

M1 — Apprentissage

Michèle Sebag — Benoit Barbot

LRI — LSV

18 nov. 2013

Hypothesis Space $\mathcal{H}$ / Navigation

	$\mathcal{H}$	navigation operators
Version Space	Logical	spec / gen
Decision Trees	Logical	specialisation
Neural Networks	Numerical	gradient
Support Vector Machines	Numerical	quadratic opt.
Ensemble Methods	—	adaptation $\mathcal{E}$

This course

- ▶ Decision Trees
- ▶ **Support Vector Machines**

$$h : \mathcal{X} = \mathbb{R}^D \mapsto \mathbb{R}$$

Binary classification

$h(\mathbf{x}) > 0 \rightarrow \mathbf{x}$ classified as True

else, classified as False

Overview

Linear SVM, separable case

Linear SVM, non separable case

The kernel trick

- The Kernel principle

- Examples

- Discussion

Extensions

- Multi-class discrimination

- Regression

- Novelty detection

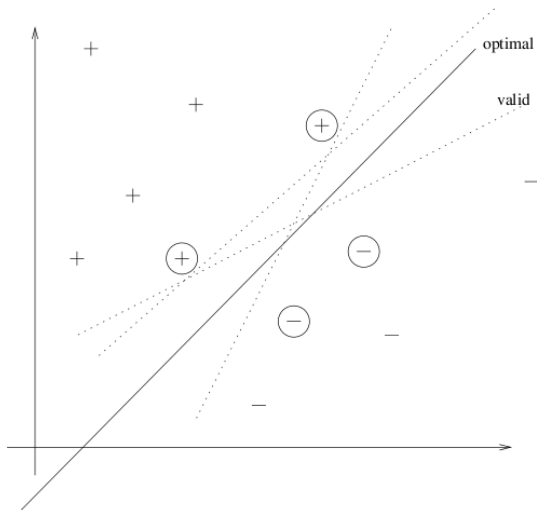
On the practitioner side

- Improve precision

- Reduce computational cost

Theory

The separable case: More than one separating hyperplane



Linear Support Vector Machines

Linear Separators

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

Region $\hat{y} = 1$: $f(\mathbf{x}) > 0$

Region $\hat{y} = -1$: $f(\mathbf{x}) < 0$

Criterion

$$\forall i, y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) > 0$$

Remark

Invariant by multiplication of $\mathbf{w}$ and b by a positive value

Canonical formulation

Fix the scale:

$$\min_i \{y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b)\} = 1$$

$\Leftrightarrow$

$$\forall i, y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1$$

Maximize the Margin

Criterion

Maximize the minimal distance (points, hyperplane).

Obtain the largest possible band

Margin

$$\langle \mathbf{w}, \mathbf{x}_+ \rangle + b = 1 \quad \langle \mathbf{w}, \mathbf{x}_- \rangle + b = -1$$

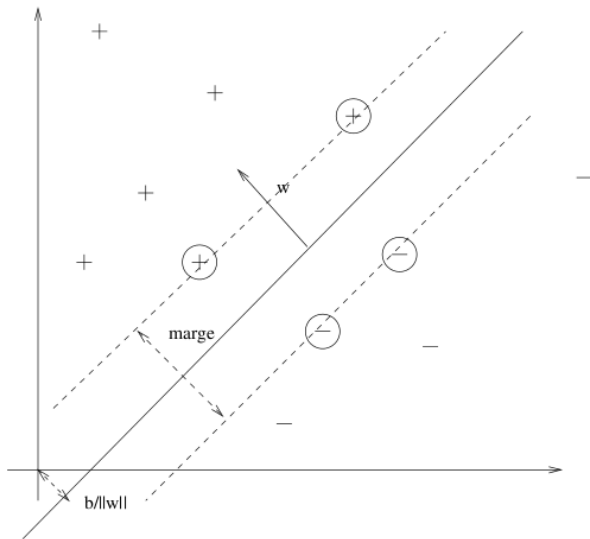
$$\langle \mathbf{w}, \mathbf{x}_+ - \mathbf{x}_- \rangle = 2$$

Margin = projection of $\mathbf{x}_+ - \mathbf{x}_-$ on the normal vector of the hyperplane, $\frac{\mathbf{w}}{\|\mathbf{w}\|_2}$

$$\Rightarrow \text{Maximize } \frac{1}{\|\mathbf{w}\|}$$

$$\Leftrightarrow \text{minimize } \|\mathbf{w}\|^2$$

Maximal Margin Hyperplane



Maximize the Margin (2)

Problem

$$\left\{ \begin{array}{ll} \text{Minimize} & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{with the constraints} & \forall i, y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 \end{array} \right.$$

Reminder: optimisation with constraints

General case

$$\left\{ \begin{array}{ll} \text{Minimize} & f_0(\mathbf{z}) \quad \mathbf{z} \in \text{domain } \mathcal{D} \\ \text{s.t.} & f_i(\mathbf{z}) \leq 0 \quad i = 1 \dots m \\ & h_j(\mathbf{z}) = 0 \quad j = 1 \dots p \end{array} \right. \quad (1)$$

Notation: $\mathbf{z}^*$ the solution of Pb (1)
 $p^* = f_0(\mathbf{z}^*)$

Define the Lagrangian as:

$$L(\mathbf{z}, \lambda, \nu) = f_0(\mathbf{z}) + \sum_{i=1}^m \lambda_i f_i(\mathbf{z}) + \sum_{j=1}^p \nu_j h_j(\mathbf{z})$$

with $\lambda_i \geq 0$

Reminder, 2

Define the Lagrange dual function as:

$$g(\lambda, \nu) = \inf \{ L(\mathbf{z}, \lambda, \nu), \mathbf{z} \in \mathcal{D}, \lambda \geq 0 \} \quad (2)$$

Notation: $\lambda^*, \nu^* = \operatorname{argmin} g(\lambda, \nu)$
 $d^* = g(\lambda^*, \nu^*)$

Intuition

- ▶ λ_i, ν_j are the penalties paid for violating constraints
- ▶ For $\mathbf{z}$ feasible (i.e. $f_i(\mathbf{z}) \leq 0$ and $h_j(\mathbf{z}) = 0$),

$$L(\mathbf{z}, \lambda, \nu) \leq f_0(\mathbf{z})$$

- ▶ Hence $g(\lambda, \nu)$ provides a lower bound on the optimum p^* .
- ▶ L is linear in λ and ν : its minimum is very easy to compute;
- ▶ g is concave in λ and ν even if f_0 , f_i and h_j are not convex in $\mathbf{z}$.

Reminder, 3

With

$$g(\lambda, \nu) = \inf \left\{ f_0(\mathbf{z}) + \sum_{i=1}^m \lambda_i f_i(\mathbf{z}) + \sum_{j=1}^p \nu_j h_j(\mathbf{z}), \mathbf{z} \in \mathcal{D}, \lambda \geq 0 \right\}$$

Then

- ▶ Weak duality (always true)

$$d^* \leq p^*$$

- ▶ Strong duality: when $d^* = p^*$.
usually holds for convex functions f_0 and f_i .

Reminder, 4

Constraint qualifications

= conditions guaranteeing strong duality in convex pbs.

Slater's conditions

if there exists $\mathbf{z}$ s.t. $f_i(\mathbf{z}) < 0$, then strong duality

Remark: if strong duality then

$$\begin{aligned} f_0(\mathbf{z}^*) &= \inf_{\mathbf{z}} \left\{ f_0(\mathbf{z}) + \sum_{i=1}^m \lambda_i^* f_i(\mathbf{z}) + \sum_{j=1}^p \nu_j^* h_j(\mathbf{z}) \right\} \\ &\leq f_0(\mathbf{z}^*) + \sum_{i=1}^m \lambda_i^* f_i(\mathbf{z}^*) + \sum_{j=1}^p \nu_j^* h_j(\mathbf{z}^*) \\ &\leq f_0(\mathbf{z}^*) \end{aligned}$$

Therefore for each $i = 1 \dots m$, $\lambda_i^* f_i(\mathbf{z}^*) = 0$:

if $\lambda_i^* > 0$, $f_i(\mathbf{z}^*) = 0$ (saturated or tight constraint).

if $f_i(\mathbf{z}^*) < 0$ (loose constraint), $\lambda_i^* = 0$.

Reminder, 5

Karun-Kush-Tucker conditions on $\tilde{\mathbf{z}}, \tilde{\lambda}, \tilde{\nu}$

- ▶ Stationarity condition: f_i, h_j differentiable

$$\frac{\partial L(\mathbf{z}, \lambda, \nu)}{\partial \mathbf{z}} \Big|_{\mathbf{z}=\tilde{\mathbf{z}}} = 0$$

- ▶ Slackness condition:

$$\tilde{\lambda}_i f_i(\tilde{\mathbf{z}}) = 0 \text{ for } i = 1 \dots m$$

- ▶ Feasibility conditions

$$f_i(\tilde{\mathbf{z}}) \leq 0; \quad \tilde{\lambda}_i \geq 0$$

Theorem

$\Rightarrow$ If $\mathbf{z}^*$ is optimum of Pb (1), and λ^*, ν^* minimize g , then KKT conditions are satisfied for $(\mathbf{z}^*, \lambda^*, \nu^*)$.

$\Leftarrow$ If f_i convex and h_j affine, then if KKT conditions are satisfied for $\tilde{\mathbf{z}}, \tilde{\lambda}, \tilde{\nu}$, then $\tilde{\mathbf{z}}$ is optimum for Pb (1) and $\tilde{\lambda}, \tilde{\nu}$ minimize g .

Primal Problem

$$\text{Min}_{\mathbf{w}, b} \text{Max}_{\alpha \geq 0} L(\mathbf{w}, b, \alpha) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_i \alpha_i (y_i (\langle \mathbf{x}_i, \mathbf{w} \rangle + b) - 1), \quad \alpha_i \geq 0$$

- Differentiate w.r.t. b : at the optimum,

$$\frac{\partial L}{\partial b} = 0 = \sum \alpha_i y_i$$

- Differentiate w.r.t. $\mathbf{w}$:

$$\frac{\partial L}{\partial \mathbf{w}} = 0 = \mathbf{w} - \sum \alpha_i y_i \mathbf{x}_i$$

- Replace in $L(\mathbf{w}, b, \alpha)$:

Dual problem (Wolfe)

$$\left\{ \begin{array}{l} \text{Maximize} \\ \text{with the constraint} \end{array} \right. \quad \begin{array}{l} W(\alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ \forall i, \alpha_i \geq 0 \\ \sum_i \alpha_i y_i = 0 \end{array}$$

Quadratic form w.r.t. α

quadratic optimization is easy

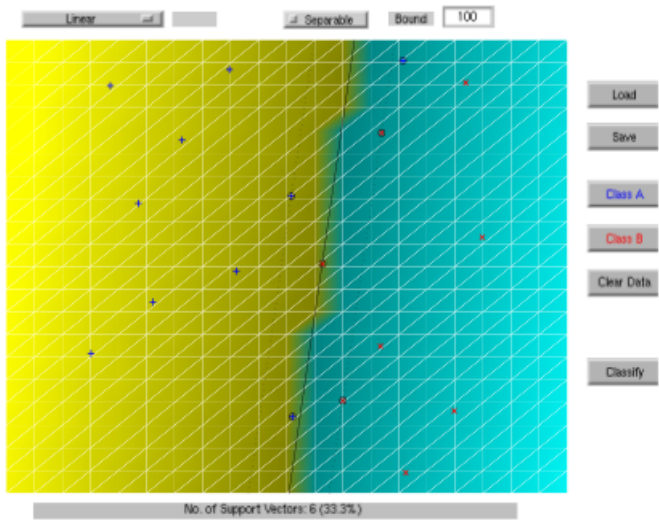
Solution: α_i^*

- Compute $\mathbf{w}^*$:

$$\mathbf{w}^* = \sum_i \alpha_i^* y_i \mathbf{x}_i$$

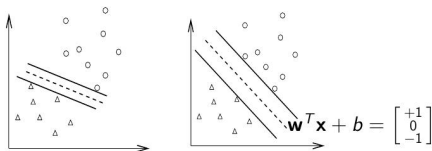
- If $(\langle \mathbf{x}_i, \mathbf{w}^* \rangle + b) y_i > 1$, $\alpha_i^* = 0$.
- If $\alpha_i^* > 0$, then $(\langle \mathbf{x}_i, \mathbf{w}^* \rangle + b) y_i = 1$ $\mathbf{x}_i$ **support vector**
- Compute b^* :

$$b^* = -\frac{1}{2}(\langle \mathbf{w}^*, \bar{\mathbf{x}}^+ \rangle + \langle \mathbf{w}^*, \bar{\mathbf{x}}^- \rangle)$$



Summary

$$\mathcal{E} = \{(\mathbf{x}_i, y_i)\}, \mathbf{x}_i \in \mathbb{R}^d, y_i \in \{-1, 1\}, i = 1..n \quad (\mathbf{x}_i, y_i) \sim P(\mathbf{x}, y)$$



$$h(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$$

Two goals

Role

- ▶ Data fitting
 $\text{sign}(y_i) = \text{sign}(h(\mathbf{x}_i)) \rightarrow \text{maximize margin } y_i \cdot h(\mathbf{x}_i)$

achieve learning

- ▶ Regularization : minimize $\|\mathbf{w}\|$

avoid overfitting

Support Vector Machines

General scheme

- ▶ Minimize the regularization term
- ▶ ... subject to data constraints $\quad \quad \quad = \text{margin} \geq 1 \text{ (*)}$

$$\begin{cases} \text{Min.} & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.t.} & y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 \quad \forall i = 1 \dots n \end{cases}$$

Constrained minimization of a convex function

→ introduce Lagrange multipliers $\alpha_i \geq 0, i = 1 \dots n$

$$\text{Min } \mathcal{L}(\mathbf{w}, b, \alpha) = \frac{1}{2} \|\mathbf{w}\|^2 + \sum_i \alpha_i (1 - y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b))$$

Primal problem

- ▶ $d + 1$ variables (+ n Lagrange multipliers)

(*) in the separable case; see later for non-separable case

Support Vector Machines, 2

At the optimum

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \frac{\partial \mathcal{L}}{\partial b} = \frac{\partial \mathcal{L}}{\partial \alpha} = 0$$

Dual problem

Wolfe

$$\left\{ \begin{array}{ll} \text{Max.} & \mathcal{Q}(\alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ \text{s.t.} & \forall i, \alpha_i \geq 0 \\ & \sum_i \alpha_i y_i = 0 \end{array} \right.$$

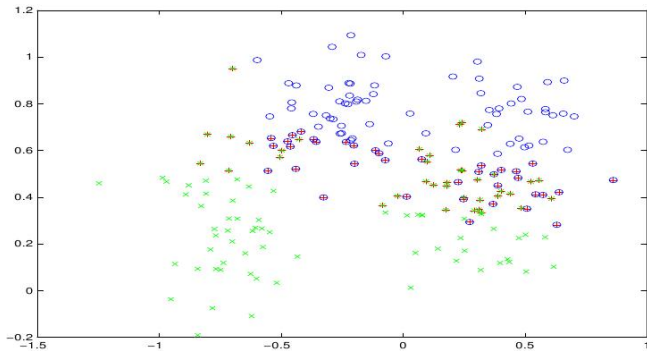
Support vectors

Examples $(\mathbf{x}_i, y_i)$ s.t. $\alpha_i > 0$

the only ones involved in the decision function

$$\mathbf{w} = \sum \alpha_i y_i \mathbf{x}_i$$

Support vectors, examples



Support vectors, examples

MNIST data



Data



Support vectors

Remarks

- ▶ Support vectors are critical examples near-miss
- ▶ Show that the Leave-One-Out error is less than $\#$ sv.

LOO: iteratively, learn on all examples but one, and test on the remaining one

Overview

Linear SVM, separable case

Linear SVM, non separable case

The kernel trick

- The Kernel principle

- Examples

- Discussion

Extensions

- Multi-class discrimination

- Regression

- Novelty detection

On the practitioner side

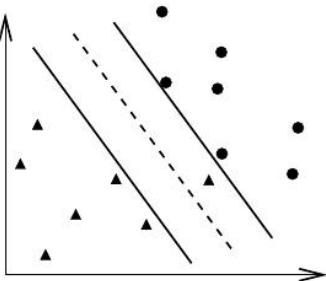
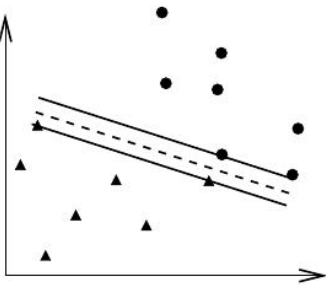
- Improve precision

- Reduce computational cost

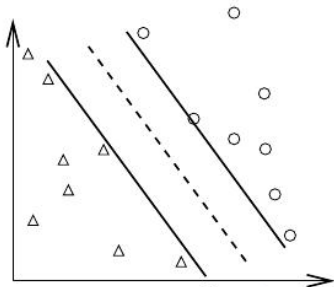
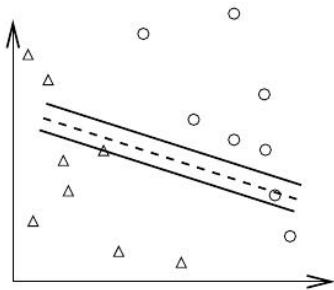
Theory

Separable vs non-separable data

Training



Test



Linear hypotheses, non separable data

Cortes & Vapnik 95

Non-separable data $\Rightarrow$ not all constraints are satisfiable

$$y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i$$

Formalization

- ▶ Introduce slack variables ξ_i
- ▶ And penalize them

$$\left\{ \begin{array}{ll} \text{Minimize} & \frac{1}{2} \|\mathbf{w}\|^2 + \mathbf{C} \sum_i \xi_i \\ \text{Subject to} & \forall i, y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{array} \right.$$

Critical decision: adjust C = error cost.

Primal problem, non separable case

Same resolution: Lagrange Multipliers α_i and β_i , with $\alpha_i \geq 0$, $\beta_i \geq 0$

$$\begin{aligned}\mathcal{L}(\mathbf{w}, b, \xi, \alpha, \beta) = & \text{Min}_{\mathbf{w}, b, \xi} \text{Max}_{\alpha, \beta} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_i \xi_i \\ & - \sum_i \alpha_i (y_i (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - 1 + \xi_i) \\ & - \sum_i \beta_i \xi_i\end{aligned}$$

At the optimum

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \frac{\partial \mathcal{L}}{\partial b} = \frac{\partial \mathcal{L}}{\partial \xi_i} = 0$$

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i \quad \sum_i \alpha_i y_i = 0 \quad C - \alpha_i - \beta_i = 0$$

Likewise

- Convex (quadratic) optimization problem $\rightarrow$ it is equivalent to solve the primal and the dual problem (expressed with multipliers α, β)

Dual problem, non separable case

$$\text{Min} \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle, \quad 0 \leq \alpha_i \leq C$$

Mathematically nice problem

- ▶ H = semi-definite positive $n \times n$ matrix

$$H_{i,j} = y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle$$

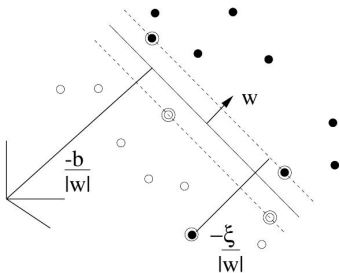
- ▶ Dual problem

quadratic form

$$\text{Minimize } \langle \alpha, e \rangle - \alpha^T H \alpha$$

with $e = (1, \dots, 1) \in \mathbb{R}^n$.

Support vectors



- ▶ Only support vectors ($\alpha_i > 0$) are involved in h

$$\mathbf{w} = \sum \alpha_i y_i \mathbf{x}_i$$

- ▶ Importance of support vector $\mathbf{x}_i$: weight α_i
- ▶ Difference with the separable case $0 < \alpha_i < C$
bounded influence of examples

The loss (error cost) function

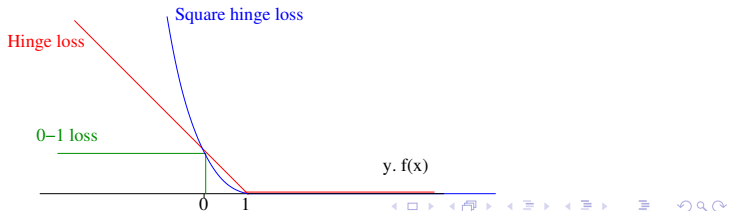
Roles

- ▶ The goal is data fitting
loss function characterizes the learning goal
- ▶ while solving a convex optimization problem
and makes it tractable/reproducible

The error cost

- ▶ Binary cost: $\ell(y, h(\mathbf{x})) = 1$ iff $y \neq h(\mathbf{x})$
- ▶ Quadratic cost: $\ell(y, h(\mathbf{x})) = (y - h(\mathbf{x}))^2$
- ▶ Hinge loss

$$\ell(y, h(\mathbf{x})) = \max(0, 1 - y \cdot h(\mathbf{x})) = (1 - y \cdot h(\mathbf{x}))_+ = \xi$$



Complexity

Learning complexity

- ▶ Worst case: $\mathcal{O}(n^3)$
- ▶ Empirical complexity: depends on C
- ▶ $\mathcal{O}(n^2 n_{sv})$ where n_{sv} is the number of s.v.

Usage complexity

- ▶ $\mathcal{O}(n_{sv})$

Overview

Linear SVM, separable case

Linear SVM, non separable case

The kernel trick

- The Kernel principle

- Examples

- Discussion

Extensions

- Multi-class discrimination

- Regression

- Novelty detection

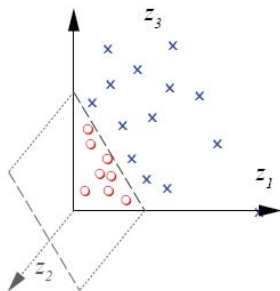
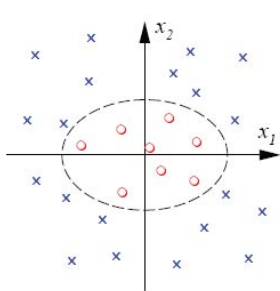
On the practitioner side

- Improve precision

- Reduce computational cost

Theory

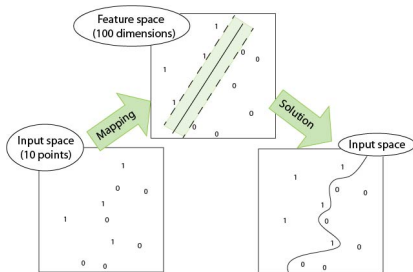
Non-separable data



Representation change

$$\mathbf{x} \in \mathbb{R}^2 \rightarrow \text{polar coordinates} \in \mathbb{R}^2$$

Principle



$$\Phi : X \mapsto \Phi(X) \subset \mathbb{R}^D$$

Intuition

- ▶ In a high-dimensional space, every dataset is linearly separable
→ Map data onto $\Phi(X)$, and we are back to linear separation

Glossary

- ▶ X : input space
- ▶ $\Phi(X)$: feature space

The kernel trick

Remark

- ▶ Generalization bounds do not depend on the dimension of input space X but on the capacity of the hypothesis space $\mathcal{H}$.
- ▶ SVMs only involve scalar products $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$.

Intuition

- ▶ Representation change is only “virtual” $\Phi : X \mapsto \Phi(X)$
- ▶ Consider scalar product in $\Phi(X)$
- ▶ ... and compute it in X

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

Example: polynomial kernel

Principle

$$\mathbf{x} \in \mathbb{R}^3 \mapsto \Phi(\mathbf{x}) \in \mathbb{R}^{10}$$

$$\mathbf{x} = (x_1, x_2, x_3)$$

$$\Phi(\mathbf{x}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_3, \sqrt{2}x_1x_2, \sqrt{2}x_1x_3, \sqrt{2}x_2x_3, x_1^2, x_2^2, x_3^2)$$

Why $\sqrt{2}$?

Example: polynomial kernel

Principle

$$\mathbf{x} \in \mathbb{R}^3 \mapsto \Phi(\mathbf{x}) \in \mathbb{R}^{10}$$

$$\mathbf{x} = (x_1, x_2, x_3)$$

$$\Phi(\mathbf{x}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_3, \sqrt{2}x_1x_2, \sqrt{2}x_1x_3, \sqrt{2}x_2x_3, x_1^2, x_2^2, x_3^2)$$

Why $\sqrt{2}$?

because

$$\langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle = (1 + \langle \mathbf{x}, \mathbf{x}' \rangle)^2 = K(\mathbf{x}, \mathbf{x}')$$

Primal and dual problems unchanged

Primal problem

$$\begin{cases} \text{Min.} & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.t.} & y_i(\langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle + b) \geq 1 \quad \forall i = 1 \dots n \end{cases}$$

Dual problem

$$\begin{cases} \text{Max.} & \mathcal{Q}(\alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{s.t.} & \forall i, \alpha_i \geq 0 \\ & \sum_i \alpha_i y_i = 0 \end{cases}$$

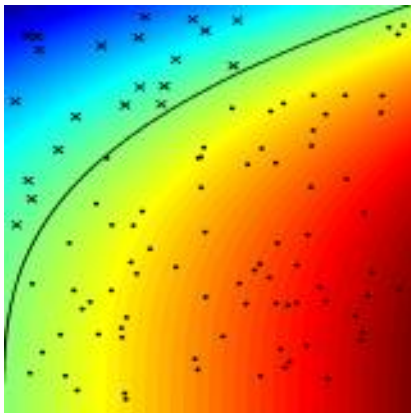
Hypothesis

$$h(\mathbf{x}) = \sum_i \alpha_i y_i K(\mathbf{x}_i, \mathbf{x})$$

Example, polynomial kernel

$$K(\mathbf{x}, \mathbf{x}') = (a\langle \mathbf{x}, \mathbf{x}' \rangle + 1)^b$$

- ▶ Choice of a, b : cross validation
- ▶ Domination of high/low degree terms ?
- ▶ Importance of normalization



Example, Radius-Based Function kernel (RBF)

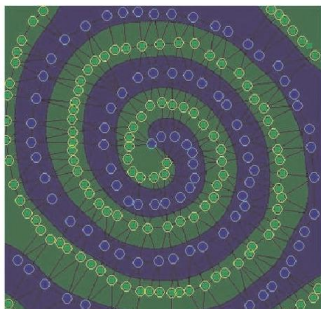
$$K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2)$$

- ▶ No closed form Φ
- ▶ $\Phi(X)$ of infinite dimension

For x in $\mathbb{R}$

$$\Phi(x) = \exp(-\gamma x^2) \left[1, \sqrt{\frac{2\gamma}{1!}}x, \sqrt{\frac{(2\gamma)^2}{2!}}x^2, \sqrt{\frac{(2\gamma)^3}{3!}}x^3, \dots \right]$$

- ▶ Choice of γ ? (intuition: think of H , $H_{i,j} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$)



String kernels

Watkins 99, Lodhi 02 **Notations**

- ▶ s a string on alphabet Σ
- ▶ $\mathbf{i} = (i_1, i_2, \dots, i_n)$ an ordered index sequence ($i_j < i_{j+1}$), avec $\ell(\mathbf{i}) = i_n - i_1 + 1$
- ▶ $s[\mathbf{i}]$ substring of s , extraction pattern is $\mathbf{i}$
 $s = \text{BICYCLE}$, $\mathbf{i} = (1, 3, 6)$, $s[\mathbf{i}] = \text{BCL}$

Definition

$$K_n(s, s') = \sum_{u \in \Sigma^n} \sum_{\mathbf{i} \text{ s.t. } s[\mathbf{i}] = u} \sum_{\mathbf{j} \text{ s.t. } s'[\mathbf{j}] = u} \varepsilon^{\ell(\mathbf{i}) + \ell(\mathbf{j})}$$

with $0 < \varepsilon < 1$ (discount)

String kernels, followed

Φ : projection on $\mathbb{R}^D$ où $D = |\Sigma|^n$

	CH	CA	CT	AT
CHAT	ε^2	ε^3	ε^4	ε^2
CARTOON	0	ε^2	ε^4	ε^3

$$K(\text{CHAT}, \text{CARTON}) = 2\varepsilon^5 + \varepsilon^8$$

Prefer the normalized version

$$\kappa(s, s') = \frac{K(s, s')}{\sqrt{K(s, s)K(s', s')}}}$$

String kernels, followed

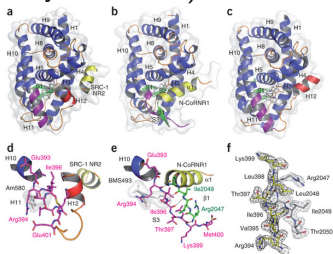
Application 1

Document mining

- ▶ Pre-processing matters a lot (stop-words, stemming)
- ▶ Multi-lingual aspects
- ▶ Document classification
- ▶ Information retrieval

Application 2, Bio-informatics

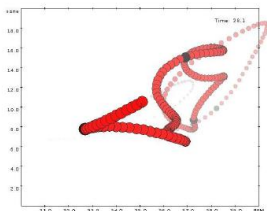
- ▶ Pre-processing matters a lot
- ▶ Classification (secondary structures)



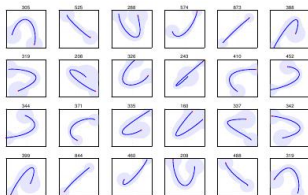
Extension to graph kernels http://videlectures.net/gbr07_vert_ckac/

Application to musical analysis

- ▶ Input: Midi files
- ▶ Pre-processing, rhythm detection
- ▶ Representation: the musical worm (tempo, loudness)
- ▶ Output: Identification of performer styles



(a)



(b)

Using String Kernels to Identify Famous Performers from their Playing Style, Saunders et al., 2004

Kernels: key features

Absolute $\rightarrow$ Relative representation

- ▶ $\langle \mathbf{x}, \mathbf{x}' \rangle \propto$ angle of $\mathbf{x}$ and $\mathbf{x}'$
- ▶ More generally $K(\mathbf{x}, \mathbf{x}')$ measures the (non-linear) similarity of $\mathbf{x}$ and $\mathbf{x}'$
- ▶ $\mathbf{x}$ is described by its similarity to other examples

Necessary condition: the Mercer condition

K must be positive semi-definite

$$\forall g \in L_2, \int K(\mathbf{x}, \mathbf{x}') g(\mathbf{x}) g(\mathbf{x}') d\mathbf{x} \geq 0$$

Why ?

Related to Φ Mercer condition holds $\rightarrow \exists \phi_1, \phi_2, ..$

$$k(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{\infty} \lambda_i \phi_i(\mathbf{x}) \phi_i(\mathbf{x}')$$

with ϕ_i eigen functions, $\lambda_i > 0$ eigen values

Kernel properties: let K, K' be p.d. kernels and $\alpha > 0$, then

- ▶ αK is a p.d. kernel
- ▶ $K + K'$ is a p.d. kernel
- ▶ $K.K'$ is a p.d. kernel
- ▶ $K(\mathbf{x}, \mathbf{x}') = \lim_{p \rightarrow \infty} K_p(\mathbf{x}, \mathbf{x}')$ is p.d. if it exists
- ▶ $K(A, B) = \sum_{\mathbf{x} \in A, \mathbf{x}' \in B} K(\mathbf{x}, \mathbf{x}')$ is a p.d. kernel

Overview

Linear SVM, separable case

Linear SVM, non separable case

The kernel trick

The Kernel principle

Examples

Discussion

Extensions

Multi-class discrimination

Regression

Novelty detection

On the practitioner side

Improve precision

Reduce computational cost

Theory

Multi-class discrimination

Input

Binary case

$$\mathcal{E} = \{(\mathbf{x}_i, y_i)\}, \mathbf{x}_i \in \mathbb{R}^d, y_i \in \{-1, 1\}, i = 1..n \quad (\mathbf{x}_i, y_i) \sim P(\mathbf{x}, y)$$

Multi-class case

$$\mathcal{E} = \{(\mathbf{x}_i, y_i)\}, \mathbf{x}_i \in \mathbb{R}^d, y_i \in \{1 \dots k\}, i = 1..n \quad (\mathbf{x}_i, y_i) \sim P(\mathbf{x}, y)$$

Output : $\hat{h} : \mathbb{R}^d \mapsto \{1 \dots k\}$.

Multi-class learning: one against all

First option: k binary learning problems

Pb 1: class 1 $\rightarrow +1$, classes $2 \dots k \rightarrow -1$

h_1

Pb 2: class 2 $\rightarrow +1$, classes $1, 3, \dots k \rightarrow -1$

h_2

...

Prediction

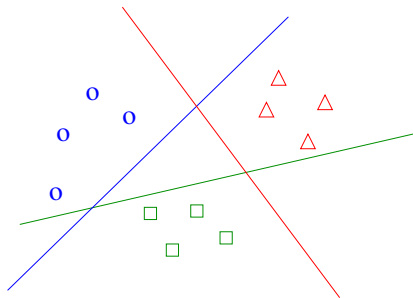
$$h(\mathbf{x}) = i \text{ iff } h_i(\mathbf{x}) = \operatorname{argmax}\{h_j(\mathbf{x}), j = 1 \dots k\}$$

Justification

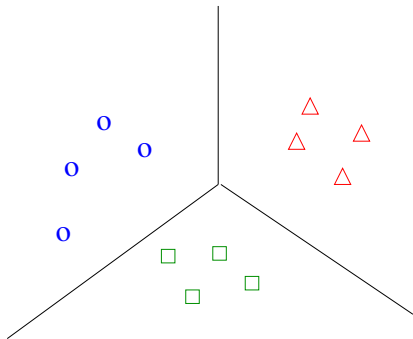
If $\mathbf{x}$ belongs to class 1, one should have

$$h_1(\mathbf{x}) \geq 1, h_j(\mathbf{x}) < -1, j \neq 1$$

Where is the difficulty ?



What we get (one vs all)



What we want

Multi-class learning: one vs one

Second option: $\frac{k(k-1)}{2}$ binary classification problems

Pb i, j class $i \rightarrow +1$, class $j \rightarrow -1$

$h_{i,j}$

Prediction

- ▶ Compute all $h_{i,j}(\mathbf{x})$
- ▶ Count the votes

Classes		winner					
1	2	1					
1	3	1					
1	4	1					
2	3	2					
2	4	4					
3	4	3					
			class	1	2	3	4
			# votes	3	1	1	1

NB: One can also use the $h_{i,j}(\mathbf{x})$ values.

Multi-class learning: additional constraints

Another option

Vapnik 98; Weston, Watkins 99

$$\left\{ \begin{array}{l} \text{Minimise} \quad \frac{1}{2} \sum_{j=1}^k \|\mathbf{w}_j\|^2 + C \sum_{i=1}^n \sum_{\ell=1, \ell \neq y_i}^k \xi_{i,\ell} \\ \text{Subject to} \quad \forall i, \forall \ell \neq y_i, \\ \quad (\langle \mathbf{w}_{y_i}, \mathbf{x}_i \rangle + b_{y_i}) \geq (\langle \mathbf{w}_\ell, \mathbf{x}_i \rangle + b_\ell) + 2 - \xi_{i,\ell} \\ \quad \xi_{i,\ell} \geq 0 \end{array} \right.$$

Hum !

- ▶ $n \times k$ constraints: $n \times k$ dual variables

Recommendations

In practice

- ▶ Results are in general (but not always !) similar
- ▶ 1-vs-1 is the fastest option

Overview

Linear SVM, separable case

Linear SVM, non separable case

The kernel trick

- The Kernel principle

- Examples

- Discussion

Extensions

- Multi-class discrimination

- Regression

- Novelty detection

On the practitioner side

- Improve precision

- Reduce computational cost

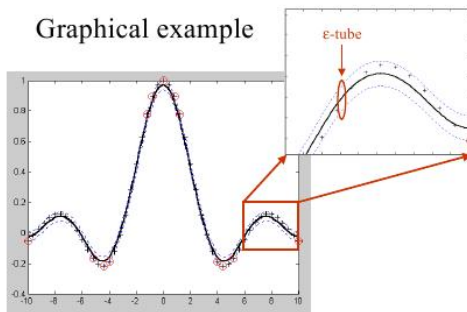
Theory

Regression

Input

$$\mathcal{E} = \{(x_i, y_i)\}, \quad x_i \in \mathbb{R}^d, \quad y_i \in \mathbb{R}, i = 1..n \quad (x_i, y_i) \sim P(x, y)$$

Output : $\hat{h} : \mathbb{R}^d \mapsto \mathbb{R}$.



Regression with Support Vector Machines

Intuition

- ▶ Find h deviating by at most ε from the data
- ▶ ... while being as flat as possible

loss function

regularization

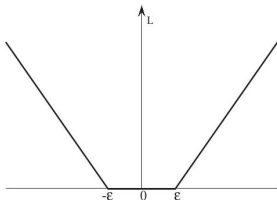
Formulation

$$\left\{ \begin{array}{ll} \text{Min.} & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.t.} & \forall i = 1 \dots n \\ & (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq y_i - \varepsilon \\ & (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \leq y_i + \varepsilon \end{array} \right.$$

Regression with Support Vector Machines, followed

Using slack variables

$$\left\{ \begin{array}{ll} \text{Min.} & \frac{1}{2} \|\mathbf{w}\|^2 + \mathbf{C} \sum_i (\xi_i^+ + \xi_i^-) \\ \text{s.t.} & \forall i = 1 \dots n \\ & (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq y_i - \varepsilon - \xi_i^- \\ & (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \leq y_i + \varepsilon + \xi_i^+ \end{array} \right.$$



Regression with Support Vector Machines, followed

Primal problem

$$\begin{aligned}\mathcal{L}(\mathbf{w}, b, \xi, \alpha, \beta) = & \text{Min } \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_i (\xi_i^+ + \xi_i^-) \\ & - \sum_i \alpha_i^+ (y_i + \varepsilon + \xi_i^+ - \langle \mathbf{w}, \mathbf{x}_i \rangle + b) \\ & - \sum_i \alpha_i^- (\langle \mathbf{w}, \mathbf{x}_i \rangle + b - y_i + \varepsilon + \xi_i^-) \\ & - \sum_i \beta_i^+ \xi_i^+ - \sum_i \beta_i^- \xi_i^-\end{aligned}$$

Dual problem

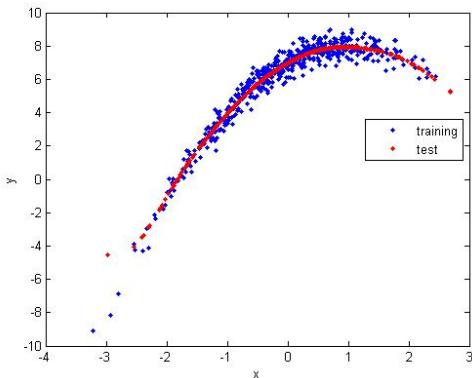
$$\left\{ \begin{array}{l} \mathcal{Q}(\alpha^+, \alpha^-) = \sum_i y_i (\alpha_i^+ - \alpha_i^-) - \varepsilon \sum_i (\alpha_i^+ + \alpha_i^-) \\ \quad + \sum_{i,j} (\alpha_i^+ - \alpha_i^-) (\alpha_j^+ - \alpha_j^-) \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ \text{s.t.} \quad \forall i = 1 \dots n \\ \quad \sum (\alpha_i^+ - \alpha_i^-) = 0 \\ \quad 0 \leq \alpha_i^+ \leq C \\ \quad 0 \leq \alpha_i^- \leq C \end{array} \right.$$

Regression with Support Vector Machines, followed Hypothesis

$$h(\mathbf{x}) = \sum (\alpha_i^+ - \alpha_i^-) \langle \mathbf{x}_i, \mathbf{x} \rangle + b$$

With no loss of generality you can replace everywhere

$$\langle \mathbf{x}, \mathbf{x}' \rangle \rightarrow K(\mathbf{x}, \mathbf{x}')$$



Beware

High-dimensional regression

$$\mathcal{E} = \{(\mathbf{x}_i, y_i)\}, \mathbf{x}_i \in \mathbb{R}^D, y_i \in \mathbb{R}, i = 1..n \quad (\mathbf{x}_i, y_i) \sim P(\mathbf{x}, y)$$

A very slippery game if $D \gg n$ curse of dimensionality

Dimensionality reduction mandatory

- ▶ Map $\mathbf{x}$ onto $\mathbb{R}^d$
- ▶ Central subspace:

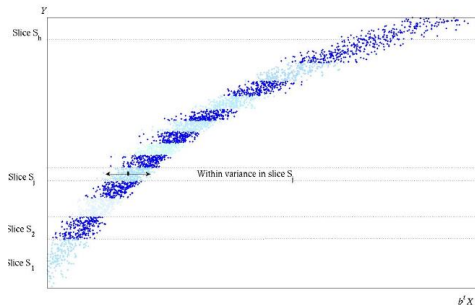
$$\pi : X \mapsto S \subset \mathbb{R}^d$$

with S minimal such that y and $\mathbf{x}$ are independent conditionally to $\pi(x)$.

Find $h, \mathbf{w} : y = h(\mathbf{w}, \mathbf{x})$

Sliced Inverse Regression

Bernard-Michel et al, 09



More:

<http://mistis.inrialpes.fr/learninria/>
S. Girard

Overview

Linear SVM, separable case

Linear SVM, non separable case

The kernel trick

- The Kernel principle

- Examples

- Discussion

Extensions

- Multi-class discrimination

- Regression

- Novelty detection

On the practitioner side

- Improve precision

- Reduce computational cost

Theory

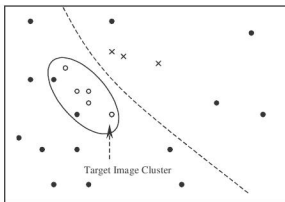
Novelty Detection

Input

$$\mathcal{E} = \{(x_i)\}, x_i \in X, i = 1..n \quad (x_i) \sim P(x)$$

Context

- Information retrieval



- Identification of the data support

estimation of distribution

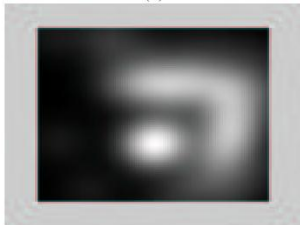
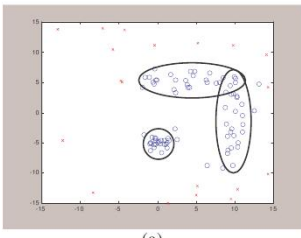
Critical issue

- Classification approaches not efficient: too much noise

One-class SVM

Formulation

$$\left\{ \begin{array}{ll} \text{Min.} & \frac{1}{2} \|\mathbf{w}\|^2 - \rho + \mathbf{C} \sum_i \xi_i \\ \text{s.t.} & \forall i = 1 \dots n \\ & \langle \mathbf{w}, \mathbf{x}_i \rangle \geq \rho - \xi_i \end{array} \right.$$



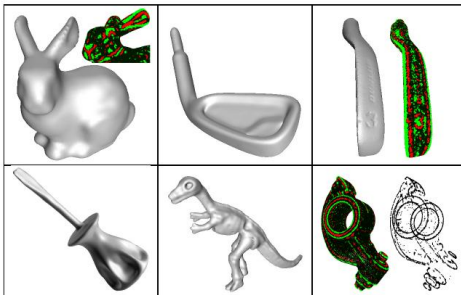
Dual problem

$$\left\{ \begin{array}{ll} \text{Min.} & \sum_{i,j} \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ \text{s.t.} & \forall i = 1 \dots n \quad 0 \leq \alpha_i \leq C \\ & \sum_i \alpha_i = 0 \end{array} \right.$$

Implicit surface modelling

Schoelkopf et al, 04 **Goal:** find the surface formed by the data points

$$\langle \mathbf{w}, \mathbf{x}_i \rangle \geq \rho \text{ becomes } -\varepsilon \leq (\langle \mathbf{w}, \mathbf{x}_i \rangle - \rho) \leq \varepsilon$$



Overview

Linear SVM, separable case

Linear SVM, non separable case

The kernel trick

- The Kernel principle

- Examples

- Discussion

Extensions

- Multi-class discrimination

- Regression

- Novelty detection

On the practitioner side

- Improve precision

- Reduce computational cost

Theory

Normalisation / Scaling

Needed to prevent attributes to steal the game

	Height	Gender	Class
x_1	150	F	1
x_2	180	M	0
x_3	185	M	0

Δ
 x_1

$\begin{matrix} 0 & 0 \\ x_2 & x_3 \end{matrix}$

⇒ Normalization

$$\text{Height} \rightarrow \frac{\text{Height} - 150}{180 - 150}$$

Beware

Usual practice

- ▶ Normalize the whole dataset
- ▶ Learn on the training set
- ▶ Test on the test set

Beware

Usual practice

- ▶ Normalize the whole dataset
- ▶ Learn on the training set
- ▶ Test on the test set

NO!

Good practice

- ▶ Normalize the training set ($\text{Scale}_{\text{train}}$)
- ▶ Learn from the normalized training set
- ▶ Scale the test set according to $\text{Scale}_{\text{train}}$ and test

Imbalanced datasets

Typically

- ▶ Normal transactions: 99.99%
- ▶ Fraudulous transactions: not many

Practice

- ▶ Define asymmetrical penalizations

std penalization

$$C \sum_i \xi_i$$

asymmetrical penalizations

$$C_+ \sum_{i, y_i=1} \xi_i + C_- \sum_{i, y_i=-1} \xi_i$$

Other options ?

Overview

Linear SVM, separable case

Linear SVM, non separable case

The kernel trick

- The Kernel principle

- Examples

- Discussion

Extensions

- Multi-class discrimination

- Regression

- Novelty detection

On the practitioner side

- Improve precision

- Reduce computational cost

Theory

Data sampling

Simple approaches

- ▶ Uniform sampling often efficient
- ▶ Stratified sampling same distribution as in $\mathcal{E}$

Incremental approaches

Syed et al. 99

- ▶ Partition $\mathcal{E} \rightarrow \mathcal{E}_1, \dots, \mathcal{E}_N$
- ▶ Learn from $\mathcal{E}_1 \rightarrow$ support vectors SV_1
- ▶ Learn from $\mathcal{E}_2 \cup SV_1 \rightarrow$ support vectors SV_2
- ▶ etc.

Data sampling, followed

Select examples

Bakir 2005

- ▶ Use k -nearest neighbors
- ▶ Train SVM on k -means (prototypes)
- ▶ Pb about distances

Hierarchical methods

Yu 2003

- ▶ Use unsupervised learning and form clusters
learning, J. Gama
- ▶ Learn a hypothesis on each cluster
- ▶ Aggregate hypotheses

Unsupervised

Reduce number of variables

Select candidate s.v. $\mathcal{F} \subset \mathcal{E}$

$$w = \sum \alpha_i y_i \mathbf{x}_i \quad \text{with } (\mathbf{x}_i, y_i) \in \mathcal{F}$$

Optimize α_i on $\mathcal{E}$

$$\left\{ \begin{array}{ll} \text{Min.} & \frac{1}{2} \sum_{i,j \in \mathcal{F}} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle + C \sum_{\ell=1}^n \xi_{\ell} \\ \text{t.q.} & \forall \ell = 1 \dots n, \\ & (\langle w, \mathbf{x}_{\ell} \rangle + b) \geq 1 - \xi_{\ell} \\ & \xi_{\ell} \geq 0 \end{array} \right.$$

Sources

- ▶ Vapnik, The nature of statistical learning, Springer Verlag 1995; Statistical Learning Theory, Wiley 1998
- ▶ Cristianini & Shawe Taylor, An introduction to Support Vector Machines, Cambridge University Press, 2000.
- ▶ <http://www.kernel-machines.org/tutorials>
- ▶ Videlectures + ML Summer Schools
- ▶ Large scale Machine Learning challenge, ICML 2008 wshop:
<http://largescale.ml.tu-berlin.de/workshop/>

Overview

- Linear SVM, separable case

- Linear SVM, non separable case

- The kernel trick

 - The Kernel principle

 - Examples

 - Discussion

- Extensions

 - Multi-class discrimination

 - Regression

 - Novelty detection

- On the practitioner side

 - Improve precision

 - Reduce computational cost

- Theory

Reminder



Vapnik, 1995, 1998

Input

$\mathcal{E} = \{(x_i, y_i)\}, x_i \in \mathbb{R}^m, y_i \in \{-1, 1\}, i = 1..n\} \quad (x_i, y_i) \sim P(x, y)$

Output : $\hat{h} : \mathbb{R}^m \mapsto \{-1, 1\}$ ou $\mathbb{R}$. $\hat{h}$ approximates y

Criterion : ideally, minimize the generalization error

$$Err(h) = \int \ell(y, \hat{h}(x)) dP(x, y)$$

ℓ = loss function: $1_{y \neq \hat{h}(x)}, (y - \hat{h}(x))^2$

$P(x, y)$ = joint distribution of the data.

The Bias-Variance Tradeoff

Choice of a model: The space $\mathcal{H}$ where we are looking for $\hat{h}$.

Bias: Distance between y and $h^* = \operatorname{argmin}\{Err(h), h \in \mathcal{H}\}$.
the best we can hope for

Variance: Distance between $\hat{h}$ and h^*
between the best h^* and the $\hat{h}$ we actually learn

Note :

Only the empirical risk (on the available data) is given

$$Err_{emp,n}(\hat{h}) = \frac{1}{n} \sum_{i=1}^n \ell(y_i, \hat{h}(x_i))$$

Principle:

$$Err(\hat{h}) < Err_{emp,n}(\hat{h}) + \mathcal{B}(n, \mathcal{H})$$

If $\mathcal{H}$ is “reasonable”, $Err_{emp,n} \rightarrow Err$ when $n \rightarrow \infty$

Statistical Learning

Statistical Learning Theory

Learning from a statistical perspective.

Goal of the theory

in general

Model a real / artificial phenomenon, in order to:

- * understand
- * predict
- * exploit

General

A theory: hypotheses $\rightarrow$ predictions

- ▶ Hypotheses on the phenomenon
- ▶ Predictions about its behavior

here, Learning
errors

Theory $\rightarrow$ algorithm

- ▶ Optimize the quantities allowing prediction
- ▶ Nothing practical like a good theory!

Vapnik

General

A theory: hypotheses $\rightarrow$ predictions

- ▶ Hypotheses on the phenomenon
- ▶ Predictions about its behavior

here, Learning
errors

Theory $\rightarrow$ algorithm

- ▶ Optimize the quantities allowing prediction
- ▶ Nothing practical like a good theory!

Vapnik

Strength/Weaknesses

+ Stronger Hypotheses $\rightarrow$ more precise predictions

BUT if the hypotheses are wrong, nothing will work

What Theory do we need?

Approach in expectation

one example
breast cancer

- ▶ A set of data
- ▶ $\bar{x}^+$: average of positive examples
- ▶ $\bar{x}^-$: average of negative examples
- ▶ $h(x) = +1$ iff $d(x, \bar{x}^+) < d(x, \bar{x}^-)$

Estimate the generalization error

- ▶ Data $\rightarrow$ Training set, test set
- ▶ Learn $\bar{x}^+$ et $\bar{x}^-$ on the training set, measure the errors on the test set

Classical Statistics vs Statistical Learning

Classical Statistics

- ▶ Mean error

We want guarantees

- ▶ PAC Model Probably Approximately Correct
- ▶ What is the probability that the error is greater than a given threshold?

Example

Assume

$$Err(h) > \varepsilon$$

What is the probability that $Err_{emp,n}(h) = 0$?

$$\begin{aligned} Pr(Err_{emp,n}(h) = 0, Err(h) > \varepsilon) &= (1 - Err(h))^n \\ &< (1 - \varepsilon)^n \\ &< \exp(-\varepsilon n) \end{aligned}$$

Example

Assume

$$Err(h) > \varepsilon$$

What is the probability that $Err_{emp,n}(h) = 0$?

$$\begin{aligned} Pr(Err_{emp,n}(h) = 0, Err(h) > \varepsilon) &= (1 - Err(h))^n \\ &< (1 - \varepsilon)^n \\ &< \exp(-\varepsilon n) \end{aligned}$$

Hence, in order to guarantee a risk δ

$$Pr(Err_{emp,n}(h) = 0, Err(h) > \varepsilon) < \delta$$

Example

Assume

$$Err(h) > \varepsilon$$

What is the probability that $Err_{emp,n}(h) = 0$?

$$\begin{aligned} Pr(Err_{emp,n}(h) = 0, Err(h) > \varepsilon) &= (1 - Err(h))^n \\ &< (1 - \varepsilon)^n \\ &< \exp(-\varepsilon n) \end{aligned}$$

Hence, in order to guarantee a risk δ

$$Pr(Err_{emp,n}(h) = 0, Err(h) > \varepsilon) < \delta$$

The error should not be greater than

$$\varepsilon < \frac{1}{n} \ln \frac{1}{\delta}$$

Statistical Learning

Principle

- ▶ Find a bound on the generalization error
- ▶ Minimize the bound.

Note

$\hat{h}$ should be considered as a random variable, depending on the training set $\mathcal{E}$ and the number of examples n .

$\hat{h}_n$

Results

- deviation of the empirical error

$$Err(\hat{h}_n) \leq Err_{emp,n}(\hat{h}_n) + \mathcal{B}_1(n, \mathcal{H})$$

- bias-variance

$$Err(\hat{h}_n) \leq Err(h^*) + \mathcal{B}_2(n, \mathcal{H})$$

Approaches

Minimization of the empirical risk

- Model selection: Choose hypothesis space $\mathcal{H}$
- Choose $\hat{h}_n = \operatorname{argmin}\{Err_n(h), h \in \mathcal{H}\}$

beware of overfitting

Minimization of the structural risk

Given $\mathcal{H}_1 \subset \mathcal{H}_2 \subset \dots \subset \mathcal{H}_k$,

$$\text{Find } \hat{h}_n = \operatorname{argmin}\{Err_n(h) + \operatorname{pen}(n, k), h \in \mathcal{H}_k\}$$

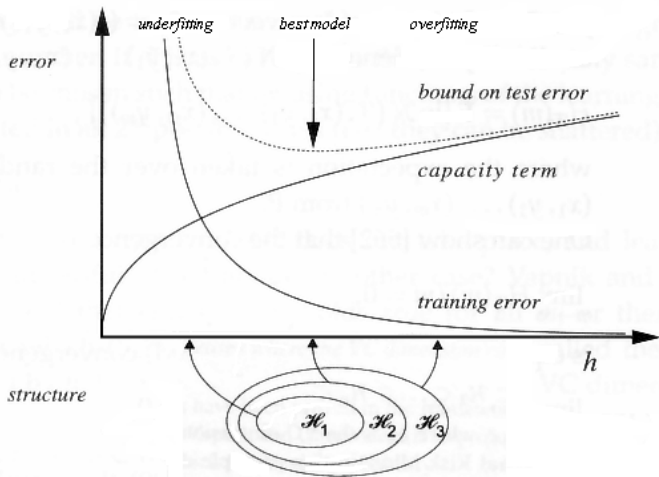
Which penalization?

Regularization

$$\text{Find } \hat{h}_n = \operatorname{argmin}\{Err_n(h) + \lambda ||h||, h \in \mathcal{H}\}$$

λ is identified by cross-validation

Structural Risk Minimization



Tool 1. Hoeffding bound

Hoeffding 1963

Let $X_1, \dots, X_n$ be independent random variables, and assume X_i takes values in $[a_i, b_i]$

Let $\bar{X} = (X_1 + \dots + X_n)/n$ be their empirical mean.

Theorem

$$\Pr(|\bar{X} - \mathbb{E}[\bar{X}]| \geq \varepsilon) \leq 2 \exp \left(- \frac{2\varepsilon^2 n^2}{\sum_{i=1}^n (b_i - a_i)^2} \right)$$

where $\mathbb{E}[\bar{X}]$ is the expectation of $\bar{X}$.

Hoeffding Bound (2)

Application: if

$$Pr(|Err(g) - Err_n(g)| > \varepsilon) < 2e^{-2n\varepsilon^2}$$

then with probability at least $1 - \delta$

$$Err(g) \leq Err_n(g) + \sqrt{\frac{\log 2/\delta}{2n}}$$

but this does not say anything about $\hat{h}_n$...

Uniform deviations

$$|Err(\hat{h}_n) - Err_n(\hat{h}_n)| \leq \sup_{h \in H} |Err(h) - Err_n(h)|$$

- if $\mathcal{H}$ is finite, consider the sum of $|Err(h) - Err_n(h)|$
- if $\mathcal{H}$ is infinite, consider its trace on the data

Statistical Learning. Definitions

Vapnik 92, 95, 98 **Trace of $\mathcal{H}$ on $\{x_1, \dots, x_n\}$**

$$Tr_{x_1, \dots, x_n}(\mathcal{H}) = \{(h(x_1), \dots, h(x_n)), h \in \mathcal{H}\}$$

Growth Function

$$S(\mathcal{H}, n) = \sup_{(x_1, \dots, x_n)} |Tr_{x_1, \dots, x_n}(\mathcal{H})|$$

Statistical Learning. Definitions (2)

Capacity of an hypothesis space $\mathcal{H}$

If the training set is of size n , and some function of $\mathcal{H}$ can have “any behavior” on n examples, nothing can be said!

$\mathcal{H}$ **shatters** $(x_1, \dots, x_n)$ iff

$$\forall (y_1, \dots, y_n) \in \{1, -1\}^n, \exists h \in \mathcal{H} \text{ s.t. } \forall i = 1 \dots n, h(x_i) = y_i$$

Vapnik Cervonenkis Dimension

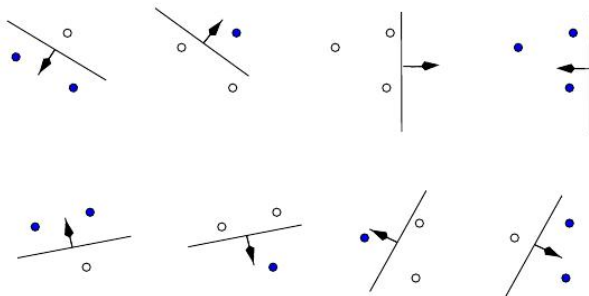
$$VC(\mathcal{H}) = \max \{n; (x_1, \dots, x_n) \text{ shattered by } \mathcal{H}\}$$

$$VC(\mathcal{H}) = \max\{n / S(\mathcal{H}, n) = 2^n\}$$

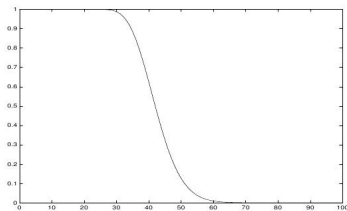
A shattered set

3 points in $\mathbb{R}^2$

$\mathcal{H}$ = lines of the plane



Growth Function of linear functions over $\mathbb{R}^{20}$



$$S(\mathcal{H}, n) \times \frac{1}{2^n} \text{ vs } n$$

The growth function is exponential w.r.t. n for $n < d = VC(\mathcal{H})$, then polynomial (in n^d).

Theorem, separable case

$\forall \delta > 0$, with probability at least $1 - \delta$

$$Err(h) \leq Err_n(h) + \sqrt{2 \frac{\log(S(H, 2n)) + \log(2/\delta)}{n}}$$

Idea 1: Double sample trick

Consider a second sample $\mathcal{E}'$

$$Pr(\sup_h (Err(h) - Err_n(h)) \geq \varepsilon) \leq$$

$$2Pr(\sup_h (Err'_n(h) - Err_n(h)) \geq \varepsilon/2)$$

where $Err'_n(h)$ is the empirical error on $\mathcal{E}'$.

Double sample trick

- ▶ There exists h s.t.
- ▶ A: $Err_{\mathcal{E}}(h) = 0$
- ▶ B: $Err(h) \geq \varepsilon$
- ▶ C: $Err_{\mathcal{E}'} \geq \frac{\varepsilon}{2}$

$$\begin{aligned} P(A(h) \& C(h)) &\geq P(A(h) \& B(h) \& C(h)) \\ &= P(A(h) \& B(h)) \cdot P(C(h) | A(h) \& B(h)) \\ &\geq \frac{1}{2} P(A(h) \& B(h)) \end{aligned}$$

Tool 2. Sauer Lemma

Sauer Lemma

If $d = VC(\mathcal{H})$

$$S(\mathcal{H}, n) = \sum_{i=1}^d \binom{n}{i}$$

For $n > d$,

$$S(H, n) \leq \left(\frac{en}{d}\right)^d$$

Idea 2: Symmetrization

Count the permutations that swap $\mathcal{E}$ et $\mathcal{E}'$.

Summary

$$Err(h) \leq Err_n(h) + \mathcal{O}\left(\sqrt{\frac{d \log n}{n}}\right)$$