COSC 522 – Machine Learning

Kernel Methods and 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

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

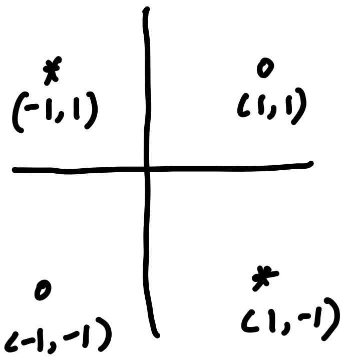
- What is kernel trick?

- 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?

PART I: KERNEL METHODS



$$K(\vec{x}, \vec{x}') = (x_1 x_1' + x_2 x_2' + c)^2$$

$$= (x_1 x_1')^2 + (x_2 x_2')^2 + c^2 + 2 x_1 x_1' x_2 x_2' + 2 x_1 x_1' c + 2 x_2 x_2' c$$

$$= \begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}\, x_1 x_2 \\ \sqrt{2c}\, x_1 \\ \sqrt{2c}\, x_2 \\ c \end{bmatrix} \begin{bmatrix} x_1'^2 \\ x_2'^2 \\ \sqrt{2}\, x_1' x_2' \\ \sqrt{2c}\, x_1' \\ \sqrt{2c}\, x_2' \\ c \end{bmatrix}$$

so $\phi(\vec{x}) =$

$\phi(\vec{x}') =$

That is, for example

$(-1, 1)$ now becomes
$x_1 \quad x_2$

$$(1, 1, -\sqrt{2}, -\sqrt{2}, \sqrt{2}, 1)$$

assuming $c = 1$

Kernel methods attempt to map the input data from a low-dimensional space to a higher-dimensional space where the mapped data is linearly separable.

A Toy Example: Explicit Mapping

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$$

The kernel trick

AICIP RESEARCH

- Kernel trick maps the data from low-dimensional data space to a high-dimensional space without explicit mapping
- It computes the inner product of data pairs in the training set and uses it as inputs to the kernel function to implicitly reflect relationships in the higher-dimensional space.
- A popular kernel function:

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

Gaussian Radial Basis Function (RBF)

PART II: SUPPORT VECTOR MACHINES (SVM)

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

AICIP RESEARCH

- 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

AICIP RESEARCH

- 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)

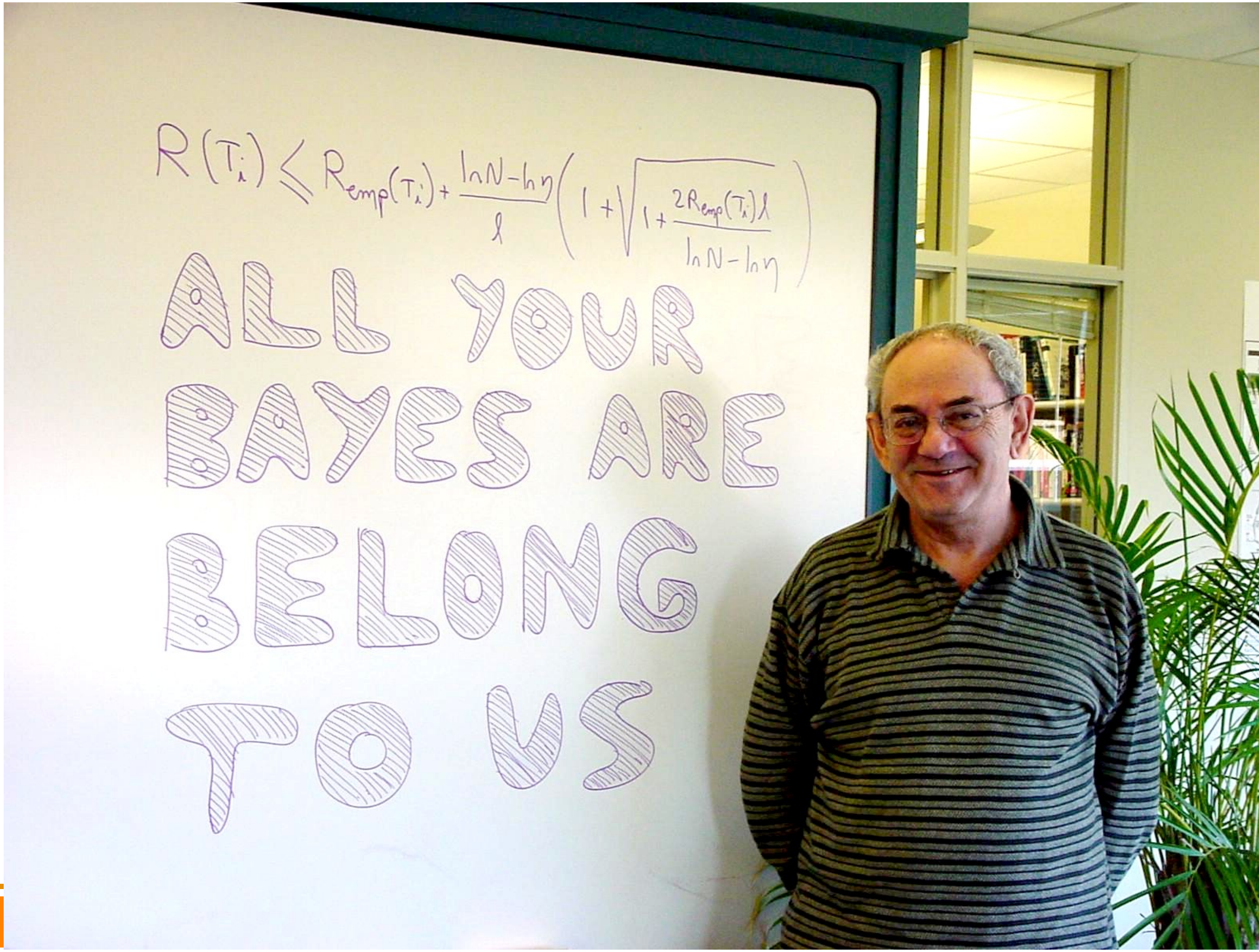
$$R(\alpha) = \int \tfrac{1}{2}\left|y - f(\mathbf{x},\alpha)\right| p(\mathbf{x},y)\, d\mathbf{x}dy$$

- 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($\mathbf{x}$,α): a machine that defines a set of mappings, $\mathbf{x} \rightarrow f(\mathbf{x},\alpha)$
α: parameter or model learned
h: VC dimension that measures the capacity. non-negative integer
R_{emp}: empirical risk
η: 1-η is confidence about the loss, η 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


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

VC dimension

AICIP RESEARCH

- 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(α) is perceptron)

AICIP RESEARCH

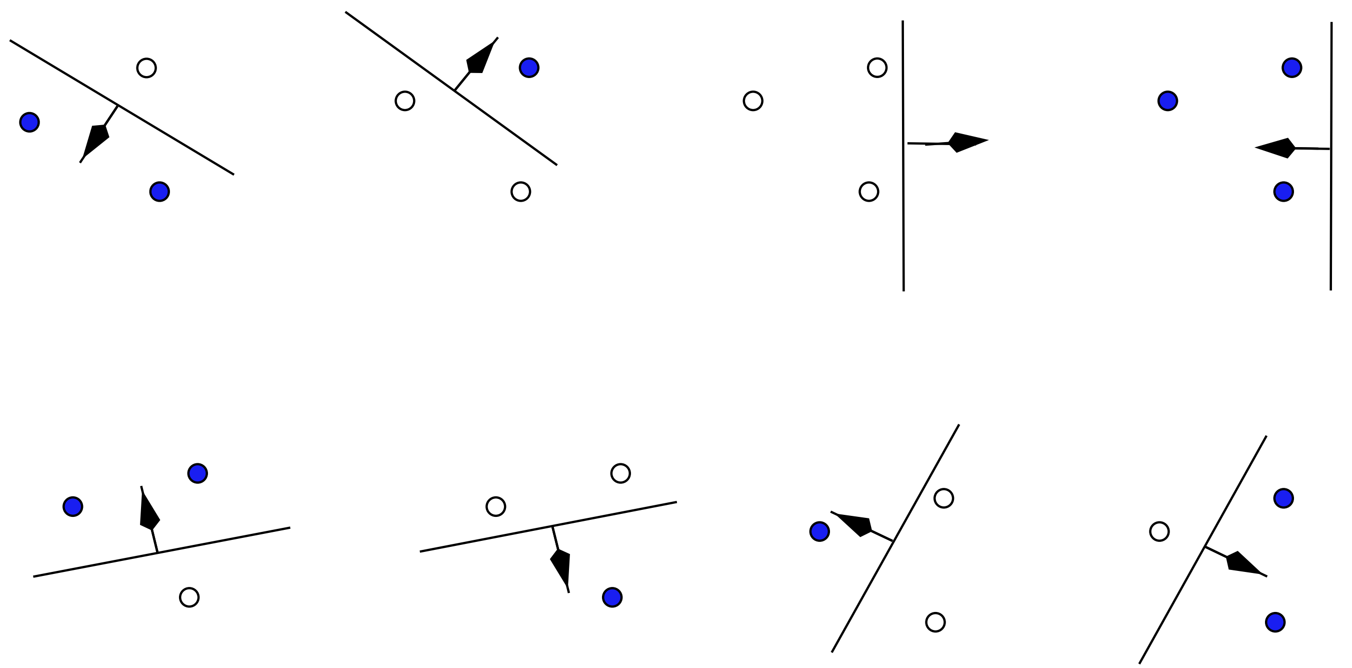
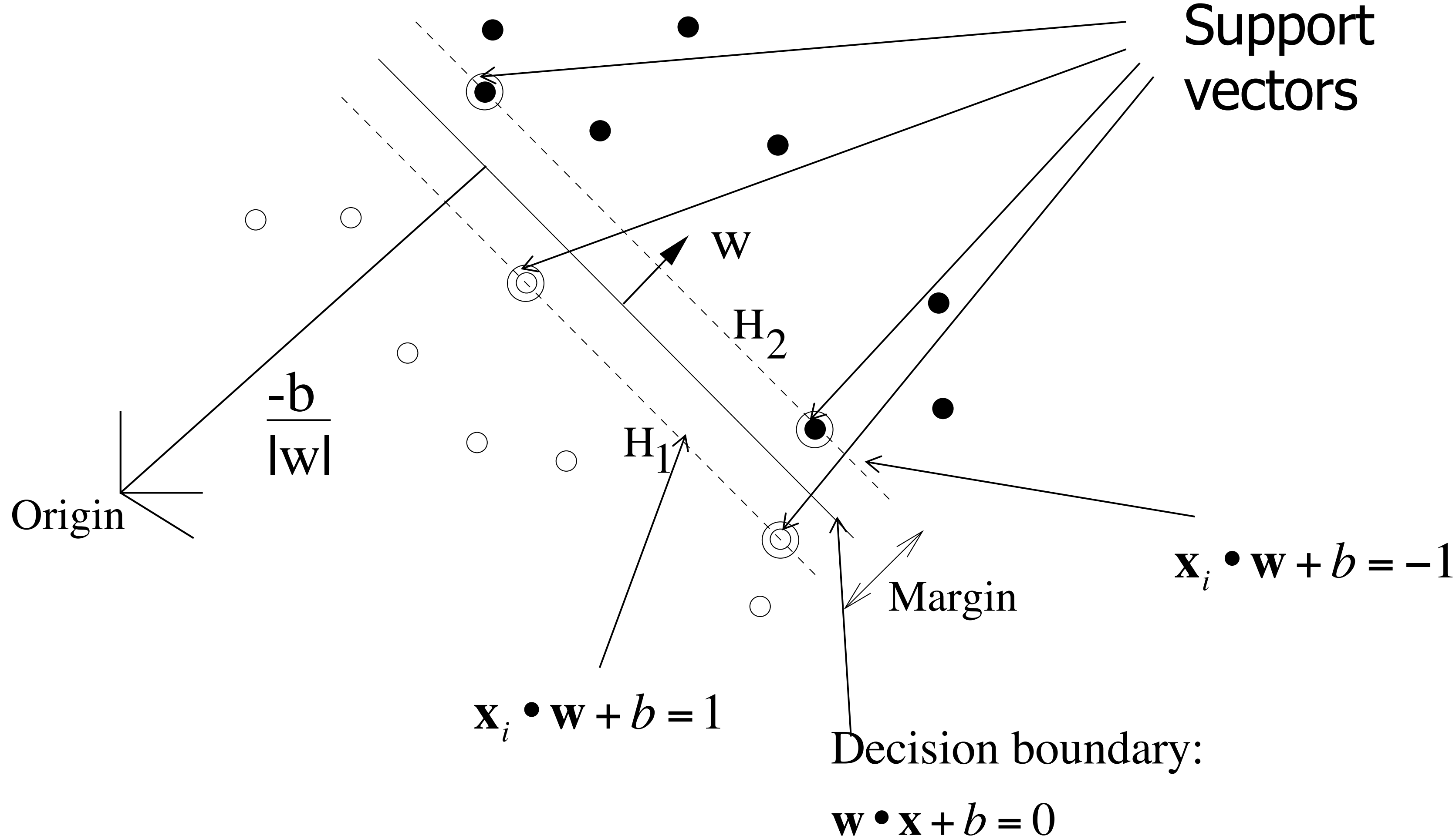


Figure 1. Three points in $\mathbf{R}^2$, shattered by oriented lines.

AICIP RESEARCH

Linear SVM – The separable case



AICIP RESEARCH

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

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

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

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

Non-separable cases

- SVM with soft margin
- Kernel trick

Non-separable case – Soft margin

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

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

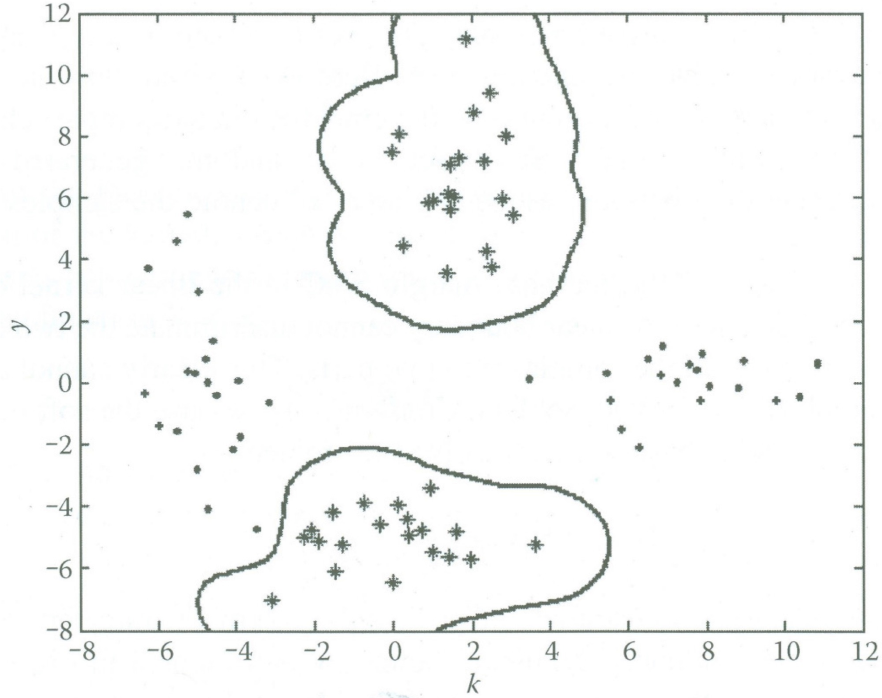
$$\text{s.j.} \quad y_i\left(\mathbf{x}_i \bullet \mathbf{w} + b\right) - 1 + \xi_i \ge 0$$

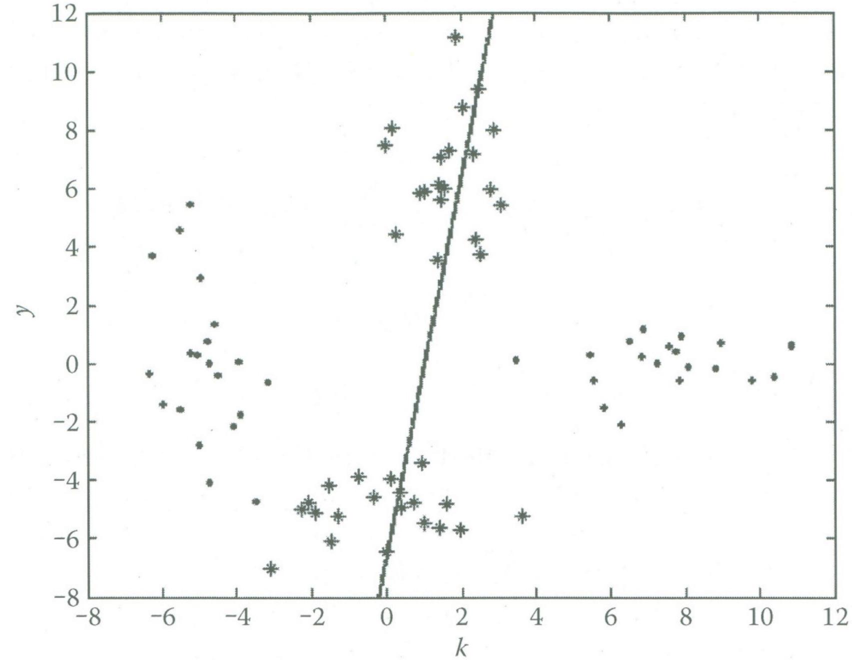
$$\text{Minimize } L_P = \tfrac{1}{2}\|\mathbf{w}\|^2 - C\left(\sum_i \xi_i\right)^k$$

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

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

Comparison - XOR





Limitation

- Need to choose parameters
  - Grid search