Support Vector Machines

Hsing-Kuo Pao (鮑興國)
National Taiwan University of Science & Technology (Taiwan Tech)

Outline

- Introduction
- Linear SVMs
- Kernel Trick and Nonlinear SVMs
- Algorithms and Tuning Procedures
- Variants and extensions of SVMs
- Applications

Introduction

— Support Vector Machines

History of Support Vector Machines

- SVMs introduced in COLT-92 by Boser, Guyon & Vapnik. Became rather popular since
- Theoretically well motivated algorithm: developed from Statistical Learning Theory (Vapnik & Chervonenkis) since the 60s
- Empirically good performance: successful applications in many fields (bioinformatics, text, image recognition, . . .)
- A large and diverse community work on them: from machine learning, optimization, statistics, neural networks, functional analysis, etc.

The Recent Variation

- The long debate between Artificial Neural Networks and SVMs

  數十年劍宗與氣宗的論劍！

Why Support Vector Machines?

- SVM classifier is an optimally defined surface
- SVMs have a good geometric interpretation
- SVMs can be generated very efficiently
- Can be extended from linear to nonlinear case
  - Typically nonlinear in the input space
  - Linear in a higher dimensional “feature” space
  - Implicitly defined by a kernel function
- Have a sound theoretical foundation
  - Based on Statistical Learning Theory

Preliminaries

- SVM aims to solve the binary classification problem in the typical sense
  - Such as to separate between the cat images and dog images
  - Can extend to multi-class classification later
- SVM is assumed to be deterministic: no probability involved in its typical form
- SVM is formulated as an optimization problem
- One of the few methods that often “prefer” to working on high dimensional space
  - Not necessarily contradicts to dimensional reduction
  - ANNs or DNs may also have shrinking (more often) or expanding structures

Risks and Error Bound

- What is the optimization problem?
- Expected risk

$$R(\alpha) = \int \frac{1}{2} |y - f(\mathbf{x}, \alpha)| \, , dP(\mathbf{x}, y)$$

- Empirical risk

$$R_{\text{emp}}(\alpha) = \frac{1}{2m} \sum_{i=1}^{m} |y_i - f(\mathbf{x}_i, \alpha)|$$

- Risk bound

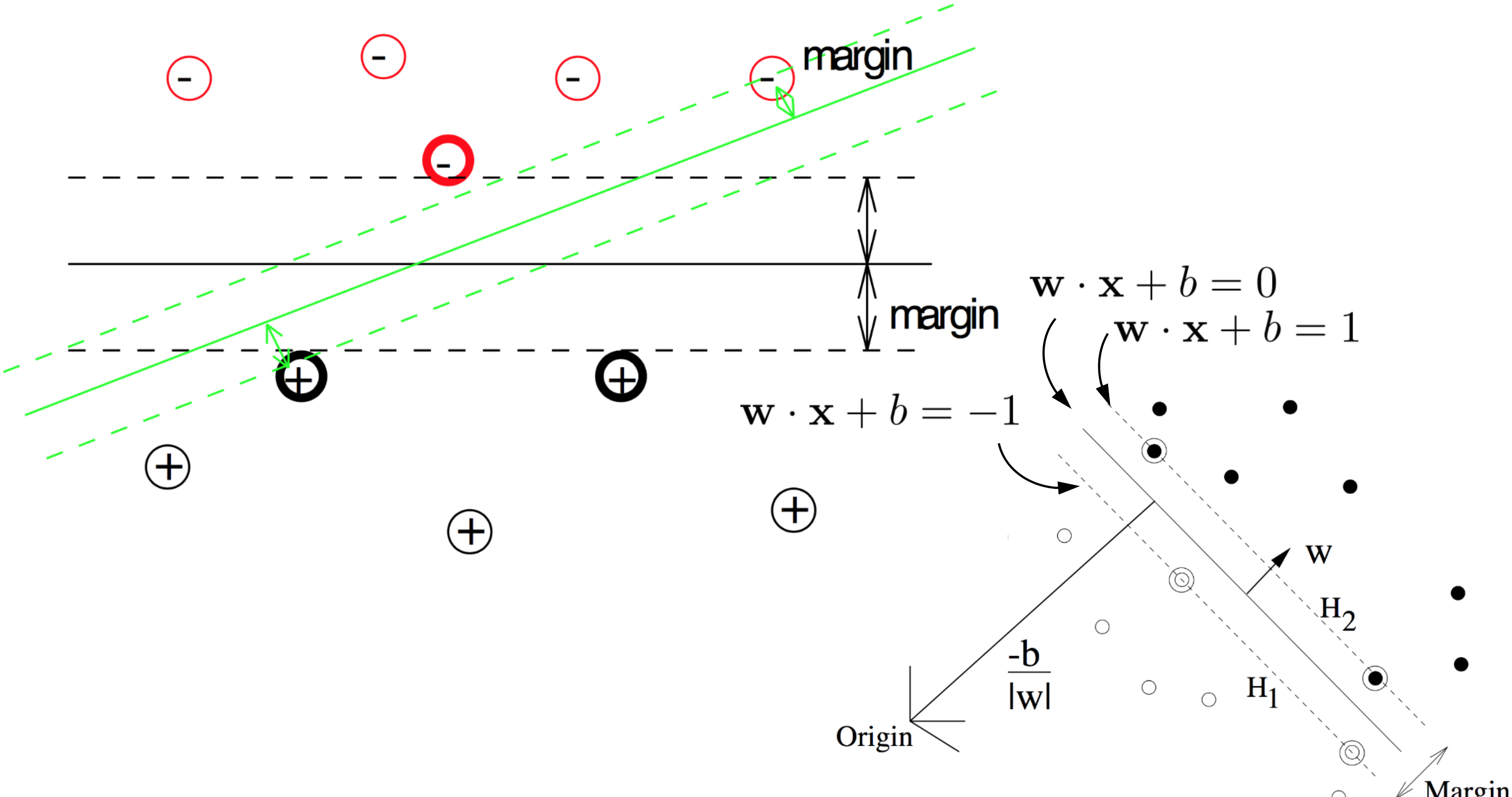
$$R(\alpha) \leq R_{\text{emp}}(\alpha) + \sqrt{\left(\frac{h(\log(2m/h) + 1) - \log(\eta/4)}{m} \right)}$$

holds with probability $1 - \eta$
for a chosen $0 \leq \eta \leq 1$, and VC dimension h

Linear Support Vector Machines

— Support Vector Machines

Maximizing the Margin between Bounding Planes

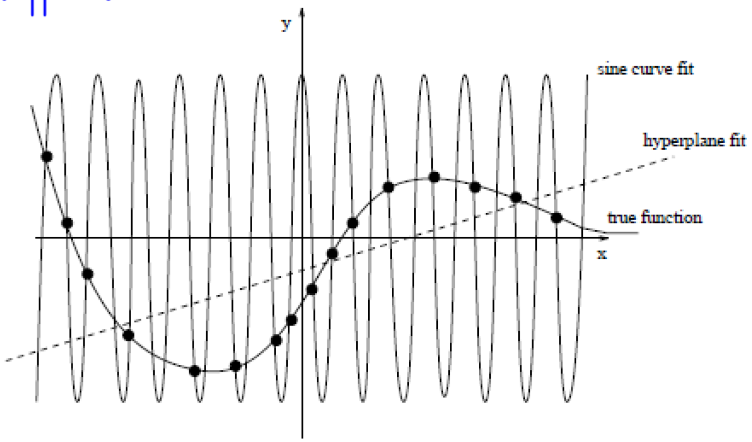


- Maximizing margin equivalent to regularization
- Boser, Guyon, Vapnik ’92, and Cortes & Vapnik ’95

A Generic ML Model

- Most machine learning models aim to minimize the following error functional:

$$
\begin{aligned}
E(\mathbf{w}) &= -\mathrm{fitting}(\mathbf{w}) - \mathrm{smoothness}(\mathbf{w}) \\
&= \mathrm{training\_error}(\mathbf{w}) + \mathrm{complexity}(\mathbf{w}) \\
&= \frac{1}{m}\sum_{i=1}^{m} L(f(\mathbf{x}_i, \alpha), y_i) + \|\mathbf{w}\|^2 \,?
\end{aligned}
$$



- Keywords: training error vs. test error, validation, regularization, generalization

The Linearly Separable Case

- Given m points in the n dimensional real space $\mathbb{R}^n$
- Two classes: Y_-, Y_+
- No error assumption
- The constraints for perfect classification:

$$\begin{aligned} \mathbf{x}_i \cdot \mathbf{w} + b &\geq +1, \quad \text{for } y_i = +1\,, \\ \mathbf{x}_i \cdot \mathbf{w} + b &\leq -1, \quad \text{for } y_i = -1\,. \end{aligned}$$

- Or combined into one:

$$y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0 \quad \forall i$$

- Predict the membership of a new data point $\mathbf{x}$

$$\mathbf{x} \cdot \mathbf{w} + b \geq 0\,,\ \mathbf{x} \in Y_+ \text{ otherwise } \mathbf{x} \in Y_-$$

In Matrix Formulation

- An $m \times n$ data matrix A
- Membership of each point A_i in the classes A_- or A_+ is specified by an $m \times m$ diagonal matrix D:

$$D_{ii} = -1 \text{ if } A_i \in A_- \text{ and } D_{ii} = 1 \text{ if } A_i \in A_+$$

- Separate A_- and A_+ by two bounding planes such that:

$$\begin{aligned} A_i\mathbf{w} + b &\geq +1, \quad \text{for } D_{ii} = +1\,, \\ A_i\mathbf{w} + b &\leq -1, \quad \text{for } D_{ii} = -1\,. \end{aligned}$$

- Predict the membership of a new data point $\mathbf{x}$

$$\mathbf{x}^T\mathbf{w} + b \geq 0\,,\ \mathbf{x} \in A_+ \text{ otherwise } \mathbf{x} \in A_-$$

Summary of Notations

- Let $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}$ be a training set represented by matrices

$$A = \begin{bmatrix} (\mathbf{x}_1)^T \\ (\mathbf{x}_2)^T \\ \vdots \\ (\mathbf{x}_m)^T \end{bmatrix} \in \mathbb{R}^{m \times n}, D = \begin{bmatrix} y_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & y_m \end{bmatrix} \in \mathbb{R}^{m \times m}$$

- $$\begin{aligned} A_i\mathbf{w} + b &\geq +1, \quad \text{for } D_{ii} = +1, \\ A_i\mathbf{w} + b &\leq -1, \quad \text{for } D_{ii} = -1. \end{aligned}$$ equivalent to

$$D(A\mathbf{w} + b\,\mathbf{e}) \geq \mathbf{e}, \quad \text{where } \mathbf{e} = [1, 1, \ldots, 1]^T \in \mathbb{R}^m.$$

Lagrange Multiplier Methods with Equality Constraints

- Problem:

$$\min_{\mathbf{x}=(x_1,x_2,\ldots,x_n)} J = f(\mathbf{x})$$

$$\text{such that } g_k(\mathbf{x}) = 0\,, \forall k = 1, \ldots, K$$

- Transformed problem and its solution:
  - Working on minimizing the augmented function

$$J_A(\mathbf{x}, \lambda_1, \ldots, \lambda_K) = f(\mathbf{x}) + \sum_{k=1}^{K} \lambda_k g_k(\mathbf{x})$$

  - No constraints on the Lagrange multipliers λ_k
  - Solving: $\nabla J_A = \mathbf{0}$

Lagrange Multiplier Methods with Inequality Constraints

- Problem:

$$\min_{\mathbf{x}=(x_1,x_2,\ldots,x_n)} \quad J = f(\mathbf{x})$$

$$\text{such that } g_k(\mathbf{x}) \leq 0\,, \forall k = 1, \ldots, K$$

- Transformed problem and its solution:
  - Working on minimizing the augmented function:

$$J_A(\mathbf{x}, \lambda_1, \ldots, \lambda_K) = f(\mathbf{x}) + \sum_{k=1}^{K} \lambda_k g_k(\mathbf{x})\,, \quad \forall \lambda_k \geq 0$$

    with nonnegative constraints on the Lagrange multipliers λ_k
  - Solving $\nabla J_A = \mathbf{0}$, with other constraints!

Primal vs. Dual Formulation

- Primal form

$$L_P \equiv \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{m} \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{w} + b) + \sum_{i=1}^{m} \alpha_i$$

- Dual form

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

- Dual form can be derived from the primal form using Lagrange multiplier method

$$\frac{\partial L_P}{\partial w_j} = w_j - \sum_i \alpha_i y_i x_{ij} = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$

$$\frac{\partial L_P}{\partial b} = -\sum_{i=1}^{m} \alpha_i y_i = 0$$

The Karush-Kuhn-Tucter (KKT) Condition

- Given a general problem:

$$\begin{array}{lll} \min\limits_{\mathbf{x}\in\mathbb{R}^n} & f(\mathbf{x}) & \\ \text{subject to} & h_i(\mathbf{x}) \le 0\,, & i = 1,\ldots,r \\ & \ell_j(\mathbf{x}) = 0\,, & j = 1,\ldots,s \end{array}$$

- The KKT conditions are:
  - For the augmented function f_A

$$\begin{array}{rcll} \nabla f_A & = & \mathbf{0} & \\ u_i \cdot h_i(\mathbf{x}) & = & 0\,, & \forall i \\ h_i(\mathbf{x}) & \le & 0\,, & \forall i, \qquad \ell_j(\mathbf{x}) \;=\; 0\,, \quad \forall j \\ u_i & \ge & 0\,, & \forall i \end{array}$$

The KKT Condition (cont'd)

- In this example:

$$
\begin{aligned}
\frac{\partial}{\partial w_j} L_P = w_j - \sum_i \alpha_i y_i x_{ij} &= 0\,, \quad j = 1, \ldots, n \\
\frac{\partial}{\partial b} L_P = -\sum_i \alpha_i y_i &= 0 \\
y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 &\geq 0\,, \quad i = 1, \ldots, m \\
\alpha_i &\geq 0\,, \quad \forall i \\
\alpha_i(y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1) &= 0\,, \quad \forall i
\end{aligned}
$$

A View from Perceptron Algorithm

- Perceptron update:

$$w_j^{(t+1)} \leftarrow w_j^{(t)} + \Delta w_j$$

$$\Delta w_j = \eta(y_i - h(\mathbf{x}_i))\, x_{ij} = \eta(y_i - s(\mathbf{w} \cdot \mathbf{x}_i + b))\, x_{ij}$$

- Algorithm:

if $y_i(\mathbf{w}^{(t)} \cdot \mathbf{x}_i + b) \leq 0$ then

$$w_j^{(t)} \leftarrow w_j^{(t)} + \eta y_i x_{ij}$$

$$b^{(t)} \leftarrow b^{(t)} + \eta y_i R^2$$

$$t \leftarrow t + 1$$

end if

- After a few iterations…

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$ (only the non-zero terms matter!)

Robust Linear Programming

- For the linearly separable case, at solution of (LP):

$$\min_{\mathbf{w},b,\xi_i} \quad \textstyle\sum_i \xi_i$$

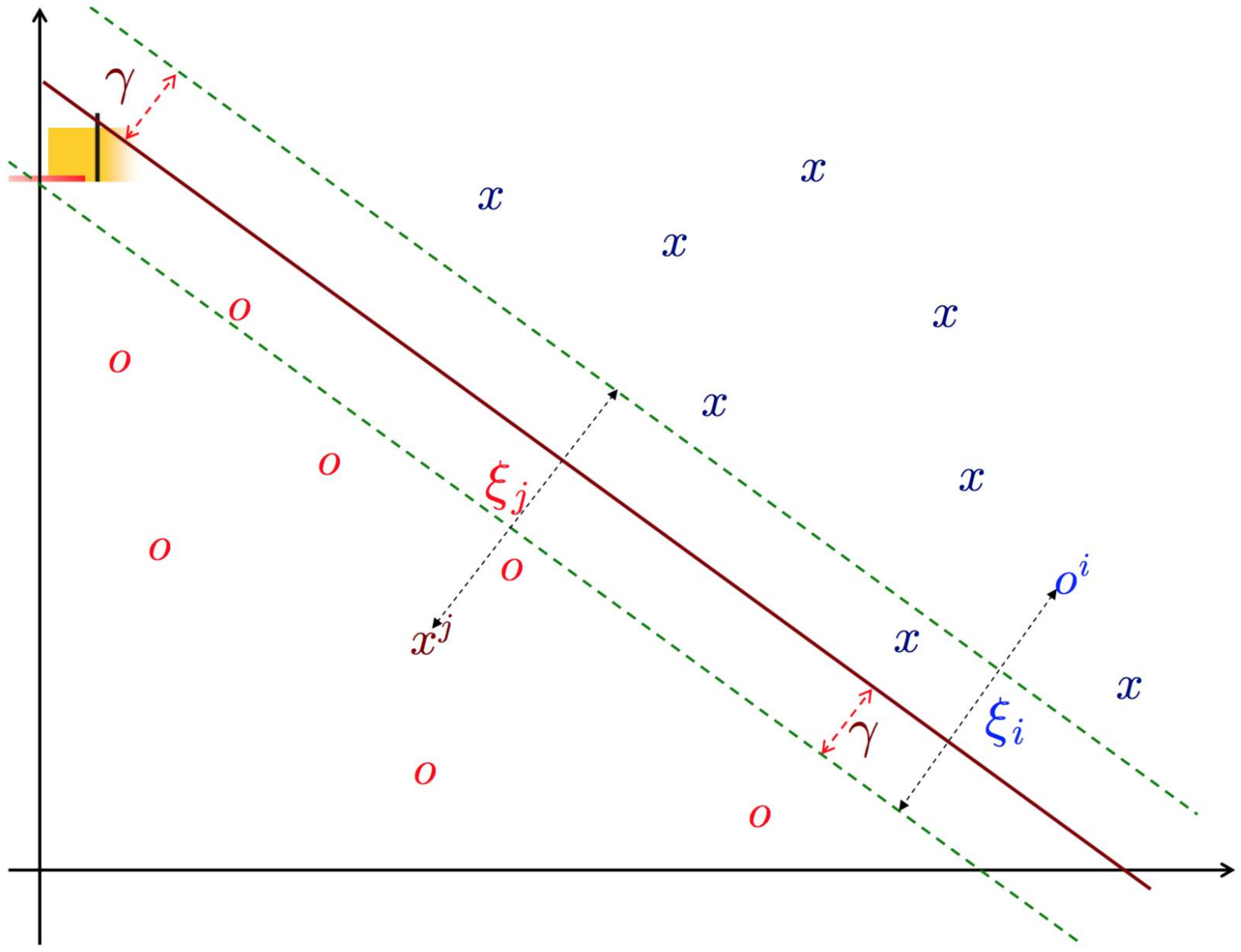
$$y_i(\mathbf{x}_i \cdot \mathbf{w} + b) + \xi_i - 1 \geq 0 \qquad \forall i$$

$$\xi_i \geq 0$$

- The training error $\sum_i \xi_i$
- For the linearly separable case, at solution of LP:

$$\xi_i = 0$$

Robust Linear Programming (cont'd)



Support Vector Machines with Different Regularizations

- 2-norm soft margin:

$$\min_{(\mathbf{w},b,\xi)\in\mathbb{R}^{n+1+m}} \frac{1}{2}\|\mathbf{w}\|_2^2 + \frac{C}{2}\|\xi\|_2^2$$
$$D(A\mathbf{w} + b\,\mathbf{e}) + \xi \geq \mathbf{e}$$

- 1-norm soft margin:

$$\min_{(\mathbf{w},b,\xi)\in\mathbb{R}^{n+1+m}} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\mathbf{e}^T\xi$$
$$D(A\mathbf{w} + b\,\mathbf{e}) + \xi \geq \mathbf{e}\,, \quad \xi \geq 0$$

- Margin is maximized by minimizing reciprocal of margin

References

- “Statistical learning theory” by V. N. Vapnik
- “An introduction to Support Vector Machines” by Cristianini and Shawe-Taylor
- “A Tutorial on Support Vector Machines for Pattern Recognition” by C. J. C. Burges, 1998.
- A resourceful website: www.kernel-machines.org
- Support vector machine slides from Yuh-Jye Lee, National Chiao Tung university
- Support vector machine tutorial slides by Jason Weston, NEC Labs America