An introduction to Support Vector Machine

報告者：黃立德

References:

Simon Haykin, "Neural Networks: a comprehensive foundation, second edition“, 1999, Chapter 2,6

Nello Christanini, John Shawe-Tayer, “An Introduction to Support Vector Machines”, 2000, Chapter 3~6

Outline

Drawbacks of learning

- The choice of the class of functions from which the input/output mapping must be sought.
  - Learning in three-node neural networks is known to be NP-complete

Drawbacks of learning (cont.)

- In practice, there are following problems
  - The learning algorithm may prove inefficient as for example in the case of local minima
  - The size of output hypothesis can frequently become very large and impractical
  - If there are only a limited number of training examples, the hypothesis found by learning algorithm will lead to overfitting and hence poor generalization
  - The learning algorithm is usually controlled by a large number of parameters that are often chosen by turning heuristics, making the system difficult and unreliable to use

Overview of SVM

- What is SVM?
  - A linear machine with some very nice properties
  - The goal is to construct a decision surface such that the margin of separation between positive and negative samples is maximized.
  - SVM is a learning system that uses a hypothesis space of linear functions in a high dimensional feature space, trained with a learning algorithm from optimization theory that implements a learning bias derived from statistical learning theory

The Empirical Risk Minimization Principle

- Given a set of data

$$(x_1, y_1), \ldots, (x_N, y_N), x \in R^n, y_i \in \{-1, 1\}$$

- Also given a set of decision functions

$$\{f_\lambda : \lambda \in I\}, \; where \; f_\lambda : R^n \to \{-1, 1\}$$

- The expected risk is

$$R(\lambda) = \int |f_\lambda(x) - y| \, dP(x, y)$$

The Empirical Risk Minimization Principle (cont.)

- The approximation (empirical risk)

$$R_{emp}\left(\lambda\right)=\frac{1}{N}\sum_{i=1}^{N}\left|f_{\lambda}\left(x_i\right)-y_i\right|$$

- Theory of uniform convergence in probability

$$\lim_{N\to\infty} P\left\{\sup_{\lambda\in I}(R(\lambda)-R_{emp}(\lambda))>\varepsilon\right\}=0, \forall\varepsilon>0$$

Vapnik-Chervonenkis dimension

- VC-dimension
  - It is a measure of the capacity or expressive power of the family of classification functions realized by the learning machine

Structure Risk Minimization

- Let I_k be a subset of I

$$S_k = \left\{ f_\lambda : \lambda \in I_k \right\}$$

- Define a structure of nested subset

$$S_1 \subset S_2 \subset \ldots \subset S_n \subset \ldots$$

- Each subset satisfies the condition

$$h_1 \le h_2 \le \ldots \le h_n \le \ldots \qquad , h_i\text{: VC dimension}$$

Structure Risk Minimization (cont.)

- Implementing SRM can be difficult because the VC dimension of S_n could be hard to compute.

$$\min\left(R_{emp}(\lambda) + \sqrt{\frac{h_n}{N}} \right)$$

- Support Vector Machine, SVM, are able to achieve the goal of minimizing the upper bound of $R(\lambda)$ by minimizing a bound on the VC dimension h and $R_{emp}(\lambda)$ at the same time.

Concepts of SVM

- SVM is an approximate implementation of the method of structural risk minimization
- It does not incorporate problem-domain knowledge

Linearly separable patterns

- A training sample $\left\{(x_i, d_i)\right\}_{i=1}^{N}$
- The patterns are linear separable.
- The equation of a decision surface that does the separation is

$$w^t x + b = 0$$

- And we can write

$$\begin{cases} w^t x_i + b \geq 0 & for \quad d_i = +1 \\ w^t x_i + b < 0 & for \quad d_i = -1 \end{cases}$$

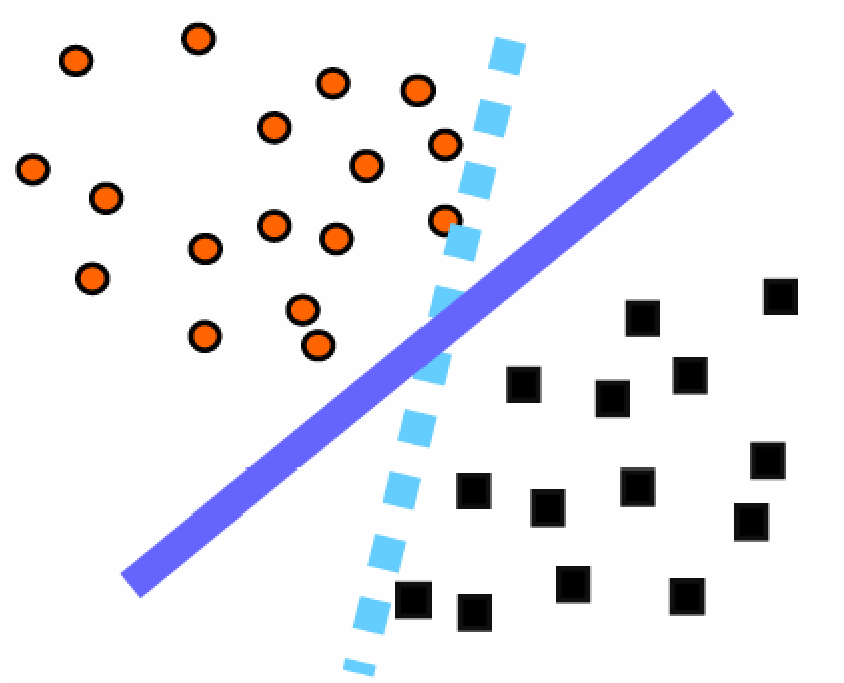
$$\Rightarrow \quad d_i(w^t x_i + b) \geq 1 \quad for \quad i = 1, 2, ..., N$$

Linearly separable patterns (cont.)

- The discriminant function of the optimal hyperplane is

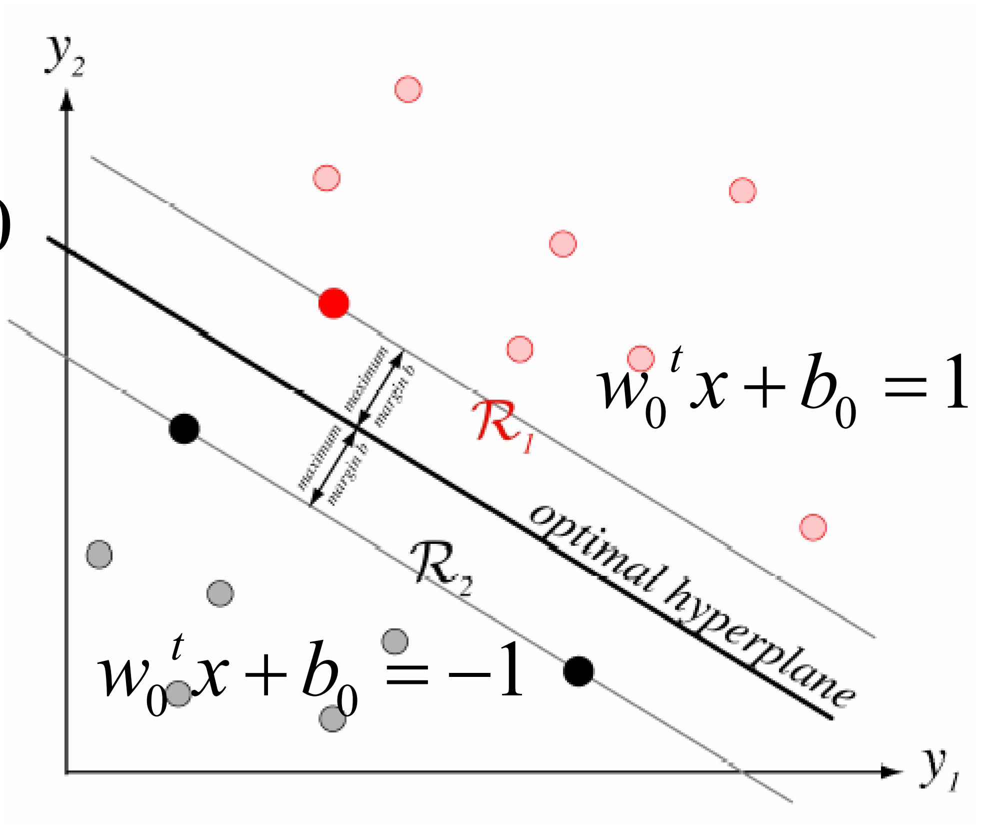
$$g(x) = w_0{}^t x + b_0$$

- Maximum Margin Rule
  - We select the hyperplane with maximum nearest the data point (support vectors).



Linearly separable patterns (cont.)

$$w_0{}^t x + b_0 = 0$$



$$w_0{}^t x + b_0 = 1$$

$$w_0{}^t x + b_0 = -1$$

$$Margin: \frac{2}{\|w_0\|}$$

Linearly separable patterns (cont.)

- The final goal is to minimizes the cost function

$$\Phi(w) = \frac{1}{2} w^t w$$

- We may solve the constrained optimization problem using the method of Lagrange multipliers.

$$\textit{Lagrangian function}$$

$$J(w, b, \alpha) = \frac{1}{2} w^t w - \sum_{i=1}^{N} \alpha_i \left[d_i (w^t w_i + b) - 1 \right]$$

$$\alpha_i : \textit{Lagrange multipliers}$$

Linearly separable patterns (cont.)

- dual form
  - Given the training sample $\{(x_i, d_i)\}_{i=1}^{N}$,find the Lagrange multipliers $\{\alpha_i\}_{i=1}^{N}$ that maximize the objective function

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j x_i^t x_j$$

subject to the constraints

(1) $\sum_{i=1}^{N} \alpha_i d_i = 0$

(2) $\alpha_i \geq 0 \quad for \quad i = 1, 2, \ldots, N$

Linearly separable patterns (cont.)

- Having determined the optimum Lagrange multipliers, we can compute the optimum weight vector and bias.

$$w_0 = \sum_{i=1}^{N} \alpha_{o,i} d_i x_i$$

$$b_0 = 1 - w_0{}^t x^{(s)} \quad for \quad d^{(s)} = 1$$

Non-linearly separable patterns

- allow training errors
- The definition of decision surface is

$$d_i(w^t x_i + b) \geq 1 - \xi_i \quad for \quad i = 1, 2, ..., N$$

- At last, we only have to minimizing the following function

$$\Phi(w, \xi) = \frac{1}{2} w^t w + C \sum_{i=1}^{N} \xi_i$$

Non-linearly separable patterns (cont.)

- dual form
  - Given the training sample $\{(x_i, d_i)\}_{i=1}^{N}$,find the Lagrange multipliers $\{\alpha_i\}_{i=1}^{N}$ that maximize the objective function

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j x_i^{\,t} x_j$$

subject to the constraints

(1) $\sum_{i=1}^{N} \alpha_i d_i = 0$

(2) $0 \le \alpha_i \le C \quad for \quad i = 1, 2, \ldots, N$

where C is a user-specified positive parameter

Non-linearly separable patterns (cont.)

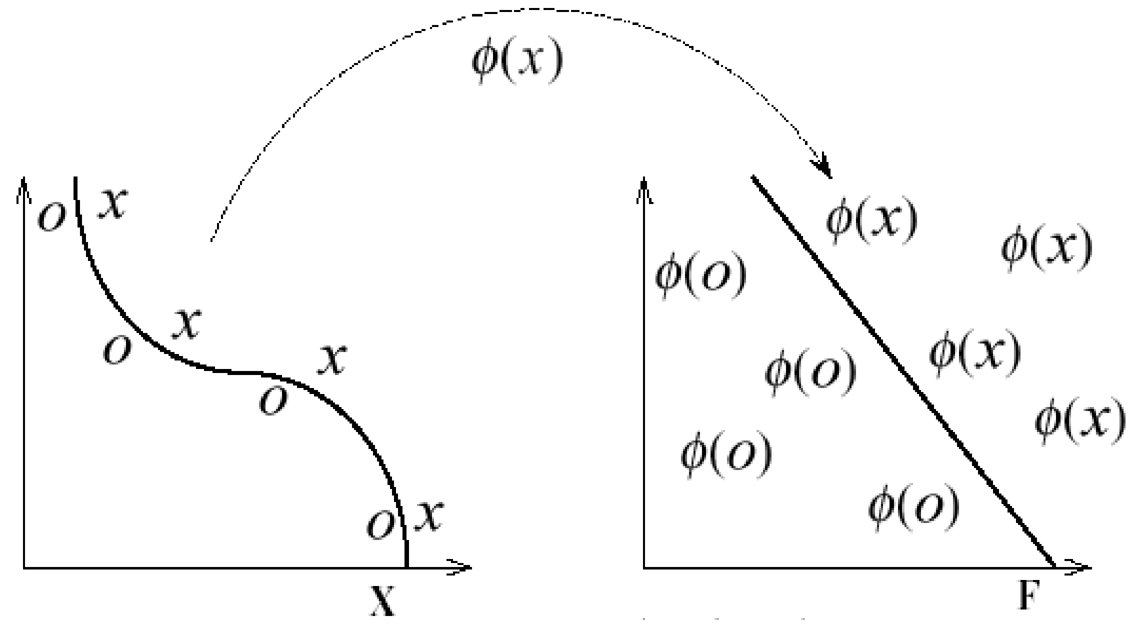
- After the optimum Lagrange multipliers have determined, we can compute the optimum weight vector and bias.

$$w_0 = \sum_{i=1}^{N_s} \alpha_{o,i} d_i x_i$$

$$b_0 = 1 - w_0{}^t x^{(s)} \quad for \quad d^{(s)} = 1$$

How to build a SVM for pattern recognition

- Steps for constructing a SVM
  - Nonlinear mapping of input vectors into a high dimensional feature space that is hidden from both the input and output.
  - Construction of an optimal hyperplane for separating the features.



How to build a SVM for pattern recognition (cont.)

- x denotes a vector from the input space.
- $\{\varphi_j(x)\}_{j=1}^{m_1}$ denotes a set of nonlinear mapping from the input space to the feature space.
- Define a hyperplane as following

$$\sum_{j=1}^{m_1} w_j \varphi_j(x) + b = 0 \quad \Longrightarrow \quad \sum_{j=0}^{m_1} w_j \varphi_j(x) = 0$$

- Define the vector

$$\varphi(x) = \left[\varphi_0(x), \varphi_1(x), \ldots, \varphi_{m_1}(x) \right]^t$$

How to build a SVM for pattern recognition (cont.)

- We can write the equation in the compact form.

$$\mathbf{w}^t \varphi(x) = 0 \qquad \boxed{1}$$

- Because the features are linear separable, we may write

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i d_i \varphi(x_i) \qquad \boxed{2}$$

- Substituting Eq.2 in Eq.1,we get

$$\sum_{i=1}^{N} \alpha_i d_i \varphi^t(x_i)\varphi(x) = 0$$

How to build a SVM for pattern recognition (cont.)

- Define the inner-product kernel denoted by

$$\begin{aligned} K(x, x_i) &= \varphi^t(x)\varphi(x_i) \\ &= \sum_{j=0}^{m_1} \varphi_j(x)\varphi_j(x_i) \qquad for\, i = 1, \ldots, N \end{aligned}$$

- Now, we may use the inner-product kernel to construct the optimal decision surface in the feature space without considering the feature space in explicit form.

$$\sum_{i=1}^{N} \alpha_i d_i K(x, x_i) = 0$$

How to build a SVM for pattern recognition (cont.)

- Optimum design of a SVM (dual form)
  - Given the training sample $\left\{(x_i, d_i)\right\}_{i=1}^{N}$,find the Lagrange multipliers $\left\{\alpha_i\right\}_{i=1}^{N}$ that maximize the objective function

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j K(x_i, x_j)$$

subject to the constraints

(1) $\sum_{i=1}^{N} \alpha_i d_i = 0$

(2) $0 \le \alpha_i \le C \quad for \quad i = 1, 2, \ldots, N$

where C is a user-specified positive parameter

How to build a SVM for pattern recognition (cont.)

- We may view $K(x_i, x_j)$ as the *ij-th* element of a symmetric N-by-N matrix $\mathbf{K}$

$$\mathbf{K} = \left\{ K(x_i, x_j) \right\}_{(i,j)=1}^{N}$$

- Having found the optimum values of $\alpha_{o,i}$, we can get

$$\mathbf{w}_o = \sum_{i=1}^{N} \alpha_{o,i} d_i \varphi(x_i)$$

How to build a SVM for pattern recognition (cont.)

Summary of Inner-Product Kernels

Type of support vector machine	Inner product kernel $K(x, x_i), i = 1, 2, \ldots, N$	Comments
Polynomial learning machine	$(x^t x_i + 1)^p$	Power p is specified *a priori* by the user
Radial-basis function network	$\exp\left(-\frac{1}{2\sigma^2}\lVert x - x_i \rVert^2\right)$	The width σ^2 ,common to all the kernels, is specified *a priori* by the user
Two-layer perceptron	$\tanh\left(\beta_0 x^t x_i + \beta_1\right)$	Mercer's theorem is satisfied only for some values of β_0 and β_1

Example: XOR problem

- First, we choose kernel as

$$K(x, x_i) = \left(1 + x^t x_i\right)^2$$

- With $x = \left[x_1, x_2\right]^t$ and $x_i = \left[x_{i1}, x_{i2}\right]^t$,we get

$$K(x, x_i) = 1 + x_1{}^2 x_{i1}{}^2 + 2x_1 x_2 x_{i1} x_{i2} + x_2{}^2 x_{i2}{}^2 + 2x_1 x_{i1} + 2x_2 x_{i2}$$

$$\Longrightarrow \quad \varphi(x) = \left[1, x_1{}^2, \sqrt{2}x_1 x_2, x_2{}^2, \sqrt{2}x_1, \sqrt{2}x_2\right]^t$$

$$\Longrightarrow \quad \varphi(x_i) = \left[1, x_{i1}{}^2, \sqrt{2}x_{i1} x_{i2}, x_{i2}{}^2, \sqrt{2}x_{i1}, \sqrt{2}x_{i2}\right]^t, i = 1, 2, 3, 4$$

Example: XOR problem (cont.)

- We also find that

$$\mathbf{K} = \begin{bmatrix} 9 & 1 & 1 & 1 \\ 1 & 9 & 1 & 1 \\ 1 & 1 & 9 & 1 \\ 1 & 1 & 1 & 9 \end{bmatrix}$$

- The objective function for the dual form is

$$Q(\alpha) = \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 - \frac{1}{2}(9\alpha_1{}^2 - 2\alpha_1\alpha_2 - 2\alpha_1\alpha_3 + 2\alpha_1\alpha_4$$
$$+9\alpha_2{}^2 + 2\alpha_2\alpha_3 - 2\alpha_2\alpha_4 + 9\alpha_3{}^2 - 2\alpha_3\alpha_4 + 9\alpha_4{}^2)$$

Example: XOR problem (cont.)

- Optimizing $Q(\alpha)$

$$\frac{\partial Q(\alpha)}{\partial \alpha_1} \Rightarrow 9\alpha_1 - \alpha_2 - \alpha_3 + \alpha_4 = 1$$

$$\frac{\partial Q(\alpha)}{\partial \alpha_2} \Rightarrow -\alpha_1 + 9\alpha_2 + \alpha_3 - \alpha_4 = 1$$

$$\frac{\partial Q(\alpha)}{\partial \alpha_3} \Rightarrow -\alpha_1 + \alpha_2 + 9\alpha_3 - \alpha_4 = 1$$

$$\frac{\partial Q(\alpha)}{\partial \alpha_4} \Rightarrow \alpha_1 - \alpha_2 - \alpha_3 + 9\alpha_4 = 1$$

Example: XOR problem (cont.)

- The optimum values of $\alpha_{o,i}$ are

$$\alpha_{o,1} = \alpha_{o,2} = \alpha_{o,3} = \alpha_{o,4} = \frac{1}{8}$$

⇨ $$Q_o(\alpha) = \frac{1}{4}$$

⇨ $$\frac{1}{2}\|\mathbf{w}_o\|^2 = \frac{1}{4} \Rightarrow \|\mathbf{w}_o\| = \frac{1}{\sqrt{2}}$$

Example: XOR problem (cont.)

- We find that the optimum weight vector is

$$\mathbf{w}_o = \frac{1}{8}\left[-\varphi(x_1)+\varphi(x_2)+\varphi(x_3)-\varphi(x_4)\right]$$

$$= \frac{1}{8}\left[-\begin{bmatrix}1\\1\\\sqrt{2}\\1\\-\sqrt{2}\\-\sqrt{2}\end{bmatrix}+\begin{bmatrix}1\\1\\-\sqrt{2}\\1\\-\sqrt{2}\\\sqrt{2}\end{bmatrix}+\begin{bmatrix}1\\1\\-\sqrt{2}\\1\\\sqrt{2}\\-\sqrt{2}\end{bmatrix}-\begin{bmatrix}1\\1\\\sqrt{2}\\1\\\sqrt{2}\\\sqrt{2}\end{bmatrix}\right]=\begin{bmatrix}0\\0\\-1/\sqrt{2}\\0\\0\\0\end{bmatrix}$$

Example: XOR problem (cont.)

- The optimal hyperplane is defined by

$$\mathbf{w}_o{}^t \varphi(x) = 0$$

$$\Rightarrow \quad \left[0, 0, \frac{-1}{\sqrt{2}}, 0, 0, 0\right] \begin{bmatrix} 1 \\ x_1{}^2 \\ \sqrt{2}x_1x_2 \\ x_2{}^2 \\ \sqrt{2}x_1 \\ \sqrt{2}x_2 \end{bmatrix} = 0$$

$$\Rightarrow \quad -x_1x_2 = 0$$

Properties and expansions of SVM

- Two important features:
  - Duality is the first feature of SVM
  - Operate in a kernel induced feature space
- Several expansions of SVM:
  - C-Support Vector Classification (binary case)
  - ν-Support Vector Classification (binary case)
  - Distribution Estimation (one-class SVM)
  - ε-Support Vector Regression (ε -SVR)
  - ν-Support Vector Regression (ν-SVR)

Conclusion

- The SVM is an elegant and highly principled learning method for the design of classifying nonlinear input data.
- Compared with back-propagation algorithm
  - Only operate in a batch mode
  - Whatever the learning task, it provide a method for controlling model complexity independently of dimensionality
  - It is guaranteed to find a global extremum of the error surface
  - The computation can be performed efficiently
  - By using a suitable inner-product kernel, the SVM computes all the important network parameters automatically.

Applications of SVM

- Classification
- Regression
- Recognition
- Bioinformatics

LIBSVM

- A Library for Support Vector Machines
- Made by Chih-Jen Lin and Chih-Chung Chang
- Both C++ and Java sources
- http://www.csie.ntu.edu.tw/~cjlin/