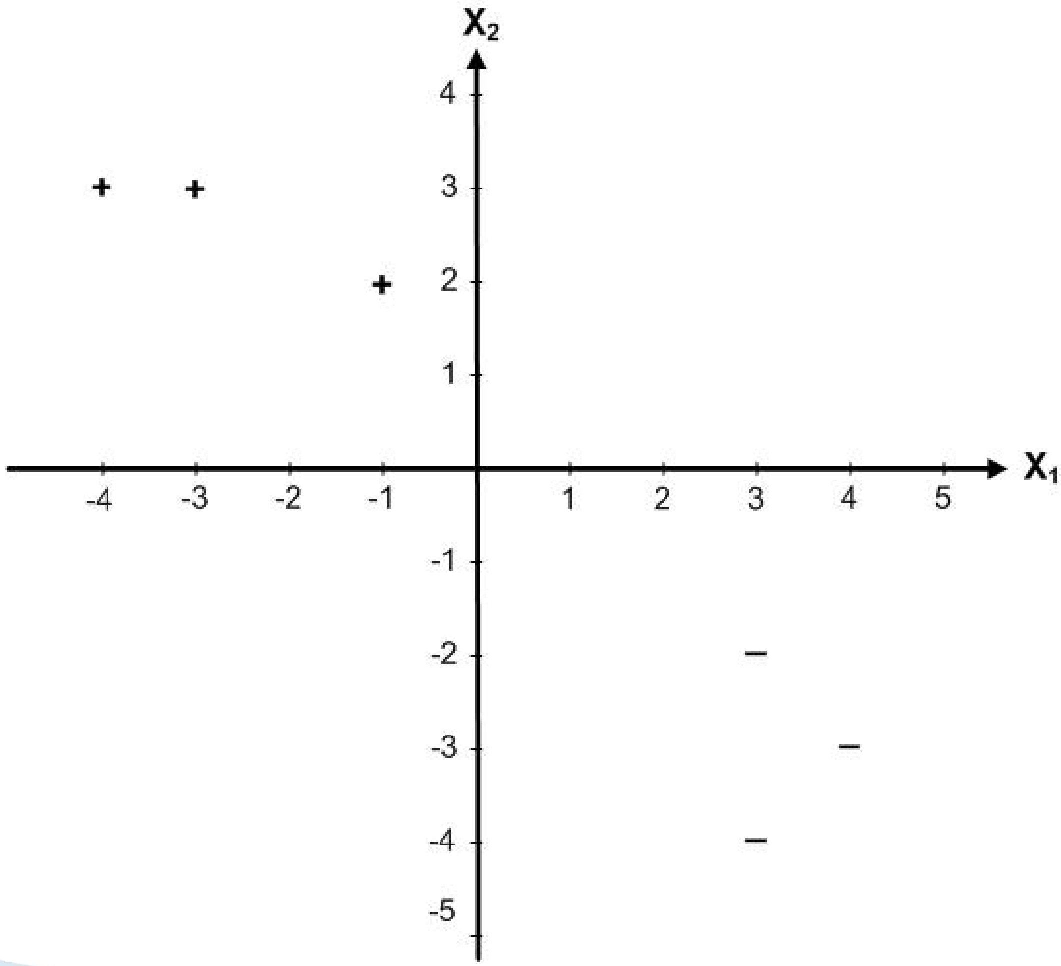
- Classification Rule

$$\boxed{\sum_{i=1}^{m}\alpha_i y_i \boldsymbol{x_i}.\boldsymbol{x_{test}} + b \geq 0 \Rightarrow Class\ is\ +}$$

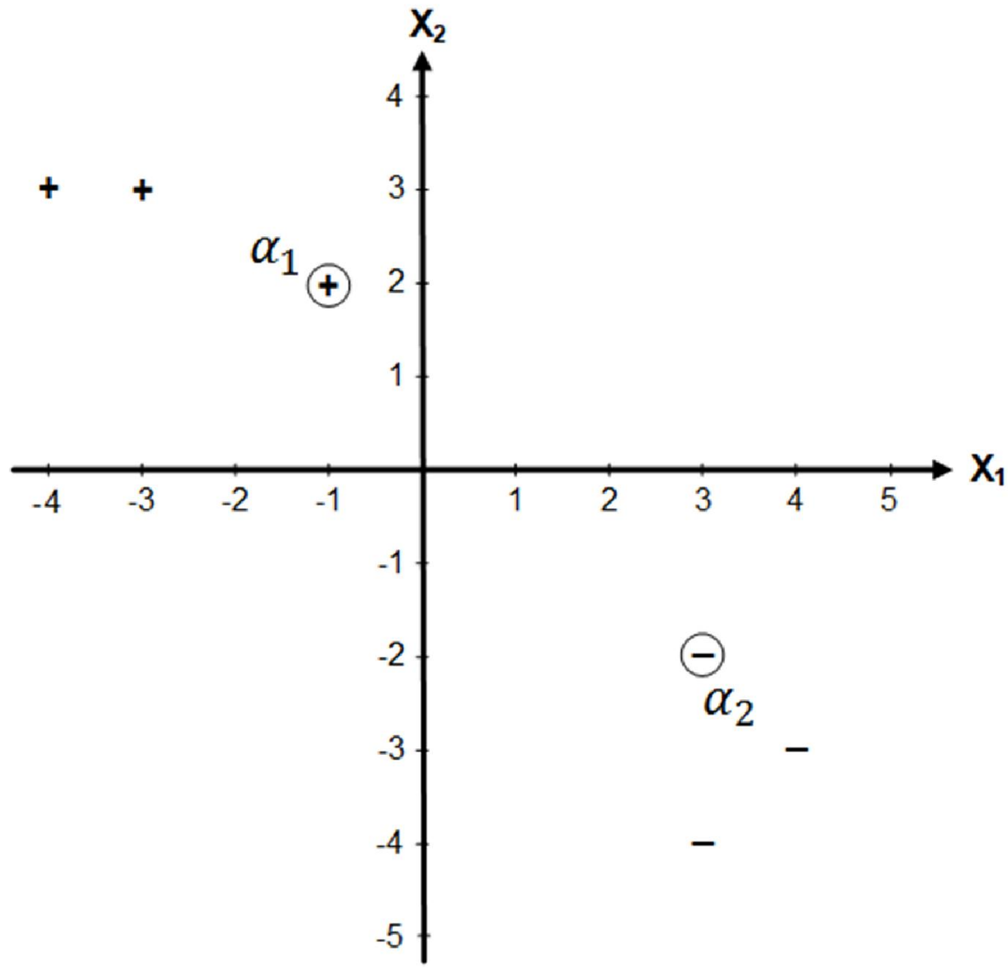
SVM Overview



SVM Example



Solution



Solution (cont.)

$$L(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i}\sum_{j}\alpha_i\alpha_j y_i y_j \boldsymbol{x_i}.\boldsymbol{x_j} + \sum_{i}\alpha_i \text{ , } (\boldsymbol{x_i} \text{ , } \boldsymbol{x_j} \text{ are SV \& } \alpha_i, a_j > 0)$$

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

Solution (cont.)

$$L(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \boldsymbol{x_i}.\boldsymbol{x_j} + \sum_i \alpha_i \text{ , } (\boldsymbol{x_i}\text{ , }\boldsymbol{x_j} \text{ are SV \& } \alpha_i, a_j > 0)$$

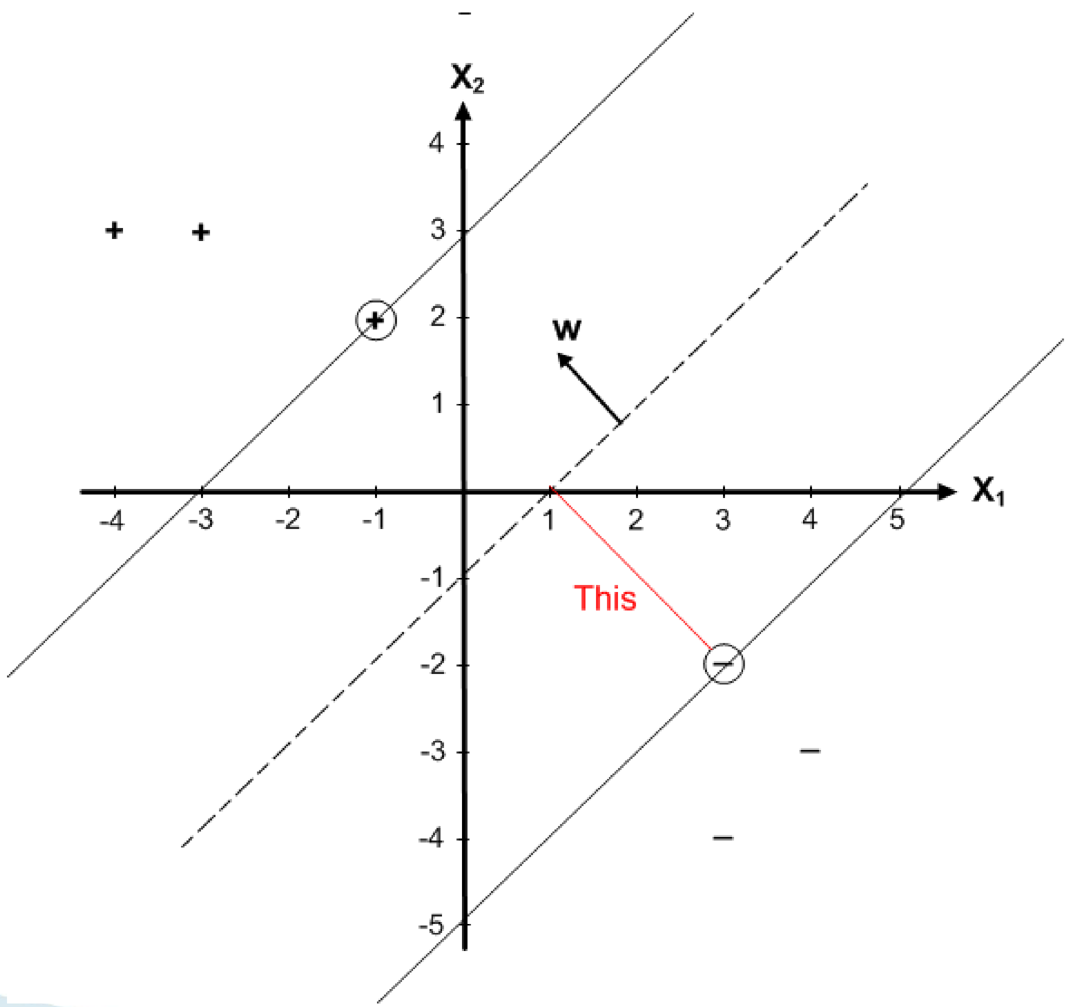
$$L(\boldsymbol{\alpha}) = -\tfrac{1}{2}(\alpha_1 \begin{bmatrix} -1 \\ 2 \end{bmatrix} - \alpha_2 \begin{bmatrix} 3 \\ -2 \end{bmatrix}).(\alpha_1 \begin{bmatrix} -1 \\ 2 \end{bmatrix} - \alpha_2 \begin{bmatrix} 3 \\ -2 \end{bmatrix}) + \alpha_1 + \alpha_2 = -\tfrac{1}{2}(\alpha_1^2 + 6\alpha_1\alpha_2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

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

Solution (cont.)

$$\begin{cases} \boldsymbol{w} = \sum_i \alpha_i y_i \boldsymbol{x_i} \rightarrow \boldsymbol{w} = \frac{1}{16}\begin{bmatrix} -1 \\ 2 \end{bmatrix} - \frac{1}{16}\begin{bmatrix} 3 \\ -2 \end{bmatrix} \rightarrow \boxed{\boldsymbol{w} = \begin{bmatrix} -1/4 \\ 1/4 \end{bmatrix}} \\ y_i(\boldsymbol{w}.\boldsymbol{x_i} + b) - 1 = 0 \rightarrow \begin{bmatrix} -\frac{1}{4} & \frac{1}{4} \end{bmatrix}\begin{bmatrix} -1 \\ 2 \end{bmatrix} + b - 1 = 0 \rightarrow \boxed{b = 1/4} \end{cases}$$

Verification



Verification (cont.)

$$\boldsymbol{w}.\boldsymbol{x} + b = 0 \;\rightarrow w_1x_1 + w_2x_2 + b = 0$$

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

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

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

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

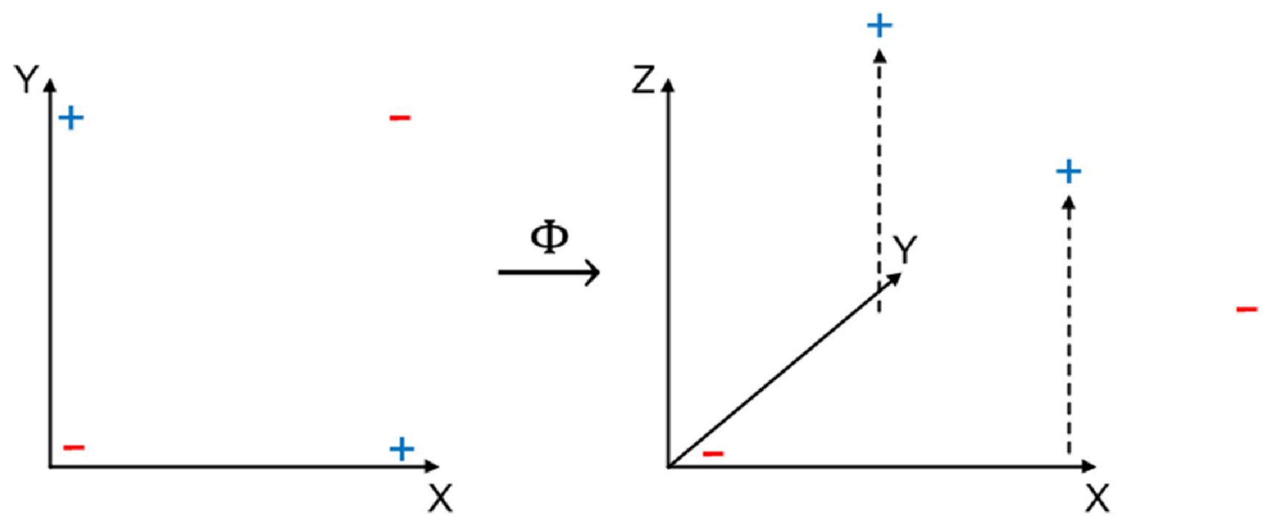
Verification (cont.)

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

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

Kernel Trick Intuition

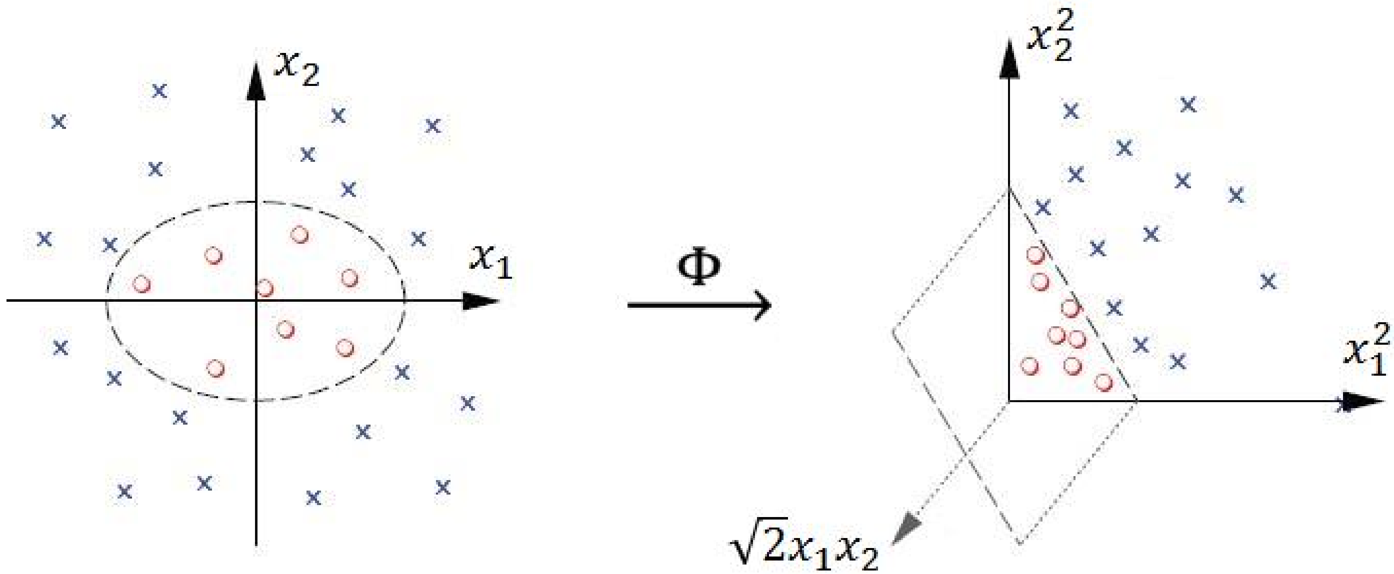
- SVM Solution for linearly inseparable problems, such as XOR
  - Kernel Trick: using a linear classifier to solve a non-linear problem.



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

$$\begin{cases} L(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \boldsymbol{x_i}.\boldsymbol{x_j} + \sum_i \alpha_i \\ h(\boldsymbol{x_{test}}) = sgn(\sum_i \alpha_i y_i \boldsymbol{x_i}.\boldsymbol{x_{test}} + b) \end{cases}$$

1

- SVM Transformation:

$$L(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \boldsymbol{\Phi}(\boldsymbol{x_i}).\boldsymbol{\Phi}(\boldsymbol{x_j}) + \sum_i \alpha_i$$

2

Kernel Trick – Formal (cont.)

- SVM Transformation:

$$L(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i}\sum_{j}\alpha_i\alpha_j y_i y_j \boldsymbol{\Phi}(\boldsymbol{x_i}).\boldsymbol{\Phi}(\boldsymbol{x_j}) + \sum_{i}\alpha_i \qquad \cancel{2}$$

- Kernel Trick (to avoid expensive data transformations)

$$L(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i}\sum_{j}\alpha_i\alpha_j y_i y_j \boxed{K(\boldsymbol{x_i}, \boldsymbol{x_j})} + \sum_{i}\alpha_i \qquad 3$$

Kernel Trick – Formal (cont.)

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

$$L(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i}\sum_{j}\alpha_i\alpha_j y_i y_j K(\boldsymbol{x_i}, \boldsymbol{x_j}) + \sum_{i}\alpha_i$$

- If $K(\boldsymbol{x_i}, \boldsymbol{x_j})$ is an inner product in some space, we are good!

Kernel Function Properties

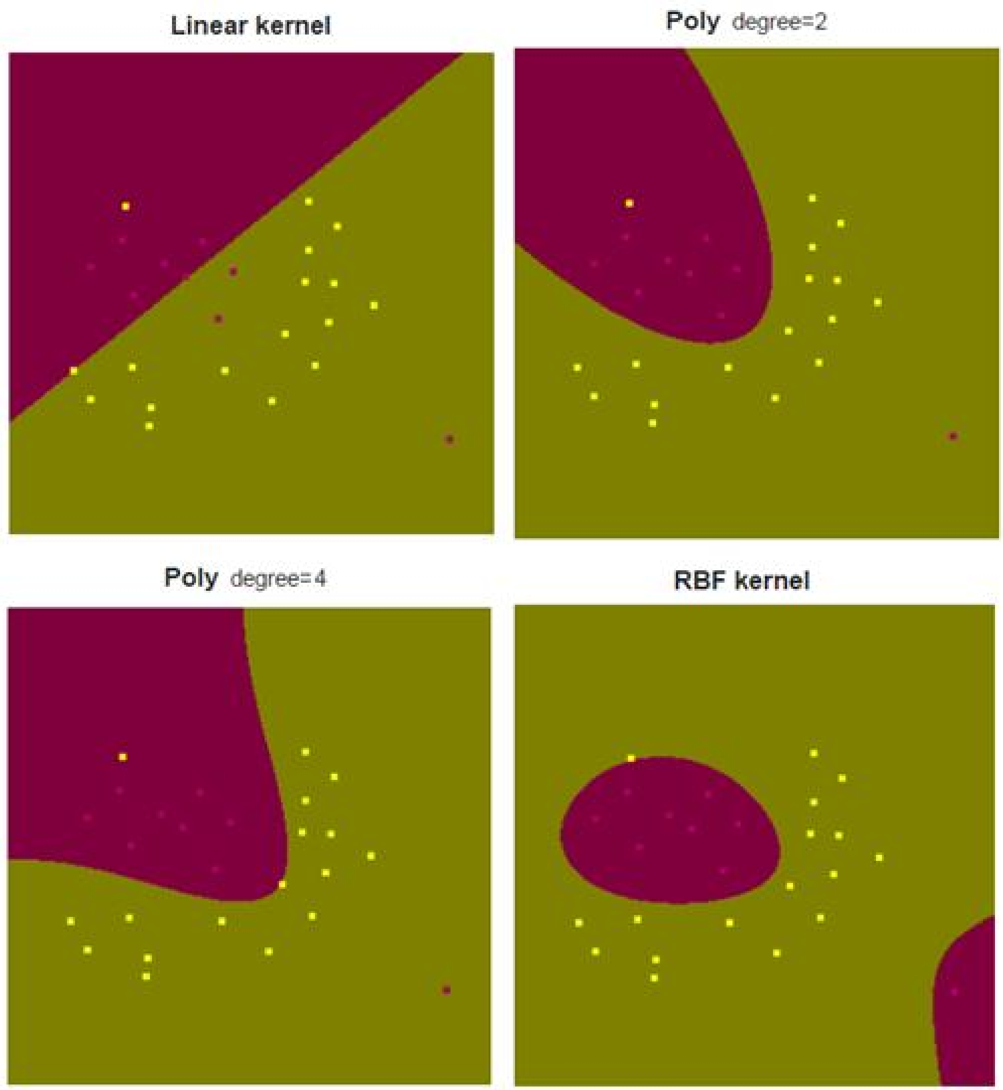
- Function $K(\boldsymbol{x}, \boldsymbol{x}')$ is a valid kernel if:
  - It computes an inner product in <u>some</u> space $\mathbb{Z}$.
    - We just need to know that space $\mathbb{Z}$ exists!
  - It is symmetric / commutative, i.e. $K(\boldsymbol{x}, \boldsymbol{x}') = K(\boldsymbol{x}', \boldsymbol{x})$.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

Kernel Types

- The most frequently used kernel types:
  - Linear: $(\boldsymbol{x_i}.\boldsymbol{x_j} + c)$
    - If $c = 0$, it is homogenous.
  - Polynomial: $(\alpha\boldsymbol{x_i}.\boldsymbol{x_j} + c)^{\mathrm{d}}$, subject to: $d > 1$
  - Gaussian RBF: $\exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}\right) = \exp\left(-\gamma\left\|\boldsymbol{x_i} - \boldsymbol{x_j}\right\|^2\right)$,

    subject to: $\gamma = \frac{1}{2\sigma^2}$

Kernel Types



Figures © Stackoverflow.com

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

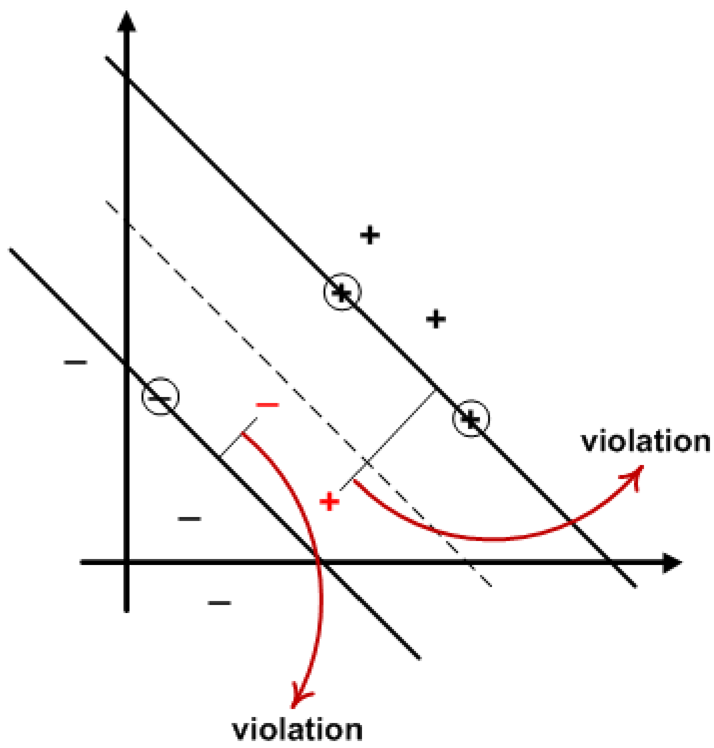
- So which one to choose? [1]
  - Occam's razor

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.



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

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

New Optimization – Primal

$$\begin{cases} minimize \, \frac{1}{2}||\boldsymbol{w}||^2 + C\sum_{i=1}^{m}\xi_i \\ subject\ to\colon\ y_i(\boldsymbol{w}.\boldsymbol{x} + b) \geq 1 - \xi_i \ \ \&\ \ \xi_i \geq 0 \\ \qquad\qquad (for\ i = 1,2,\ldots,m) \end{cases}$$

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

New Optimization – Primal (cont.)

$$\begin{cases} minimize \dfrac{1}{2}\left\|\boldsymbol{w}\right\|^2 + C\displaystyle\sum_{i=1}^{m}\xi_i \\ subject\ to\colon\ y_i(\boldsymbol{w}.\boldsymbol{x}+b) \geq 1-\xi_i\ \ \&\ \ \xi_i \geq 0 \\ \qquad\qquad (for\ i = 1,2,\dots,m) \end{cases}$$

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

small C | regularization parameter C | large C

underfitting | overfitting (hard margin)

Soft Margin SVM Decision Boundary

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

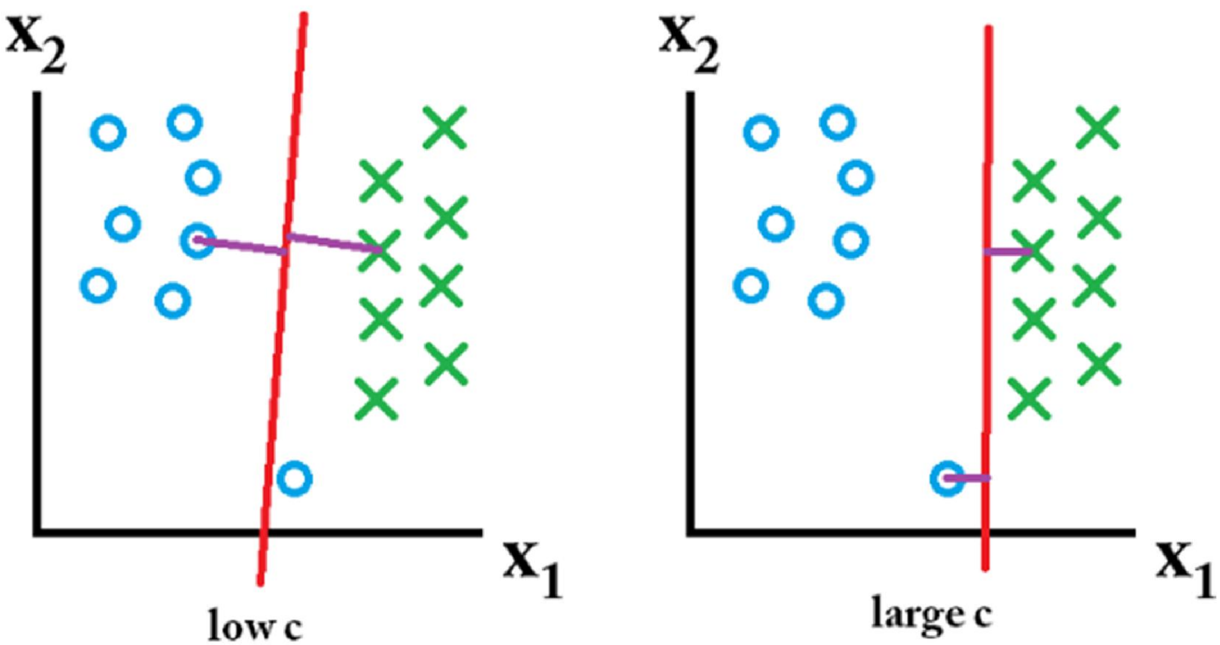


Figure © https://stats.stackexchange.com/

Lagrange Multipliers

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

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

Lagrange Multipliers (cont.)

$$L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}||\boldsymbol{w}||^2 - \sum_{i=1}^{m} \alpha_i \left[y_i(\boldsymbol{w}.\boldsymbol{x_i} + b) - 1\right] - \sum_{i=1}^{m} \xi_i(C - \alpha_i - \beta_i)]$$

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

$$\begin{cases} \dfrac{\partial L}{\partial \boldsymbol{w}} = w - \displaystyle\sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x_i} = 0 \\ \dfrac{\partial L}{\partial b} = -\displaystyle\sum_{i=1}^{m} \alpha_i y_i = 0 \\ \dfrac{\partial L}{\partial \xi_i} = C - \alpha_i - \beta_i = 0 \end{cases}$$

Dual Problem – Soft Margin SVM

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

$$\boxed{\begin{gathered} L(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \boldsymbol{x_i}.\boldsymbol{x_j} + \sum_{i=1}^{m}\alpha_i\,, \forall i \\ subject\ to\colon\ 0 \leq \alpha_i \leq C, \forall i \end{gathered}}$$

Support Vector (SV) Types

$$
\begin{cases}
\text{non} - \text{SV:}\ \alpha_i = 0 \ \rightarrow \ y_i(\boldsymbol{w}.\boldsymbol{x_i} + b) > 1 \\
\text{magin SV:}\ 0 < \alpha_i < C \ \rightarrow y_i(\boldsymbol{w}.\boldsymbol{x_i} + b) = 1 \\
\text{non} - \text{margin SV:}\ \alpha_i = C \ \rightarrow y_i(\boldsymbol{w}.\boldsymbol{x_i} + b) < 1
\end{cases}
$$

- 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.