Support Vector Machine Learning

CS478 Machine Learning

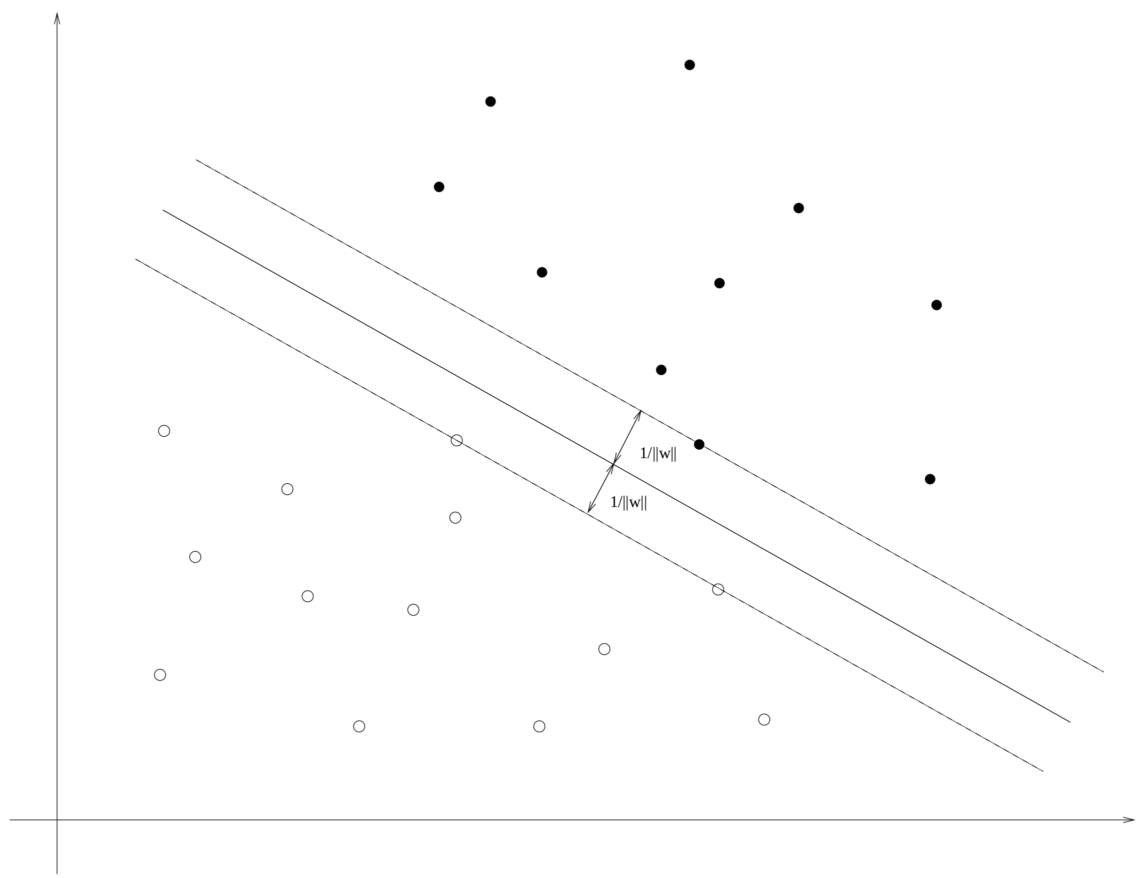
Alin Dobra

May 2, 2000

Overview

- Optimal Margin Classifier Algorithm
- Kernels
- Soft Margin Classifier
- Optimization problem
- Applications and practical results

Optimal Margin Classifier



Optimal Margin Classifier Algorithm

- Choose $y = 1$ for *positive* labels and $y = -1$ for *negative* labels

$$y(\mathbf{w}\mathbf{x} + b) \geq 1$$

- Problem: minimize

$$\Gamma(\mathbf{w}, b) = \frac{1}{2}\mathbf{w}^2, \quad y_i(\mathbf{w}\mathbf{x}_i + b) - 1 \geq 0$$

$$L = \frac{1}{2}\mathbf{w}^2 - \sum_{i=1}^{l} \alpha_i[y_i(\mathbf{w}\mathbf{x}_i + b) - 1], \quad \frac{\partial L}{\partial \alpha_i} = 0, \quad \alpha_i \geq 0$$

- Wolfe dual formulation: maximize L as a function of α_i with the constrains:

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i, \quad \frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^{l} \alpha_i y_i = 0$$

Optimal Margin Classifier Algorithm (cont.)

- Transformed problem: maximize

$$L = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \mathbf{x}_j, \quad \sum_{i=1}^{l} \alpha_i y_i = 0, \quad \alpha_i \geq 0$$

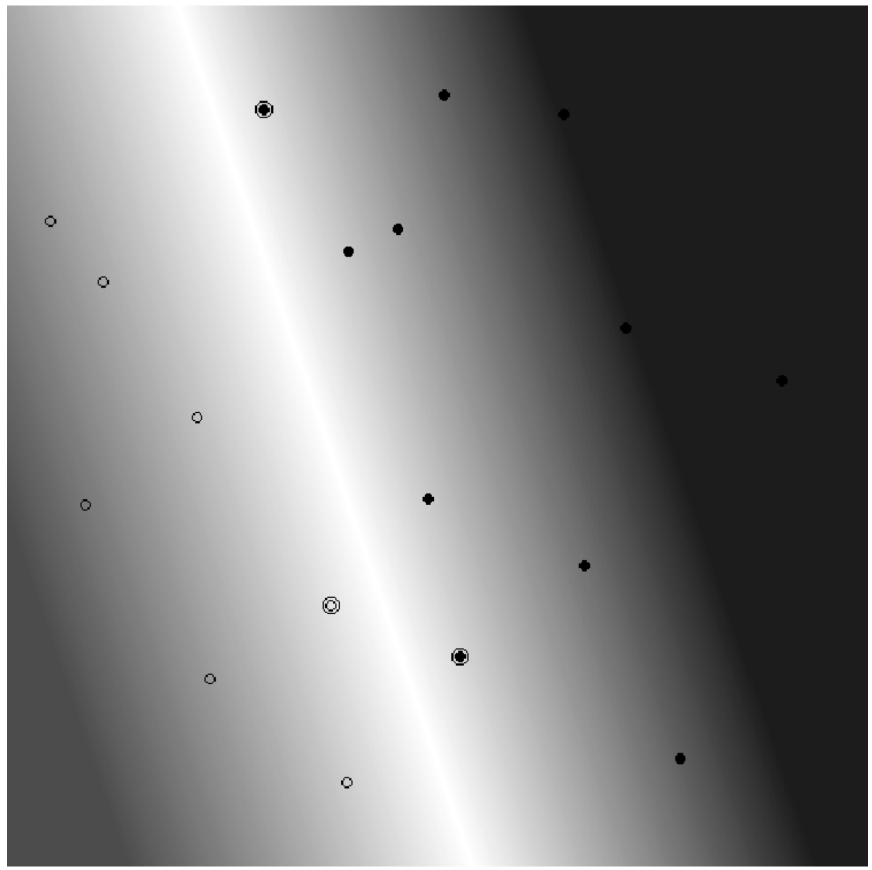
- Karush-Kuhn-Tucker conditions at extremum:

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

- Separating surface:

$$\sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i \mathbf{x} + b = 0$$

Optimal Margin Classifier Algorithm. Example



Kernels

- **Idea:**
  - use a transformation $\Phi(\mathbf{x})$ from the input space to a higher dimensional space
  - find the separating hyperplane
  - make the inverse transformation

  Eg:

$$\Phi(\mathbf{x}) : \mathbb{R}^2 \to \mathbb{R}^3, \quad \Phi(\mathbf{x}) = \{x_1^2, \sqrt{2}x_1x_2, x_2^2\}$$

- **Kernel**: dot product in a Banach space

$$K(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x})\Phi(\mathbf{x}')$$

- **Mercer's Theorem:** $K(\mathbf{x}, \mathbf{x}')$ is a dot product in a Banach space if

$$\int_{\mathcal{X}\times\mathcal{X}} K(\mathbf{x}, \mathbf{x}')f(\mathbf{x})f(\mathbf{x}')d\mathbf{x}d\mathbf{x}' \geq 0, \ \forall f \in L_2(\mathcal{X})$$

Examples of kernels

- polynomial kernels:

$$K(\mathbf{x}, \mathbf{x}') = (\mathbf{x}\mathbf{x}' + c)^p$$

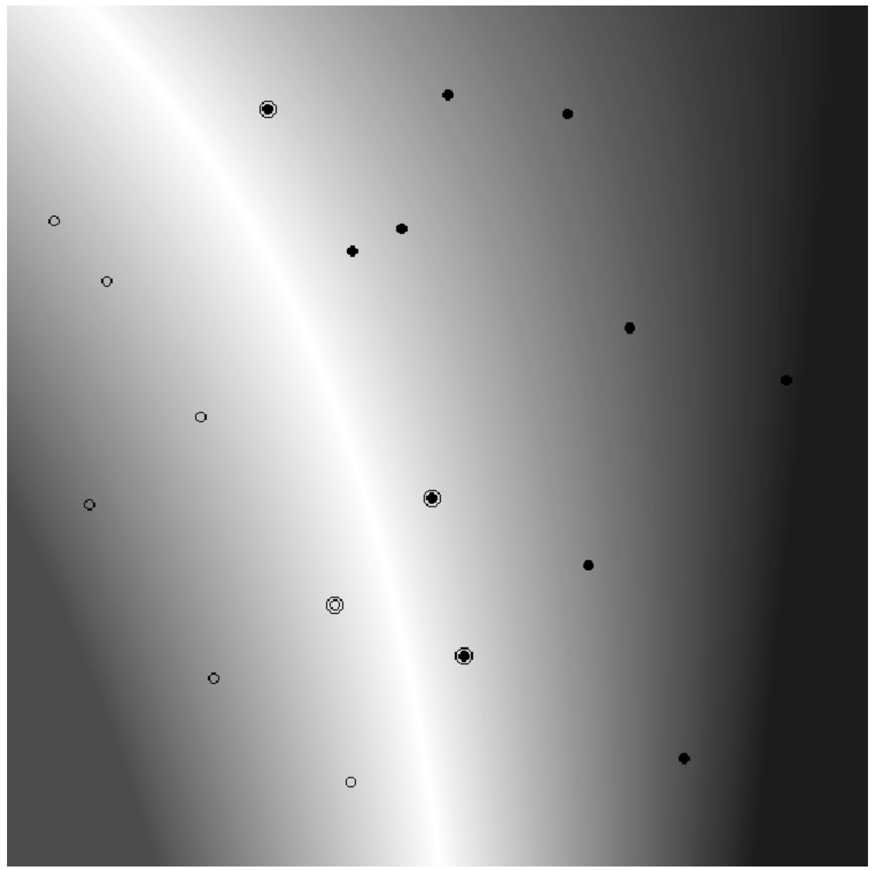
- Neural Network like kernel:

$$K(\mathbf{x}, \mathbf{x}') = \tanh(\theta + \phi \mathbf{x}\mathbf{x}')$$

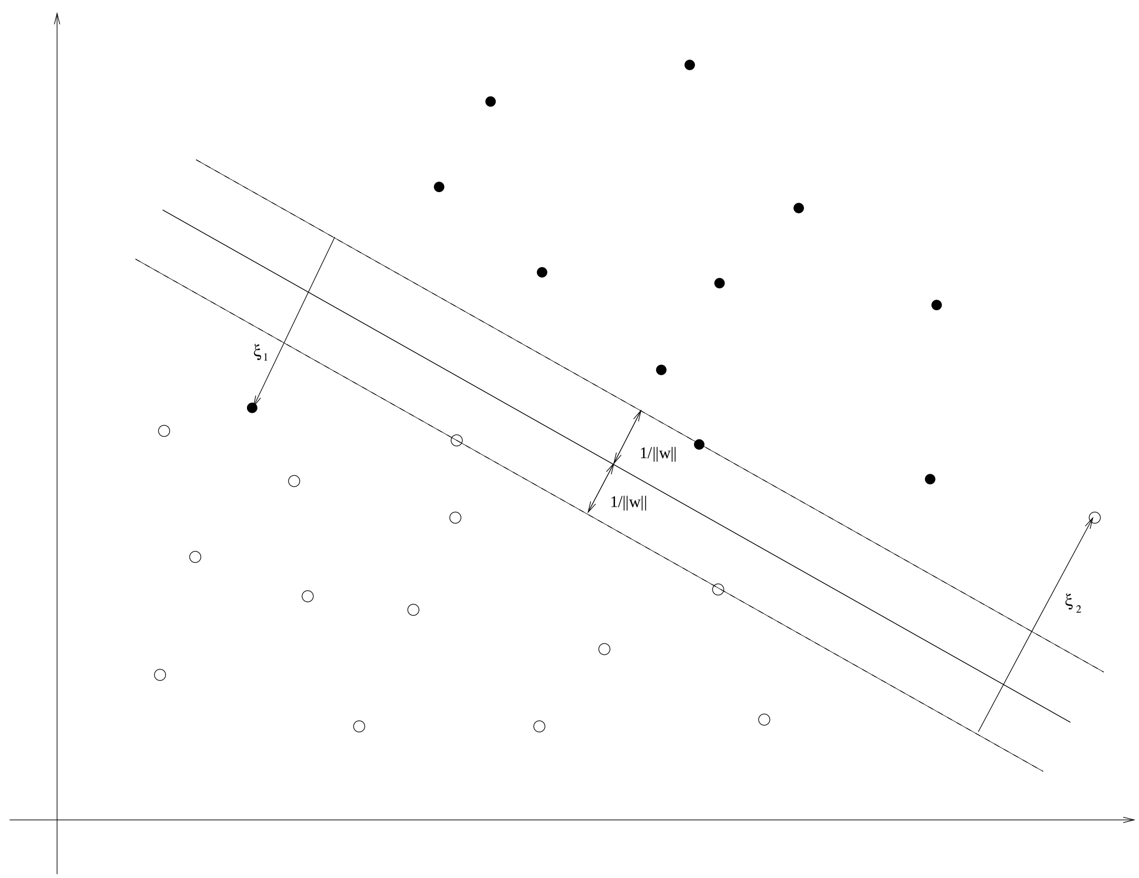
- Radial Function kernel:

$$K(\mathbf{x}, \mathbf{x}') = \exp(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\sigma^2})$$

Kernels. Example



Soft Margin Classifier



Soft Margin Classifier (cont.)

- Minimize:

$$\Gamma(\mathbf{w}, b, C, \xi) = \frac{1}{2}\mathbf{w}^2 + C\sum_i \xi_i, \quad y_i(\mathbf{w}\mathbf{x}_i + b) - 1 \geq -\xi_i,\ \xi_i \geq 0$$

$$L = \frac{1}{2}\mathbf{w}^2 + C\sum_{i=1}^{l} \xi_i - \sum_{i=1}^{l} \alpha_i[y_i(\mathbf{w}\mathbf{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^{l} \mu_i \xi_i,$$

$$\frac{\partial L}{\partial \alpha_i} = 0, \quad \frac{\partial L}{\partial \mu_i} = 0, \quad \alpha_i \geq 0, \quad \mu_i \geq 0$$

- Wolfe dual formulation: maximize L as a function of α_i and μ_i with the constrains:

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i, \quad \frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^{l} \alpha_i y_i = 0, \quad \frac{\partial L}{\partial \xi_i} = 0 \Rightarrow C = \alpha_i + \mu_i$$

Soft Margin Classifier (cont.)

- Karush-Kuhn-Tucker conditions at extremum:

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

$$\mu_i \xi_i = 0$$

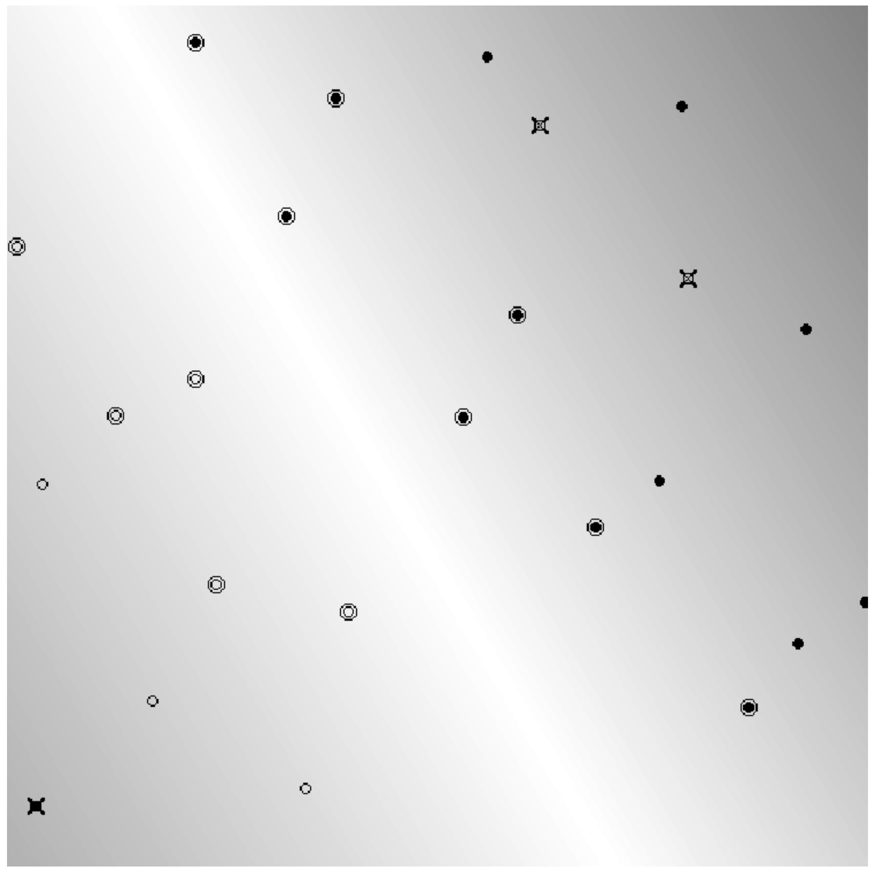
- Final optimization problem: maximize L as function of α_i

$$L = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \mathbf{x}_j, \quad \sum_{i=1}^{l} \alpha_i y_i = 0, \quad \alpha_i \in [0, C]$$

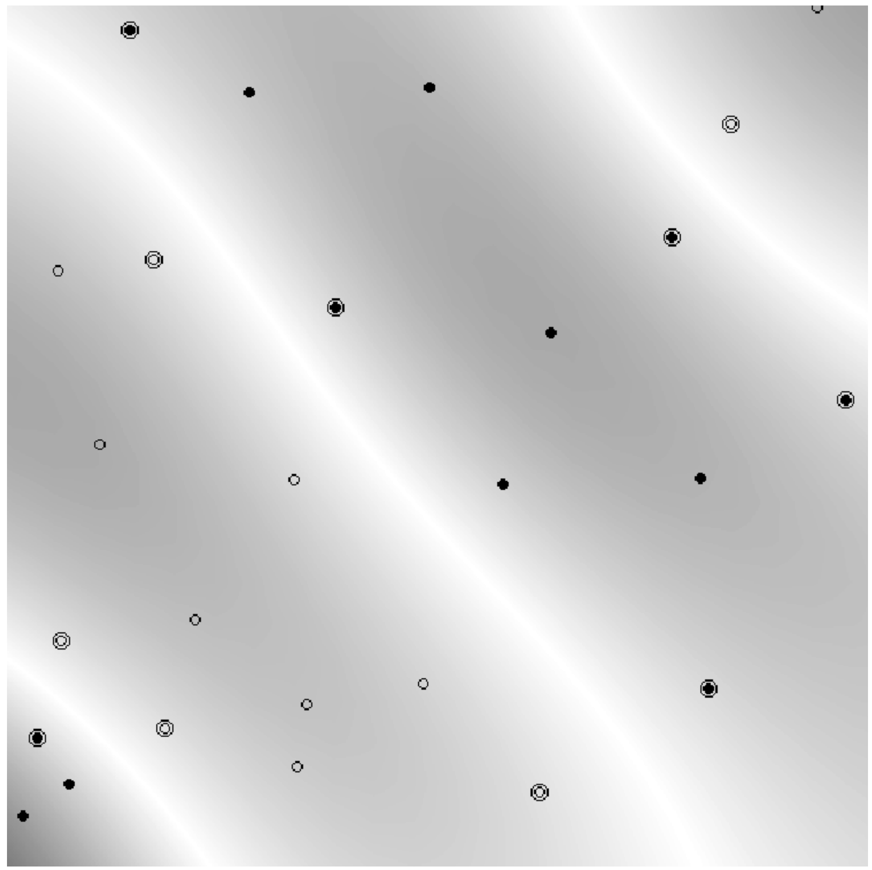
- Separating surface:

$$\sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i \mathbf{x} + b = 0$$

Soft Margin Classifier. Example 1



Soft Margin Classifier. Example 2



Optimization Problem

- The only practical issue is solving the Convex Quadratic Optimization Problem
- Properties of the optimization problem
  - has only one local optimum that is the global optimum
  - dimension proportional with the square of the number of training data (the quadratic constrain); solution usually cubic in the number of training data
  - the problem is the same if non support vectors are omitted from the training data
  - the solution is robust with respect to noise in the training data
- Commercial and Free packages to solve the optimization problem (OSL, MINOS, CPLEX, LOQO, BOTTOU, etc.)
- Large training sets are a big problem (50.000 training sets require 10GBytes only for the problem)
- Most of the research in this area is concentrated in finding better (approximative) optimization techniques. Some of the approximation methods: *chunking* and *working set*.

Applications and practical results

- Optical Character Recognition:
  - US Postal Service Database (9200 character samples):
    1. two layer neural network: 5.9%
    2. carefully tuned 5 layer neural network: 5.1%
    3. Vapnik et oth.(1992, Optimal Margin Classifier): 4.9%
    4. Cortes (1995, Soft Margin Classifier): 4.9%
    5. Vapnik et oth.(1996, Radial Based Kernel): 4.2%
    6. Vapnik et oth. (1997, Neural Network like Kernel): 4.1%
    7. humans: 2.5%
  - 1200 data-point from 10 subjects
    1. Typical back-propagation neural network: 12.7%
    2. Vapnik (Optimal Margin Classifier): 3.2% with linear kernel and 1.3% with second order polynomials
- State of the art results in *face detection* (Osuna, 1996) and *chair recognition* (Vapnik, 1996)