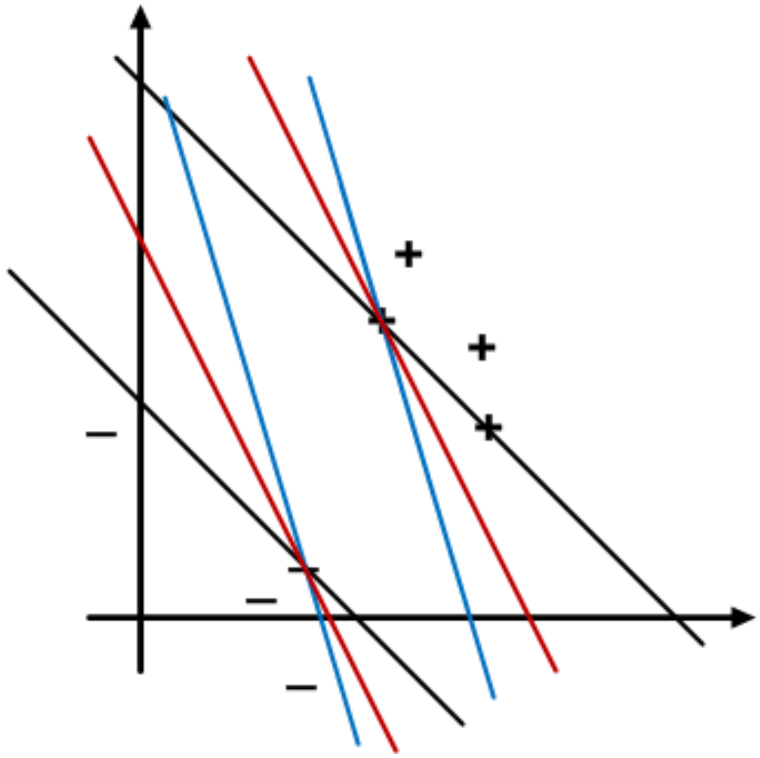
Support Vector Machines (SVMs)

- Vladimir Vapnik

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

  - Widest street approach
  - Maximum margin classifier

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



Support Vector Machines (cont.)

- Vladimir Vapnik

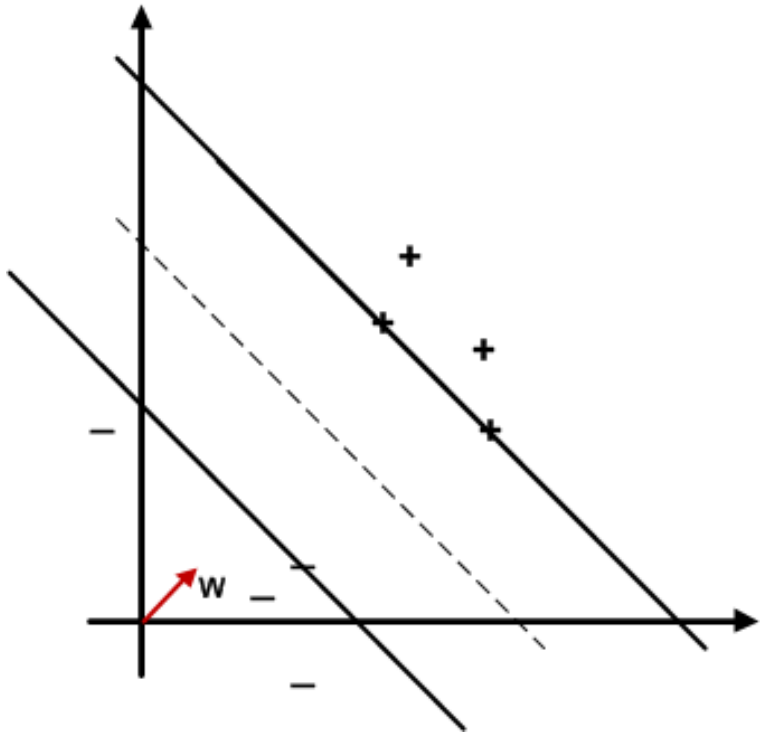
$$h: \mathrm{X} \rightarrow \{-1, +1\}$$

- 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?)

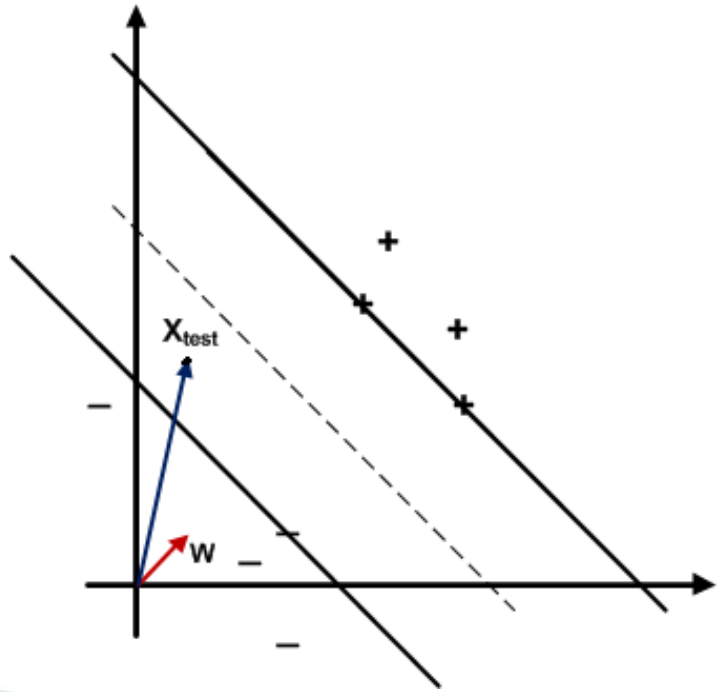
$$^{**}\ \boldsymbol{w}^T\boldsymbol{x} + b = 0$$



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. $b = -c$

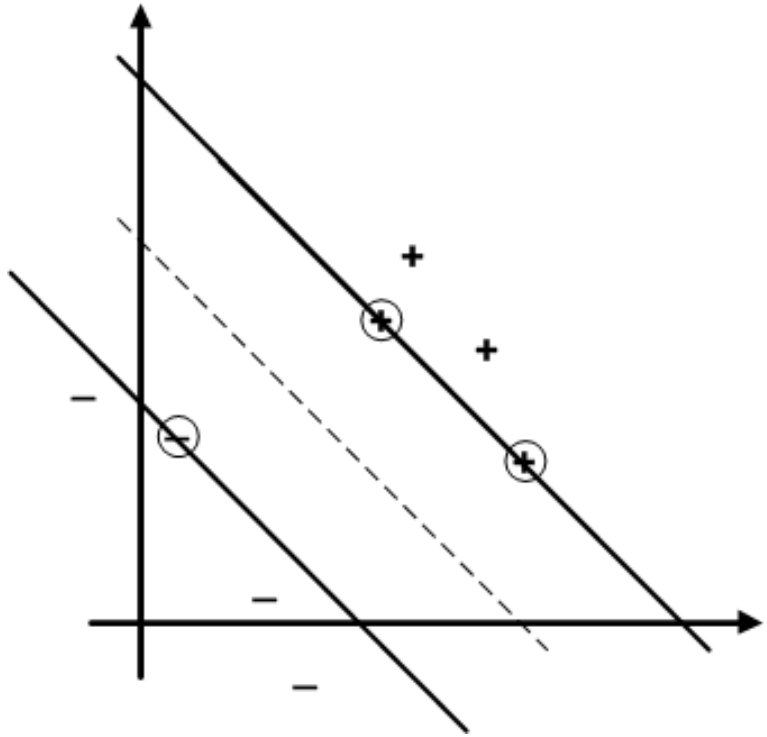


Constraints

- Let assume the following constraints:

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

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



Constraints (cont.)

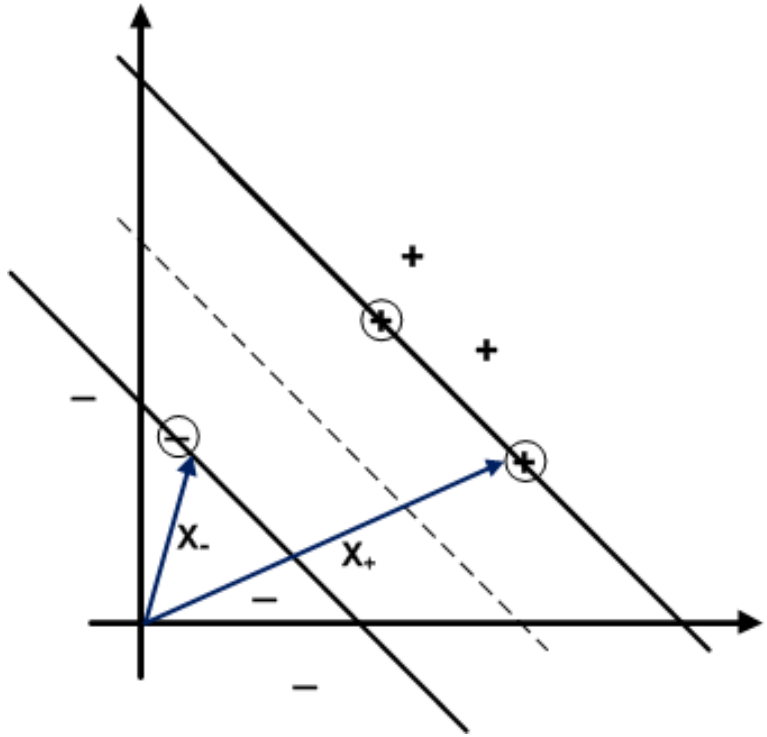
- Let assume the following constraints:

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

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

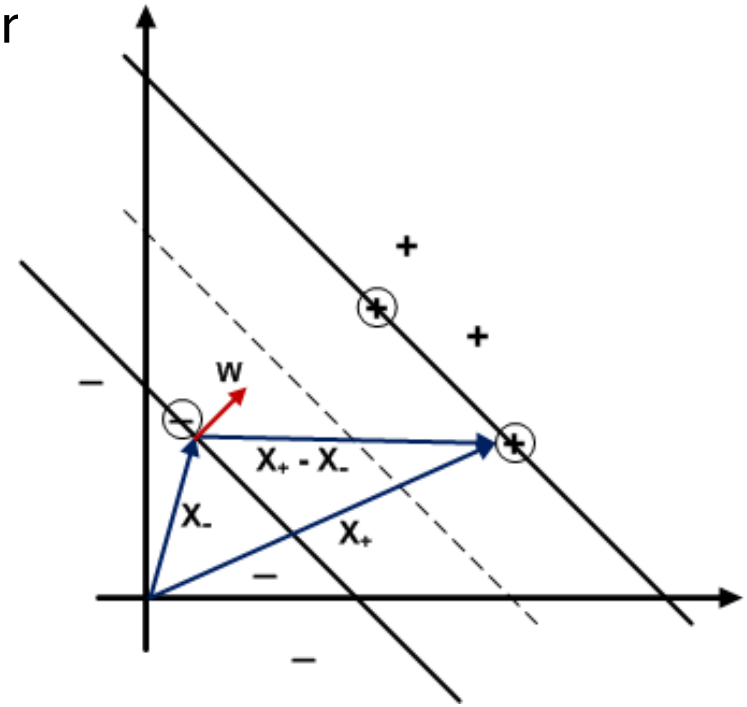
$$\boxed{y_i(\boldsymbol{w}.\boldsymbol{x_i} + b) - 1 = 0}$$

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



Width of the Street

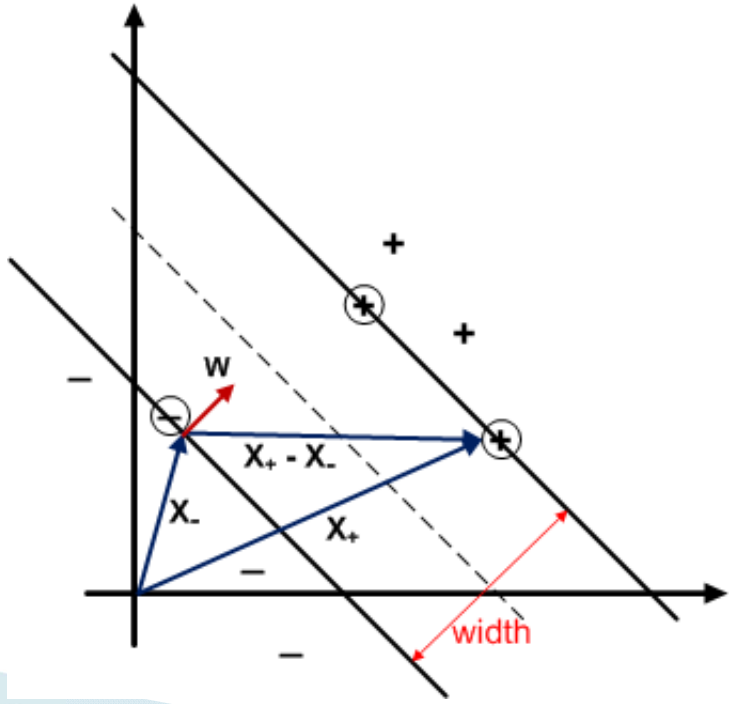
- Recall that vector $\boldsymbol{w}$ is perpendicular to the street.
- If we project vector $(\boldsymbol{x}_+ - \boldsymbol{x}_-)$ to vector $\boldsymbol{w}$, the width of the street is obtained.
  - Vector $\boldsymbol{w}$ has to be a unit vector



Width of the Street (cont.)

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

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



Constrained Optimization Problem

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

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

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

$$\Downarrow$$

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

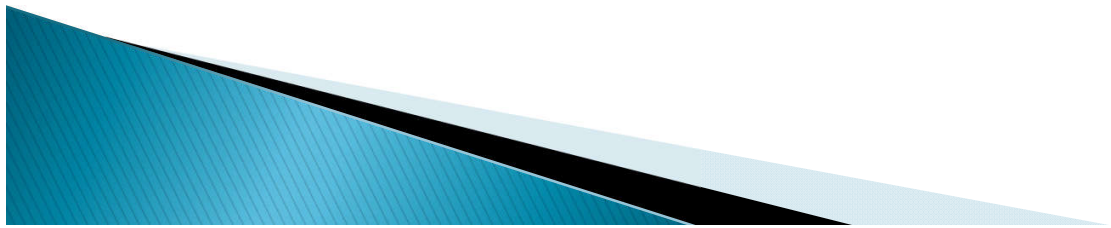
Lagrange Multipliers

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

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

$$subject\ to\colon\ \alpha_i \geq 0, \forall \mathrm{i}$$

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



Dual Problem

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

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

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

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

Dual Problem (cont.)

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

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

($\sum_i \alpha_i y_i$ → 0)

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

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

  - 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

$$L(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i \text{ is a SV}}\sum_{j \text{ is a SV}} \alpha_i \alpha_j y_i y_j \boldsymbol{x_i}.\boldsymbol{x_j} + \sum_{i \text{ is a SV}} \alpha_i$$

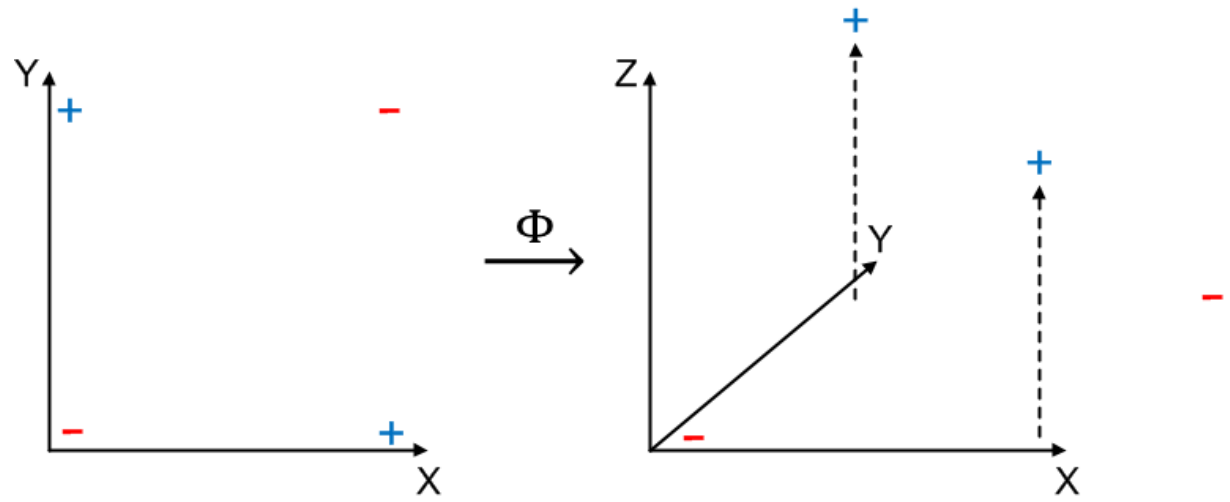
$$subject\ to:\ \alpha_i > 0, \forall i$$

- Classification Rule

$$\sum_{i \text{ is a SV}} \alpha_i y_i \boldsymbol{x_i}.\boldsymbol{x_{test}} + b \geq 0 \Rightarrow Class\ is +$$

Kernel Trick Intuition

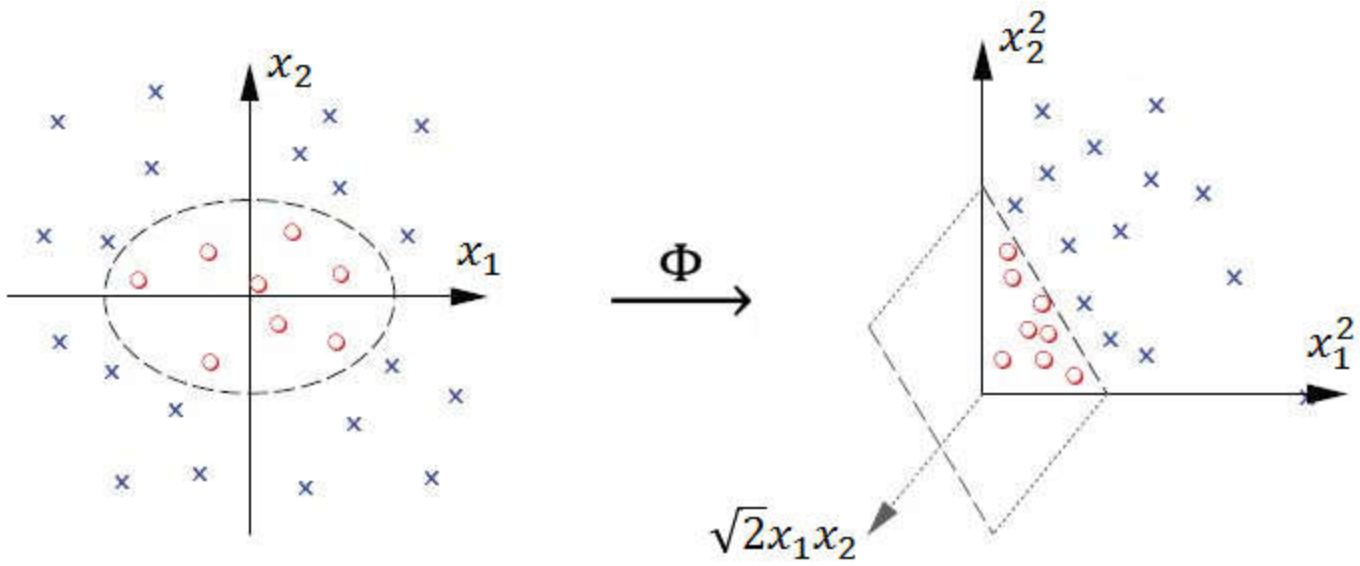
- SVM Solution for linearly inseparable problems, such as XOR
  - Kernel Trick: using a linear classifier to solve a non-linear problem.



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

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

- SVM Transformation:

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

Kernel Trick – Formal (cont.)

- SVM Transformation:

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

2

- Kernel Trick (to avoid expensive data transformations)

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

3

Kernel Trick – Formal (cont.)

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

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

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

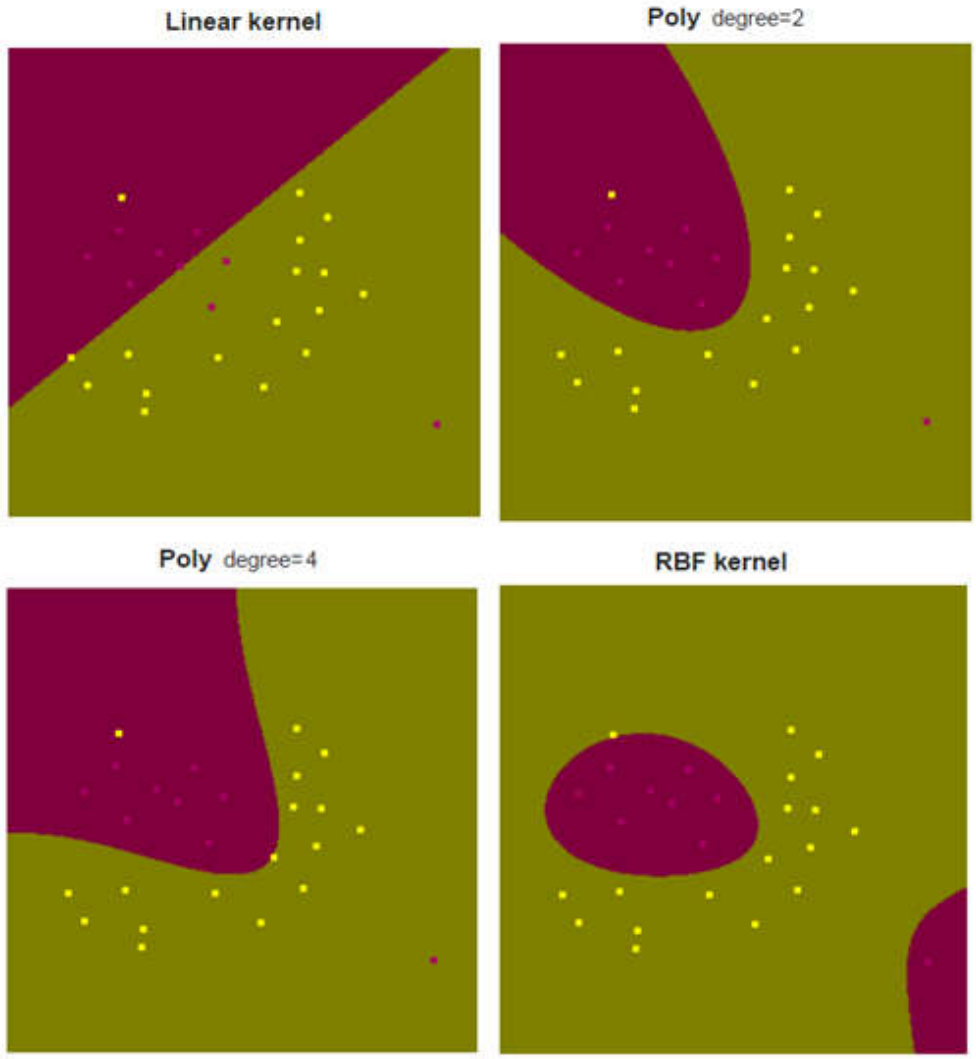
Kernel Function Properties

- Function $K(\boldsymbol{x}, \boldsymbol{x}')$ is a valid kernel if:
  - It computes an inner product in some 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: $\exp\left(-\frac{\left|\left|x_i - x_j\right|\right|^2}{2\sigma^2}\right) = \exp\left(-\gamma \left|\left|\boldsymbol{x_i} - \boldsymbol{x_j}\right|\right|^2\right)$, subject to: $\gamma = \frac{1}{2\sigma^2}$

Kernel Types

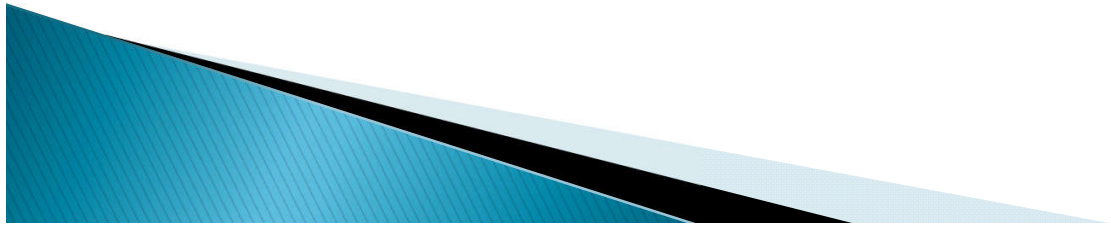


Figures © Stackoverflow.com

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]
  - Ockham'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

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.