

Support Vector Machines and Kernel-Based Algorithms for Machine Learning

An Introduction

Mehmet Gönen

Department of Computer Engineering
Boğaziçi University

18.05.2007

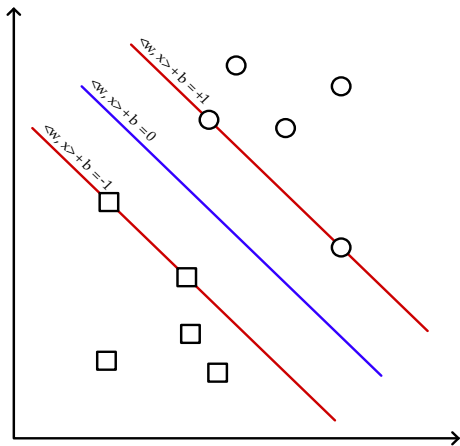
Support Vector Machine (SVM) Theory

- 1 Binary Classification Problem
- 2 Hard Margin SVM
- 3 Soft Margin SVM
- 4 Regression Problem
- 5 Regression SVM
- 6 Kernel Functions
- 7 Comments

Binary Classification Problem Definition

- Given empirical dataset (\mathbf{X}, \mathbf{Y})
 $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n) \in \mathbb{R}^d \times \{-1, +1\}$
- Separate two classes linearly
 $(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq +1$ if $y_i = +1$
 $(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \leq -1$ if $y_i = -1$
- More succinctly, find a hyperplane such that
 $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq +1$
- Decision function becomes
 $f(\mathbf{x}) = \mathbf{sgn}(\langle \mathbf{w}, \mathbf{x} \rangle + b)$

Geometric Motivation for Hard Margin SVM



- Distance to discriminant $|\langle \mathbf{w}, \mathbf{x}_i \rangle + b| / \|\mathbf{w}\|$
- We require $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) / \|\mathbf{w}\| \geq \rho$
- To obtain a unique solution $\rho \|\mathbf{w}\| = 1$

Optimization Problem

$$\begin{array}{ll} \underset{\mathbf{w}, b}{\text{minimize}} & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{subject to} & y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 \quad i = 1, \dots, n \end{array}$$

Lagrangian Dual

$$\begin{aligned} L(\mathbf{w}, b, \boldsymbol{\alpha}) &= \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i (y_i (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - 1) \\ \frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial \mathbf{w}} &= 0 \Rightarrow \mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i \\ \frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial b} &= 0 \Rightarrow \sum_{i=1}^n \alpha_i y_i = 0 \end{aligned}$$

Dual Optimization Problem

$$\begin{aligned} \text{maximize}_{\alpha} \quad & \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ \text{subject to} \quad & \sum_{i=1}^n \alpha_i y_i = 0 \\ & \alpha_i \geq 0 \quad i = 1, \dots, n \end{aligned}$$

Decision Function

$$\begin{aligned} f(\mathbf{x}) &= \text{sgn}(\langle \mathbf{w}, \mathbf{x} \rangle + b) \\ f(\mathbf{x}) &= \text{sgn}\left(\sum_{i=1}^n \alpha_i y_i \langle \mathbf{x}_i, \mathbf{x} \rangle + b\right) \end{aligned}$$

- Only positive α_i 's contribute
- They are called **support vectors**

Karush-Kuhn-Tucker Theorem

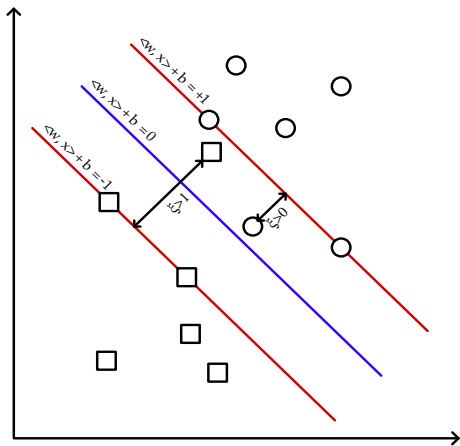
$$\alpha_i [y_i(\langle w, \mathbf{x}_i \rangle + b) - 1] = 0 \quad i = 1, \dots, n$$

$$\alpha_i > 0 \Rightarrow y_i(\langle w, \mathbf{x}_i \rangle + b) - 1 = 0$$

$$\alpha_i = 0 \Rightarrow y_i(\langle w, \mathbf{x}_i \rangle + b) - 1 > 0$$

- \mathbf{x}_i 's with $\alpha_i > 0$ are on separating hyperplanes (**support vectors**)
- b can be calculated on one of these instances
(Numerically safer to get average on all \mathbf{x}_i 's with $\alpha_i > 0$)
- \mathbf{x}_i 's with $\alpha_i = 0$ are beyond separating hyperplanes
- No need to store them

Geometric Motivation for Soft Margin SVM



- Allow misclassification
 $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i$
- Minimize misclassification
 $\#(\xi_i \geq 1)$
- Hard to solve
- Instead use total soft error
$$\sum_{i=1}^n \xi_i$$

Optimization Problem

$$\begin{aligned} \underset{\mathbf{w}, b, \xi}{\text{minimize}} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i \\ \text{subject to} \quad & y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i \quad i = 1, \dots, n \\ & \xi_i \geq 0 \quad i = 1, \dots, n \end{aligned}$$

Lagrangian Dual

$$L(\mathbf{w}, b, \xi, \alpha, \beta) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i (y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - 1 + \xi_i) - \sum_{i=1}^n \beta_i \xi_i$$

$$\frac{\partial L(\mathbf{w}, b, \xi, \alpha, \beta)}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i$$

$$\frac{\partial L(\mathbf{w}, b, \xi, \alpha, \beta)}{\partial b} = 0 \Rightarrow \sum_{i=1}^n \alpha_i y_i = 0$$

$$\frac{\partial L(\mathbf{w}, b, \xi, \alpha, \beta)}{\partial \xi_i} = 0 \Rightarrow C = \alpha_i + \beta_i$$

Dual Optimization Problem

$$\begin{aligned} \underset{\alpha}{\text{maximize}} \quad & \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ \text{subject to} \quad & \sum_{i=1}^n \alpha_i y_i = 0 \\ & C \geq \alpha_i \geq 0 \quad i = 1, \dots, n \end{aligned}$$

Decision Function

$$f(\mathbf{x}) = \text{sgn}(\langle \mathbf{w}, \mathbf{x} \rangle + b)$$

$$f(\mathbf{x}) = \text{sgn}\left(\sum_{i=1}^n \alpha_i y_i \langle \mathbf{x}_i, \mathbf{x} \rangle + b\right)$$

- Decision function does not change
- Solve this QP with an optimization software to find α_i 's
ILOG CPLEX, MATLAB's quadprog function, ...

Karush-Kuhn-Tucker Theorem

$$\alpha_i [y_i(\langle w, \mathbf{x}_i \rangle + b) - 1 + \xi_i] = 0 \quad i = 1, \dots, n$$

$$\beta_i \xi_i = 0$$

$$\begin{aligned} \alpha_i = C &\Rightarrow y_i(\langle w, \mathbf{x}_i \rangle + b) - 1 + \xi_i = 0 & \xi_i > 0 \\ C > \alpha_i > 0 &\Rightarrow y_i(\langle w, \mathbf{x}_i \rangle + b) - 1 + \xi_i = 0 & \xi_i = 0 \\ \alpha_i = 0 &\Rightarrow y_i(\langle w, \mathbf{x}_i \rangle + b) - 1 + \xi_i > 0 \end{aligned}$$

- \mathbf{x}_i 's with $\alpha_i = C$ make soft error ($\xi_i > 0$) (**bound support vectors**)
- \mathbf{x}_i 's with $C > \alpha_i > 0$ are on separating hyperlanes (**in-bound support vectors**)
- b can be calculated on one of these instances
(Numerically safer to get average on all \mathbf{x}_i 's with $C > \alpha_i > 0$)
- No need to store \mathbf{x}_i 's with $\alpha_i = 0$

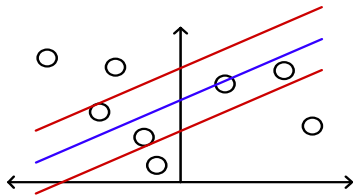
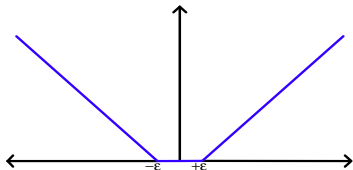
Regression Problem Definition

- Given empirical dataset (\mathbf{X}, \mathbf{Y})
 $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n) \in \mathbb{R}^d \times \mathbb{R}$

- Use a linear model
 $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$

- Use the ϵ -insensitive error function

$$e(y_i, f(\mathbf{x}_i)) = \begin{cases} 0 & \text{if } |y_i - f(\mathbf{x}_i)| \leq \epsilon \\ |y_i - f(\mathbf{x}_i)| - \epsilon & \text{otherwise} \end{cases}$$



Optimization Problem

$$\begin{aligned} \underset{\mathbf{w}, b, \xi^+, \xi^-}{\text{minimize}} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n (\xi_i^+ + \xi_i^-) \\ \text{subject to} \quad & y_i - (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \leq \epsilon + \xi_i^+ \quad i = 1, \dots, n \\ & (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - y_i \leq \epsilon + \xi_i^- \quad i = 1, \dots, n \\ & \xi_i^+ \geq 0 \quad i = 1, \dots, n \\ & \xi_i^- \geq 0 \quad i = 1, \dots, n \end{aligned}$$

Lagrangian Dual

$$\begin{aligned} L(\mathbf{w}, b, \xi^+, \xi^-, \alpha^+, \alpha^-) = & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n (\xi_i^+ + \xi_i^-) - \sum_{i=1}^n (\beta_i^+ \xi_i^+ + \beta_i^- \xi_i^-) \\ & - \sum_{i=1}^n \alpha_i^+ (\epsilon + \xi_i^+ - y_i + \langle \mathbf{w}, \mathbf{x}_i \rangle + b) \\ & - \sum_{i=1}^n \alpha_i^- (\epsilon + \xi_i^- + y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - b) \end{aligned}$$

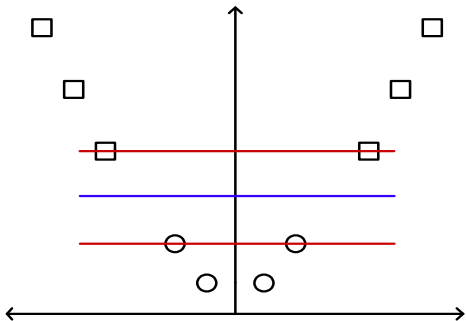
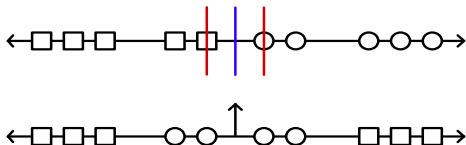
Dual Optimization Problem

$$\begin{aligned} \text{maximize}_{\alpha^+, \alpha^-} \quad & \sum_{i=1}^n y_i (\alpha_i^+ - \alpha_i^-) - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\alpha_i^+ - \alpha_i^-) (\alpha_j^+ - \alpha_j^-) \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ & - \sum_{i=1}^n \epsilon (\alpha_i^+ + \alpha_i^-) \\ \text{subject to} \quad & \sum_{i=1}^n (\alpha_i^+ - \alpha_i^-) = 0 \\ & C \geq \alpha_i^+ \geq 0 \quad i = 1, \dots, n \\ & C \geq \alpha_i^- \geq 0 \quad i = 1, \dots, n \end{aligned}$$

Decision Function

$$\begin{aligned} f(\mathbf{x}) &= \langle \mathbf{w}, \mathbf{x} \rangle + b \\ f(\mathbf{x}) &= \sum_{i=1}^n (\alpha_i^+ - \alpha_i^-) \langle \mathbf{x}_i, \mathbf{x} \rangle + b \end{aligned}$$

Why Do We Need Kernels?



- Define a transformation function (ϕ) from input space to feature space
 $\phi : \mathbf{X} \mapsto \mathbf{H}$
- Map data from input space to feature space
 $\mathbf{x} \mapsto \phi(\mathbf{x})$
 $\langle \mathbf{x}_i, \mathbf{x}_j \rangle \mapsto \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$
- Learn discriminant in feature space
- No need to calculate $\phi(\cdot)$ explicitly. Just replace $\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$ with $K(\mathbf{x}_i, \mathbf{x}_j)$

How Do We Integrate Kernels Into Models?

Embed Kernel Function into Dual Optimization Model and Decision Function

$$\text{maximize}_{\alpha} \quad \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle$$

$$\text{maximize}_{\alpha} \quad \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$$

$$\text{maximize}_{\alpha} \quad \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$






$$f(\mathbf{x}) = \text{sgn}\left(\sum_{i=1}^n \alpha_i y_i \langle \mathbf{x}_i, \mathbf{x} \rangle + b\right)$$

$$f(\mathbf{x}) = \text{sgn}\left(\sum_{i=1}^n \alpha_i y_i \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}) \rangle + b\right)$$

$$f(\mathbf{x}) = \text{sgn}\left(\sum_{i=1}^n \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right)$$

- Advantages
 - Adds non-linearity to linear models
 - Works with non-vectorial data
- Popular kernels
 - Linear Kernel
$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$$
 - Polynomial Kernel
$$K(\mathbf{x}_i, \mathbf{x}_j) = (\langle \mathbf{x}_i, \mathbf{x}_j \rangle + 1)^d$$
 - Gaussian Kernel
$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / \sigma^2)$$
 - Sigmoid Kernel
$$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(2\langle \mathbf{x}_i, \mathbf{x}_j \rangle + 1)$$

Comments on SVMs

-  Finds global minimum (no local minimum)
-  Complexity depends on support vector count not on dimensionality of feature space
-  Avoids over-fitting and works well with small datasets
-  Choice of kernel and its parameters
-  Multi-class classification is an open problem

For Further Reading



Vladimir N. Vapnik

The Nature of Statistical Learning Theory

Springer-Verlag, 1995



Bernhard Schölkopf and Alexander J. Smola

Learning with Kernels

The MIT Press, 2002.



Christopher J. C. Burges

A Tutorial on Support Vector Machines for Pattern Recognition

Data Mining and Knowledge Discovery, 2(2):121–167, 1998.



Alexander J. Smola and Bernhard Schölkopf

A Tutorial on Support Vector Regression

NeuroCOLT2 Technical Report NC2-TR-1998-030, 1998.



Javier M. Moguerza and Alberto Munoz

Support Vector Machines with Applications

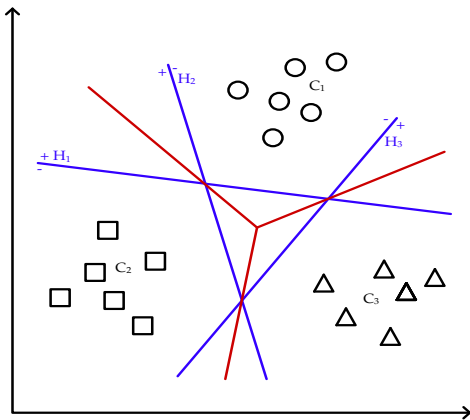
Statistical Science, 21(3):322–336, 2006.

Multi-Class SVMs

- 8 Multi-Machine Approaches
 - One-Versus-All Approach (OVA)
 - All-Versus-All Approach (AVA)

- 9 Single Machine Approaches

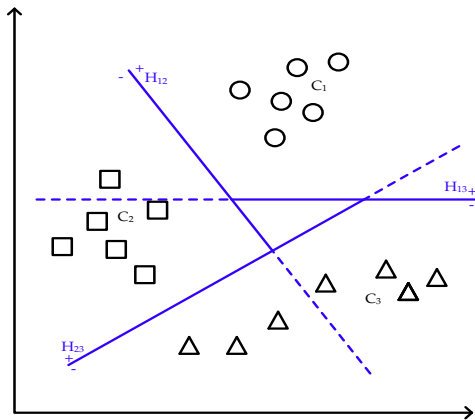
One-Versus-All Approach (OVA)



One-Versus-All Approach (OVA)

- k distinct binary SVM that separates one class from others
- One class has label $+1$, others -1
- $+1$ labeled class of SVM with maximum output value is assigned to test instance
- k optimization problem with n decision variables
- Comparison between SVM outputs may be problematic

All-Versus-All Approach (AVA)



All-Versus-All Approach (AVA)

- $k(k - 1)/2$ distinct binary SVM for each possible pair of classes
- A voting scheme is required for testing
- $k(k - 1)/2$ optimization problem with $2n/k$ decision variables (homogeneous dataset)
- Possible variance increase due to small training set sizes

Single Machine Multi-Class SVM

- More natural way is to consider all classes at once
- Following SVM learns k discriminant together

Optimization Problem

$$\begin{aligned} \underset{\mathbf{w}, b, \xi}{\text{minimize}} \quad & \frac{1}{2} \sum_{m=1}^k \|\mathbf{w}_m\|^2 + C \sum_{i=1}^n \sum_{m \neq y_i} \xi_i^m \\ \text{subject to} \quad & \langle \mathbf{w}_{y_i}, \mathbf{x}_i \rangle + b_{y_i} \geq \langle \mathbf{w}_m, \mathbf{x}_i \rangle + b_m + 2 - \xi_i^m \quad i = 1, \dots, n \quad m \neq y_i \\ & \xi_i^m \geq 0 \quad i = 1, \dots, n \quad m \neq y_i \end{aligned}$$

Decision Function

$$f(\mathbf{x}) = \arg \max_m (\langle \mathbf{w}_m, \mathbf{x} \rangle + b_m)$$

For Further Reading



Vladimir N. Vapnik
Statistical Learning Theory
John Wiley and Sons, 1998



Jason Weston and Chris Watkins
Multi-Class Support Vector Machines
*Technical Report CSD-TR-98-04, Department of Computer Science,
Royal Holloway, University of London, 1998.*



Chih-Wei Hsu and Chih-Jen Lin
A Comparison of Methods for Multi-Class Support Vector Machines
Neural Networks, *IEEE Transactions on*, 13(2):415–425, 2002.



Ryan Rifkin and Aldebaro Klautau
In Defense of One-Vs-All Classification
Journal of Machine Learning Research, 5:101–141, 2004.



Eddy Mayoraz and Ethem Alpaydın
Support Vector Machines for Multi-Class Classification
Engineering Applications of Bio-Inspired Artificial Neural Networks,
833–842, 1999.