# COSC 522 – Machine Learning

# Lecture 14 – Kernel Methods: Support Vector Machine

Hairong Qi, Gonzalez Family Professor
Electrical Engineering and Computer Science
University of Tennessee, Knoxville
https://www.eecs.utk.edu/people/hairong-qi/
Email: hqi@utk.edu

Course Website: http://web.eecs.utk.edu/~hqi/cosc522/

# Roadmap

- Supervised learning
  - Classification
    - Maximum Posterior Probability (MPP): For a given x, if $P(w_1|x) > P(w_2|x)$, then x belongs to class 1, otherwise 2.
      - Parametric Learning
        - Three cases
        - Estimate Gaussian parameters using MLE
      - Nonparametric Learning
        - Parzon window (fixed window size)
        - K-Nearest Neighbor (variable window size)
    - Neural Network
    - SVM
  - Regression (linear regression with nonlinear basis functions)
    - Neural Network
    - SVM
- Unsupervised learning
  - Non-probabilistic approaches
    - kmeans, wta
  - Hierarchical approaches
    - Agglomerative clustering
  - Neural Network
- Supporting preprocessing techniques
  - Dimensionality Reduction
    - Supervised linear (FLD)
    - Unsupervised linear (PCA)
    - Unsupervised nonlinear (t-SNE)
- Supporting postprocessing techniques
  - Classifier Fusion
  - Performance Evaluation
- Optimization techniques
  - Gradient Descent (GD)

# References

- Christopher J.C. Burges, “A tutorial on support vector machines for pattern recognition,” *Data Mining and Knowledge Discovery*, 2, 121-167, 1998
- M. Mohri, A. Rostamizadeh, A. Talwalkar, *Foundations of Machine Learning*, 2nd Edition, The MIT Press, 2018.
- Cortes, Corinna; Vapnik, Vladimir (1995-09-01). "Support-vector networks". *Machine Learning*. **20** (3): 273–297.

# Questions

- What does generalization and capacity mean?
- What is VC dimension?
- What is the principled method?
- What is the VC dimension for perceptron?
- What are support vectors?
- What is the cost function for SVM?
- What is the optimization method used?
- How to handle non-separable cases using SVM?
- What is kernel trick?

# A bit about Vapnik

- Started SVM study in late 70s
- Fully developed in late 90s
- While at AT&T lab

http://en.wikipedia.org/wiki/Vladimir_Vapnik

# Generalization and capacity

- For a given learning task, with a given finite amount of training data, the best generalization performance will be achieved if the right balance is struck between the accuracy attained on that particular training set, and the “capacity” of the machine
- Capacity – the ability of the machine to learn any training set without error
  - Too much capacity - overfitting

# Bounds on the balance

- Under what circumstances, and how quickly, the mean of some empirical quantity converges uniformly, as the number of data point increases, to the true mean
- True mean error (or actual risk)

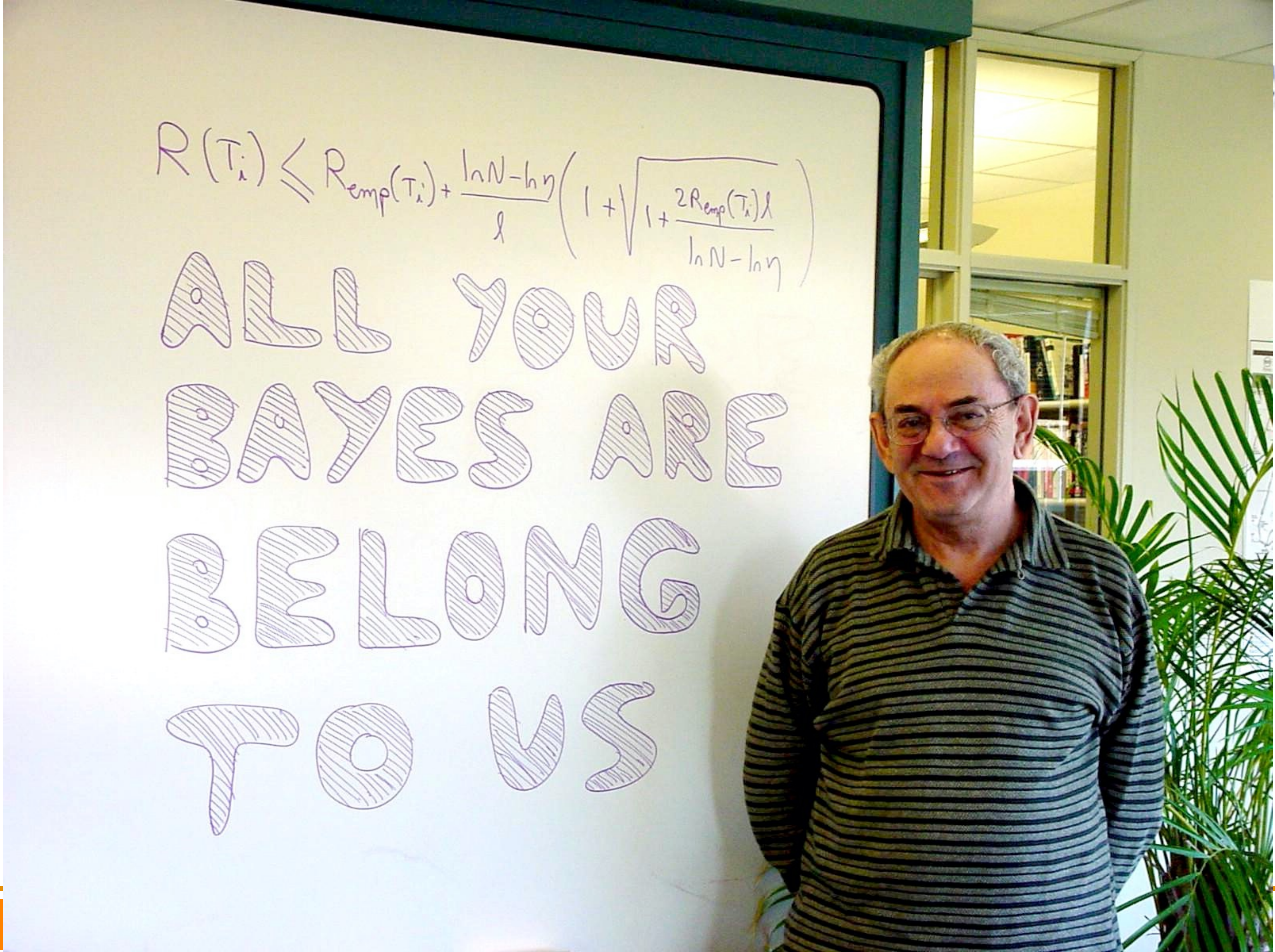
$$R(\alpha) = \int \tfrac{1}{2}\left|y - f(\mathbf{x},\alpha)\right| p(\mathbf{x},y)\,d\mathbf{x}dy$$

- One of the bounds

$$R(\alpha) \le R_{emp}(\alpha) + \sqrt{\left(\frac{h(\log(2l/h)+1) - \log(\eta/4)}{l}\right)} \qquad R_{emp}(\alpha) = \frac{1}{2l}\sum_{i=1}^{l}\left|y_i - f(\mathbf{x}_i,\alpha)\right|$$

f($\mathbf{x}$,$\alpha$): a machine that defines a set of mappings, $\mathbf{x} \rightarrow f(\mathbf{x},\alpha)$
$\alpha$: parameter or model learned
h: VC dimension that measures the capacity. non-negative integer
$R_{emp}$: empirical risk
$\eta$: 1-$\eta$ is confidence about the loss, $\eta$ is between [0, 1]
*l*: number of observations, $y_i$: label, {+1, -1}, $\mathbf{x}_i$ is n-D vector

Principled method: choose a learning machine that minimizes the RHS with a sufficiently small $\eta$


$R(T_i) \leq R_{emp}(T_i) + \frac{\ln N - \ln \eta}{l}\left(1 + \sqrt{1 + \frac{2R_{emp}(T_i)\,l}{\ln N - \ln \eta}}\right)$
ALL YOUR
BAYES ARE
BELONG
TO US

# VC dimension

- For a given set of $l$ points, there can be $2^l$ ways to label them. For each labeling, if a member of the set $\{f(\alpha)\}$ can be found that correctly classifies them, we say that set of points is shattered by that set of functions.
- VC dimension of that set of functions $\{f(\alpha)\}$ is defined as the maximum number of training points that can be shattered by $\{f(\alpha)\}$
- We should minimize h in order to minimize the bound

# Example (f($\alpha$) is perceptron)

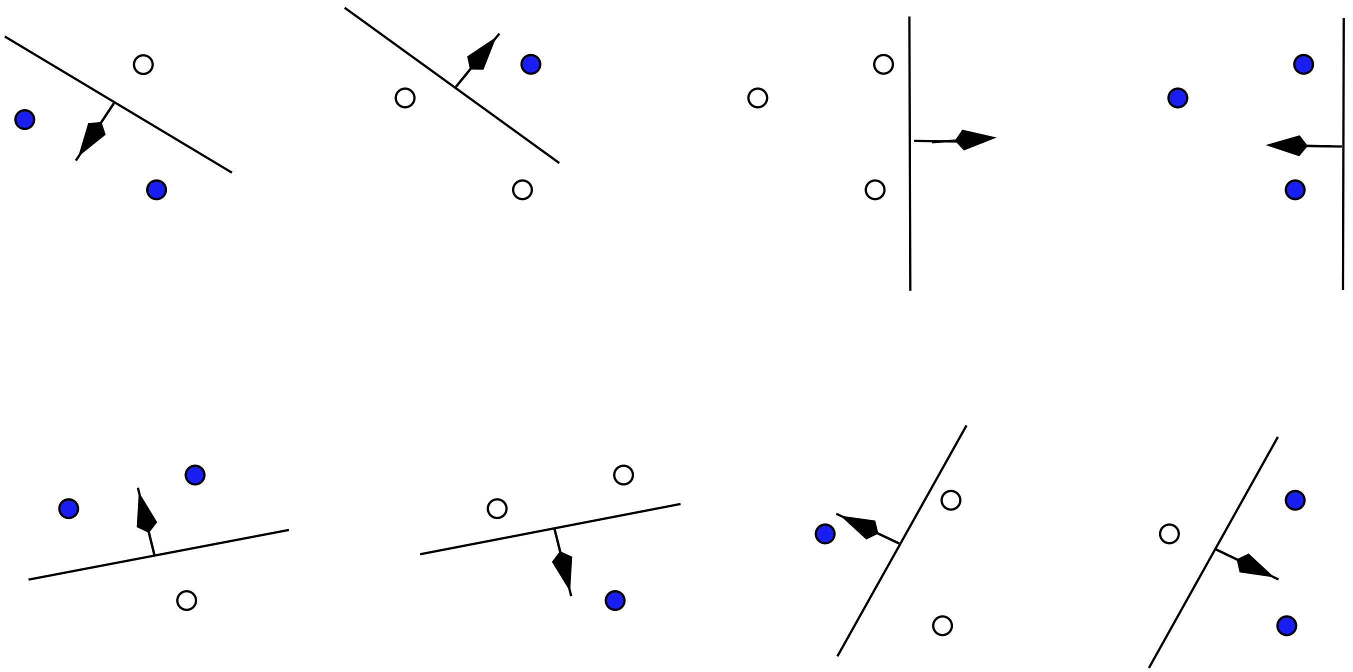
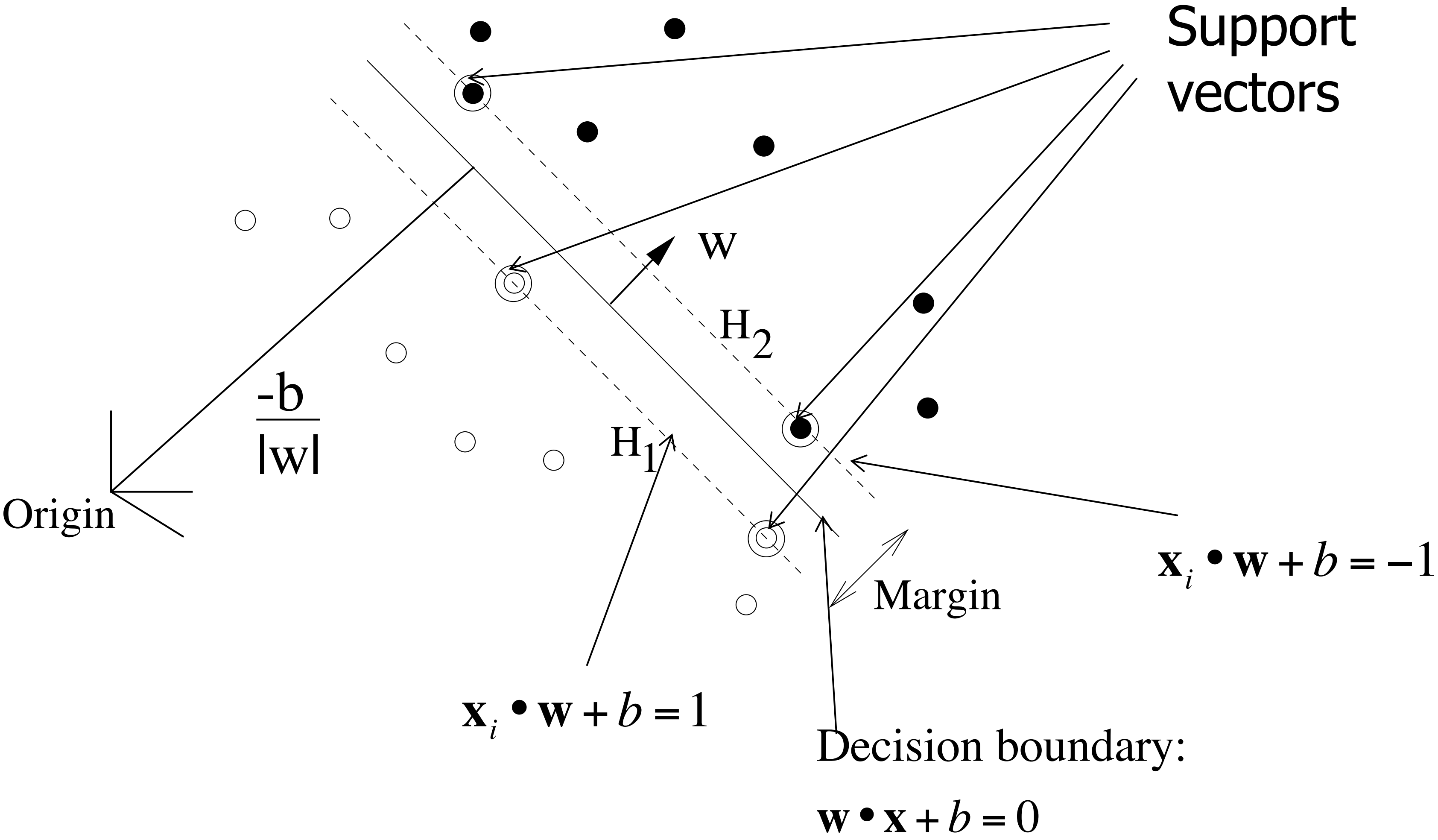


*Figure 1*. Three points in $\mathbf{R}^2$, shattered by oriented lines.

# Questions

- What does generalization and capacity mean?
- What is VC dimension?
- What is the principled method?
- What is the VC dimension for perceptron?
- What are support vectors?
- What is the cost function for SVM?
- What is the optimization method used?
- How to handle non-separable cases using SVM?
- What is kernel trick?

# Linear SVM – The separable case



Support vectors

$\mathbf{x}_i \bullet \mathbf{w} + b = -1$

$\mathbf{x}_i \bullet \mathbf{w} + b = 1$

Decision boundary:

$\mathbf{w} \bullet \mathbf{x} + b = 0$

$$\begin{cases} \mathbf{x}_i \bullet \mathbf{w} + b \geq 1 \quad \text{for} \quad y_i = +1 \\ \mathbf{x}_i \bullet \mathbf{w} + b \leq -1 \quad \text{for} \quad y_i = -1 \end{cases}$$

Minimizing $\|\mathbf{w}\|^2$

s.j. $y_i\left(\mathbf{x}_i \bullet \mathbf{w} + b\right) - 1 \geq 0$

Minimize $L_P = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{l} \alpha_i y_i \left(\mathbf{x}_i \bullet \mathbf{w} + b\right) + \sum_{i=1}^{l} \alpha_i$

$$\frac{\partial L_P}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i, \qquad \frac{\partial L_P}{\partial b} = 0 \Rightarrow \sum_i \alpha_i y_i = 0$$

Maximize $L_D = -\frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \mathbf{x}_j + \sum_i \alpha_i$

# Questions

- What does generalization and capacity mean?
- What is VC dimension?
- What is the principled method?
- What is the VC dimension for perceptron?
- What are support vectors?
- What is the cost function for SVM?
- What is the optimization method used?
- How to handle non-separable cases using SVM?
- What is kernel trick?

# Non-separable cases

- SVM with soft margin
- Kernel trick

# Non-separable case – Soft margin

$$\begin{cases} \mathbf{x}_i \bullet \mathbf{w} + b \geq 1 - \xi_i & \text{for} \quad y_i = +1 \\ \mathbf{x}_i \bullet \mathbf{w} + b \leq -1 + \xi_i & \text{for} \quad y_i = -1 \end{cases} \qquad \text{for } \xi_i \geq 0$$

$$\text{Minimizing } \|\mathbf{w}\|^2$$

$$\text{s.j.} \quad y_i\left(\mathbf{x}_i \bullet \mathbf{w} + b\right) - 1 + \xi_i \geq 0$$

$$\text{Minimize } L_P = \tfrac{1}{2}\|\mathbf{w}\|^2 - C\left(\sum_i \xi_i\right)^k$$

$$\text{Maximize } L_D = -\tfrac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \mathbf{x}_j + \sum_i \alpha_i$$

$$\text{s.j.} \quad 0 \leq \alpha_i \leq C, \quad \sum_i \alpha_i y_i = 0$$

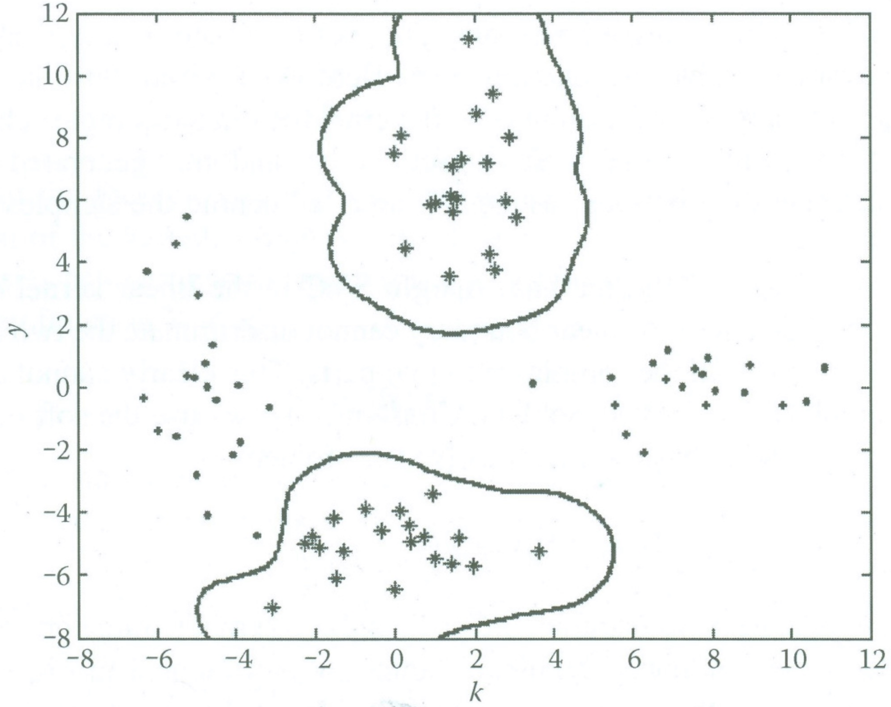
# Non-separable cases – The kernel trick

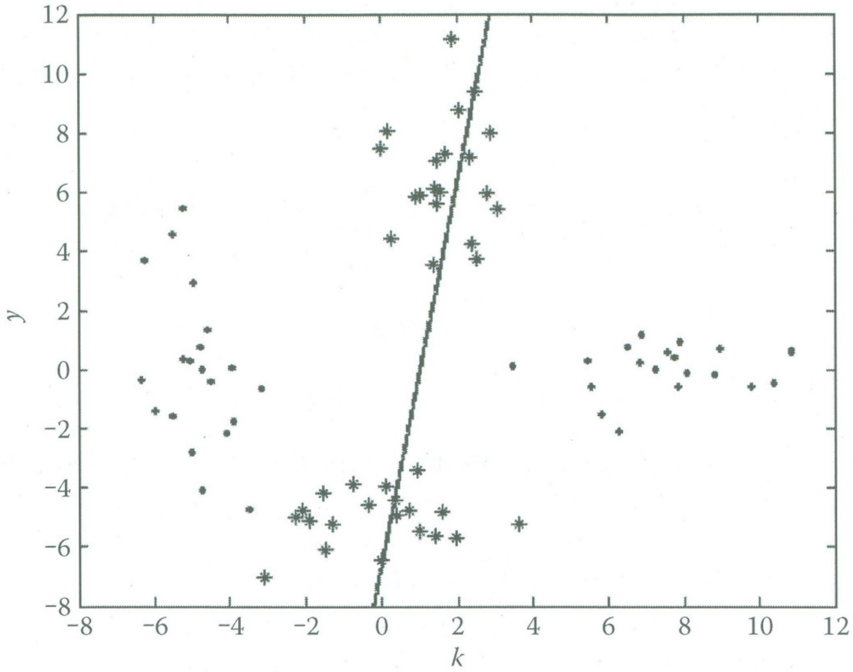
- If there were a “kernel function”, K, s.t.

$$K\left(\mathbf{x}_i,\mathbf{x}_j\right)=\Phi\left(\mathbf{x}_i\right)\cdot\Phi\left(\mathbf{x}_j\right)=e^{-\left\|\mathbf{x}_i-\mathbf{x}_j\right\|^2 / 2\sigma^2}$$

Gaussian Radial Basis Function (RBF)

# Comparison - XOR





# Limitation

- Need to choose parameters

# A toy example