Machine Learning

Alexandre Allauzen – Michèle Sebag
Thomas Schmitt
LIMSI – LRI

Oct. 7th, 2015

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 Discriminant Analysis

R. Guttierez-Osuna, http://research.cs.tamu.edu/prism/lectures/pr/pr_l10.pdf

Input

$$\mathcal{E} = \{(\mathbf{x}_i, y_i), \mathbf{x}_i \in \mathbb{R}^d, y_i \in \{+1, -1\}, i = 1 \ldots n\}$$

Output

$$h(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle \qquad \hat{y} = sg(h(\mathbf{x}))$$

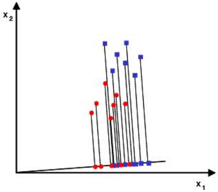
Remark

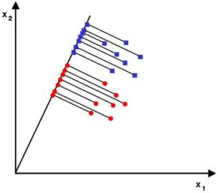
One might need $\langle \mathbf{w}, \mathbf{x} \rangle + b$

Solution: $\mathbf{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d :\mapsto \mathbf{x}' = (x_1, \ldots, x_d, 1) \in \mathbb{R}^{d+1}$

and $\mathbf{w} \in \mathbb{R}^d \mapsto \mathbf{w}' = (w_1, \ldots, w_d, b) \in \mathbb{R}^{d+1}$

$$\langle \mathbf{w}, \mathbf{x} \rangle + b \quad = \quad \langle \mathbf{w}', \mathbf{x}' \rangle$$





LDA, 2

Criterion

Find **w** s.t. it maximizes the discrimination.

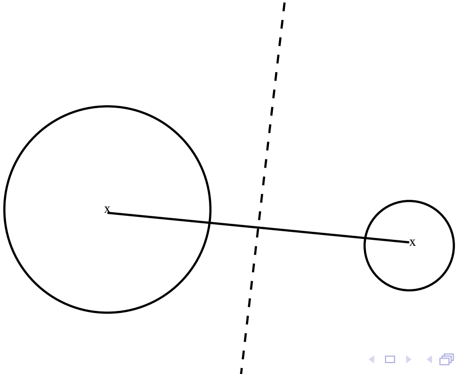
Define

$$\mu_+ = \text{average of } \mathbf{x}_i \text{ s.t. } y_i = +1$$

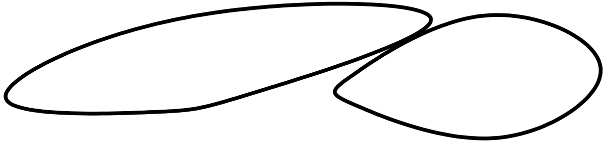
$$\mu_- = \text{average of } \mathbf{x}_i \text{ s.t. } y_i = \text{-}1$$

Build

$$\mathbf{w} \text{ s.t. } \langle \mathbf{w}, \mu_+ - \mu_- \rangle = 0$$



LDA, 3



LDA, 4

Intuition

Characterize the variance: within-class scatter matrix

$$S_W = \sum_{x_i, y_i = 1} (\mathbf{x}_i - \mu_+).(\mathbf{x}_i - \mu_+)' + \sum_{x_i, y_i = -1} (\mathbf{x}_i - \mu_-).(\mathbf{x}_i - \mu_-)'$$

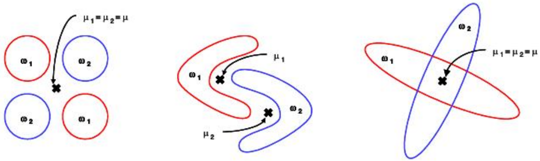
Characterize the difference: between-class scatter matrix

$$S_B = (\mu_+ - \mu_-).(\mu_+ - \mu_-)'$$

Solution

$$\text{find } \mathbf{w} = argmax \frac{\mathbf{w}' S_B \mathbf{w}}{\mathbf{w}' S_W \mathbf{w}}$$

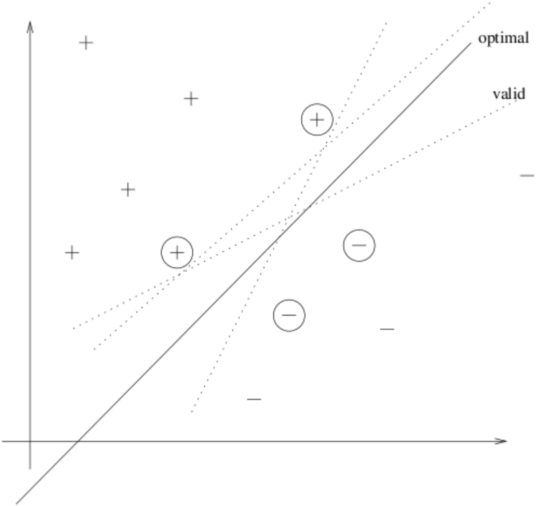
Some limitations



There is another limitation: any idea ?

Overview

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

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

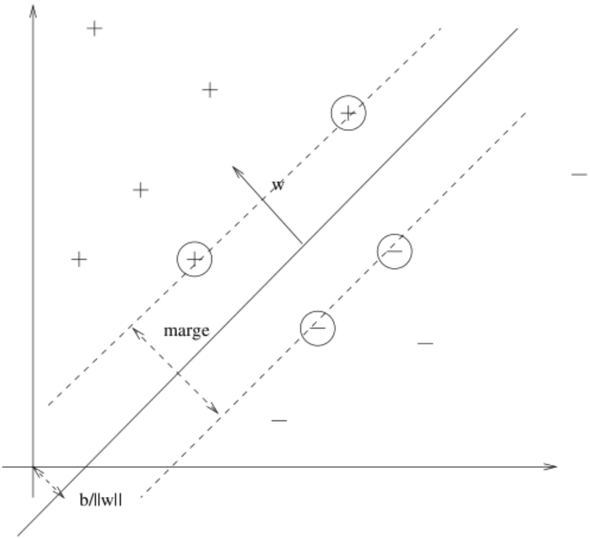
$$\langle \mathbf{w}, \mathbf{x}_+ - \mathbf{x}_- \rangle = 2$$

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

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

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

Maximal Margin Hyperplane



Maximize the Margin (2)

Problem

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

Primal Problem

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

- 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)

$$\begin{cases} \text{Maximize} & W(\alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \ < \mathbf{x}_i, \mathbf{x}_j > \\ \text{with the constraint} & \forall\ i,\ \ \alpha_i \geq 0 \\ & \sum_i \alpha_i y_i = 0 \end{cases}$$

Quadratic form w.r.t. α quadratic optimization is easy

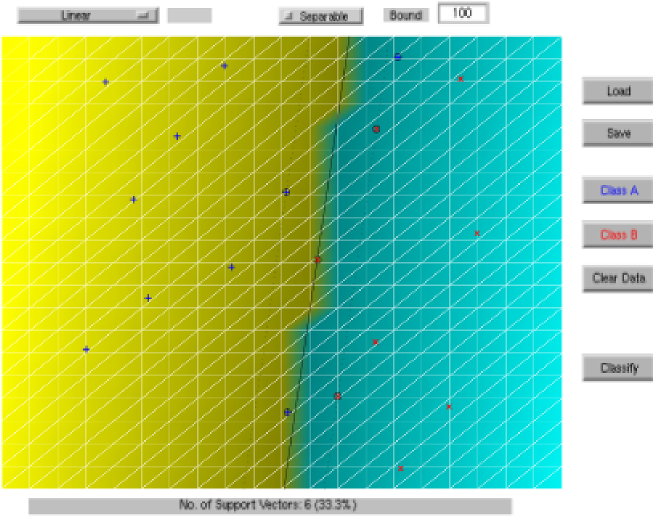
Solution: α_i^*

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

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

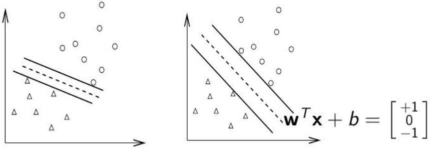
- If $(\langle \mathbf{x}_i, \mathbf{w}^* \rangle + b) y_i > 1$, $\alpha_i^* = 0$.
- If $\alpha_i^* > 0$, then $(\langle \mathbf{x}_i, \mathbf{w}^* \rangle + b) y_i = 1$ $\mathbf{x}_i$ **support vector**
- Compute b^* :

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


Linear
Separable
Bound
100
Load
Save
Class A
Class B
Clear Data
Classify
No. of Support Vectors: 6 (33.3%)

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
  $\text{sign}(y_i) = \text{sign}\ (h(\mathbf{x}_i)) \rightarrow$ maximize margin $y_i.h(\mathbf{x}_i)$
  achieve learning
- Regularization : minimize $||\mathbf{w}||$
  avoid overfitting

Support Vector Machines

General scheme

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

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

Constrained minimization of a convex function

$\rightarrow$ introduce Lagrange multipliers $\alpha_i \geq 0$, $i = 1 \ldots n$

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

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

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

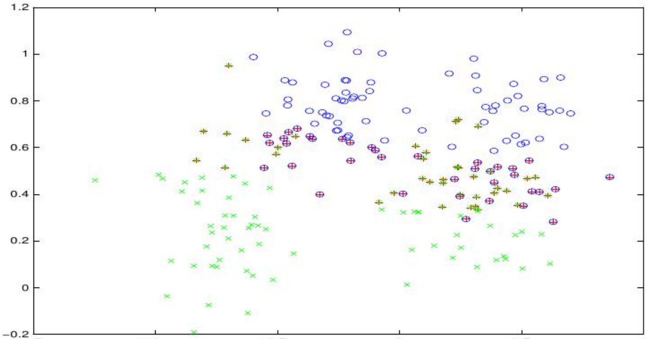
Support vectors

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

the only ones involved in the decision function

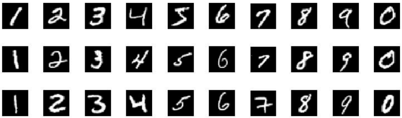
$$\mathbf{w} = \sum \alpha_i y_i \mathbf{x}_i$$

Support vectors, examples

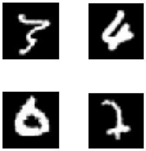


Support vectors, examples

MNIST data



Data



Support vectors

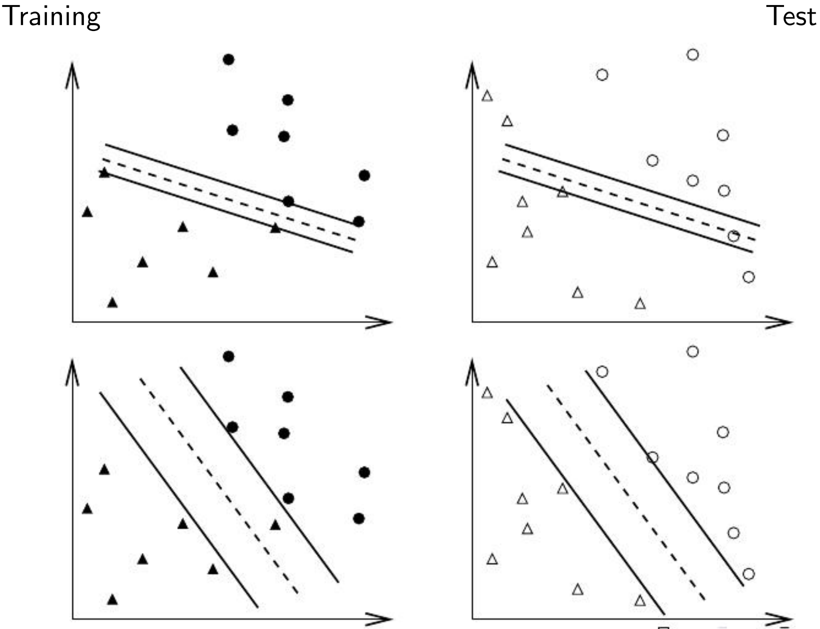
Remarks

- 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

Separable vs non-separable data



Linear hypotheses, non separable data

Cortes & Vapnik 95

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

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

Formalization

- Introduce slack variables ξ_i
- And penalize them

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

Critical decision: adjust C = error cost.

Primal problem, non separable case

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

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

At the optimum

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

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

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

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

Mathematically nice problem

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

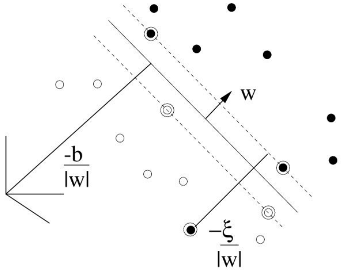
$$H_{i,j} = y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle$$

- Dual problem quadratic form

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

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

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

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

Complexity

Learning complexity

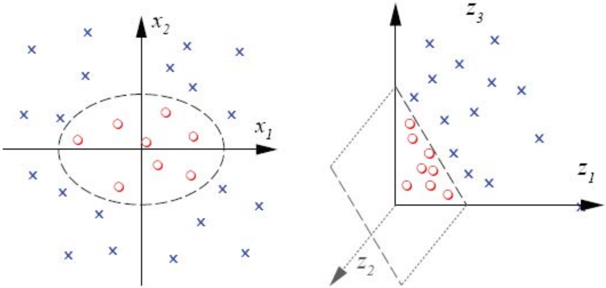
- Worst case: $\mathcal{O}(n^3)$
- Empirical complexity: depends on C
- $\mathcal{O}(n^2 n_{sv})$ where n_{sv} is the number of s.v.

Usage complexity

- $\mathcal{O}(n_{sv})$

Overview

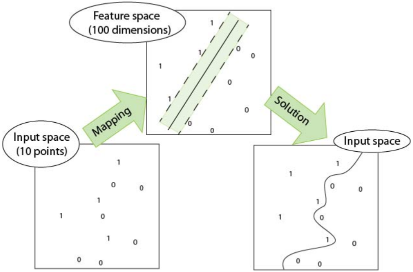
Non-separable data



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
  $\rightarrow$ 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

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

Dual problem

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

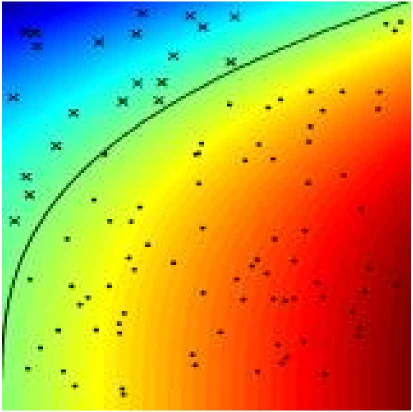
Hypothesis

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

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



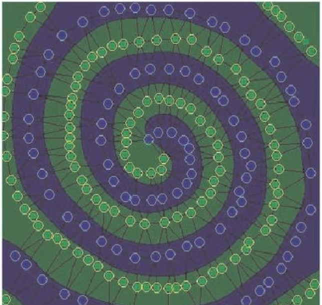
Example, Radius-Based Function kernel (RBF)

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

- No closed form Φ
- $\Phi(X)$ of infinite dimension
  For x in $\mathbb{R}$

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

- 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
- $\mathbf{i} = (i_1, i_2, \ldots, i_n)$ 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 $\mathbf{i}$
  $s = BICYCLE$, $\mathbf{i}$= (1, 3, 6), $s[\mathbf{i}] = BCL$

Definition

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

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

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

Prefer the normalized version

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

String kernels, followed

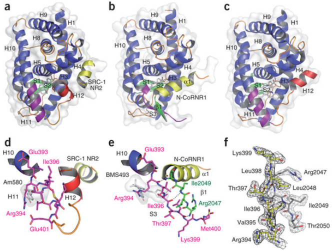
Application 1

Document mining

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

Application 2, Bio-informatics

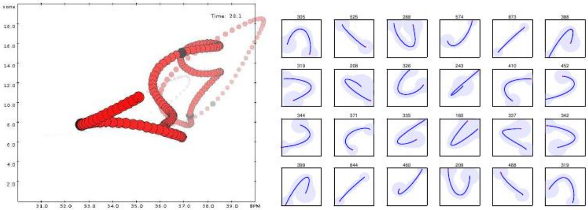
- Pre-processing matters a lot
- Classification (secondary structures)



Extension to graph kernels *http://videolectures.net/gbr07_vert_ckac/*

Application to musical analysis

- Input: Midi files
- Pre-processing, rythm 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
- $K(\mathbf{x}, \mathbf{x}') = limit_{p \rightarrow \infty} K_p(\mathbf{x}, \mathbf{x}')$ is p.d. if it exists
- $K(A, B) = \sum_{\mathbf{x} \in A, \mathbf{x}' \in B} K(x, x')$ is a p.d. kernel

Overview

Multi-class discrimination

Input

Binary case

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

Multi-class case

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

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 $2 \ldots k \rightarrow -1$ $\quad h_1$

Pb 2: class $2 \rightarrow +1$, classes $1, 3, \ldots k \rightarrow -1$ $\quad h_2$

...

Prediction

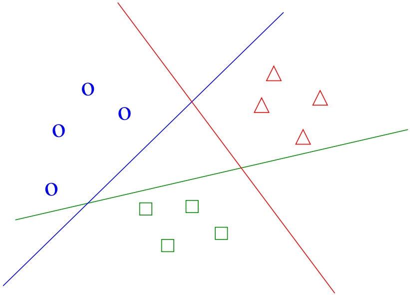
$$h(\mathbf{x}) = i \text{ iff } h_i(\mathbf{x}) = \; argmax\{h_j(\mathbf{x}), j = 1 \ldots k\}$$

Justification

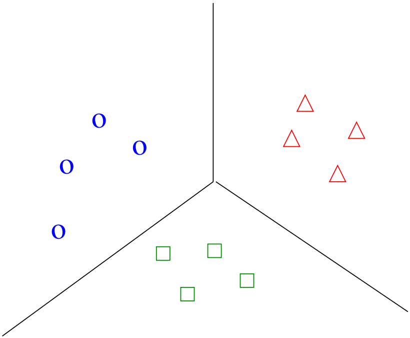
If $\mathbf{x}$ belongs to class 1, one should have

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

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: additionnal constraints

Another option Vapnik 98; Weston, Watkins 99

$$\begin{cases} \text{Minimise} & \frac{1}{2}\sum_{j=1}^{k} ||\mathbf{w}_j||^2 + \; C\sum_{i=1}^{n}\sum_{\ell=1,\ell\neq y_i}^{k} \xi_{i,\ell} \\ \text{Subject to} & \forall i, \forall \ell \neq y_i, \\ & \qquad (\langle w_{y_i}, \mathbf{x}_i\rangle + b_{y_i}) \geq (\langle w_\ell, \mathbf{x}_i\rangle + b_\ell) + 2 - \xi_{i,\ell} \\ & \qquad \xi_{i,\ell} \geq 0 \end{cases}$$

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

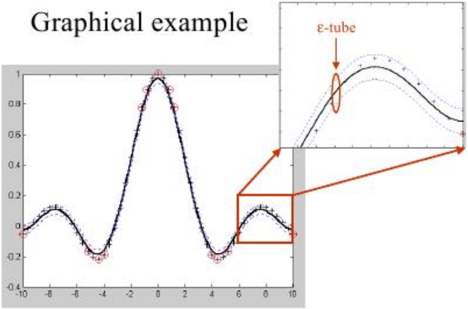
Overview

Regression

Input

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

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 — loss function
- ... while being as flat as possible — regularization

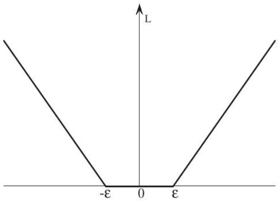
Formulation

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

Regression with Support Vector Machines, followed

Using slack variables

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



Regression with Support Vector Machines, followed

Primal problem

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

Dual problem

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

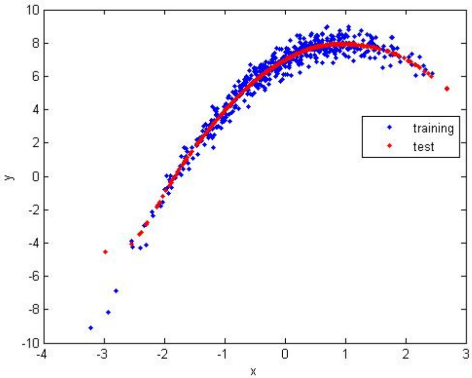
Regression with Support Vector Machines, followed

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

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



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 $D >> n$ curse of dimensionality

Dimensionality reduction mandatory

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

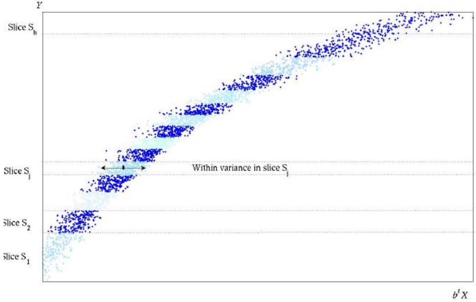
$$\pi : X \mapsto S \subset \mathbb{R}^d$$

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

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

Sliced Inverse Regression

Bernard-Michel et al, 09



More: http://mistis.inrialpes.fr/learninria/
S. Girard

Overview

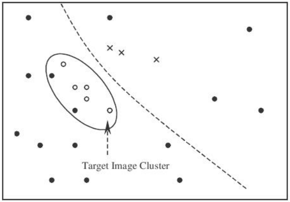
Novelty Detection

Input

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

Context

- Information retrieval



- Identification of the data support

estimation of distribution

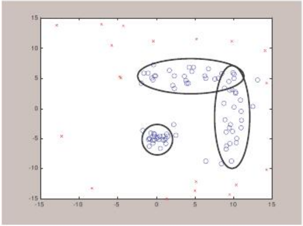
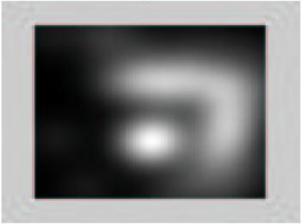
Critical issue

- Classification approaches not efficient: too much noise

One-class SVM

Formulation

$$\left\{\begin{array}{ll} \text{Min.} & \frac{1}{2}\,||\mathbf{w}||^2 - \rho \; {\color{red}+\mathbf{C}\sum_\mathbf{i} \xi_\mathbf{i}} \\ \text{s.t.} & \forall\; i = 1 \ldots n \\ & \quad \langle \mathbf{w}, \mathbf{x}_i \rangle \geq \rho \; {\color{red}-\xi_\mathbf{i}} \end{array}\right.$$

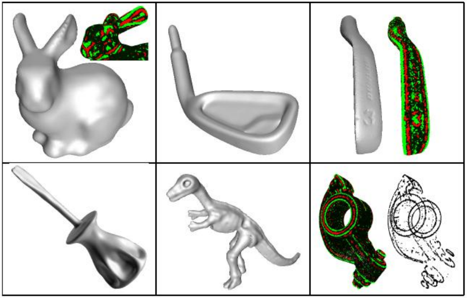
Dual problem

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

Implicit surface modelling

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

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

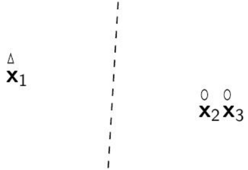


Overview

Normalisation / Scaling

Needed to prevent attributes to steal the game

	Height	Gender	Class
$\mathbf{x}_1$	150	F	1
$\mathbf{x}_2$	180	M	0
$\mathbf{x}_3$	185	M	0



$\Rightarrow$ 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 (Scale_{train})
- Learn from the normalized training set
- Scale the test set according to Scale_{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

Data sampling

Simple approaches

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

Incremental approaches

Syed et al. 99

- Partition $\mathcal{E} \rightarrow \mathcal{E}_1, \ldots \mathcal{E}_N$
- 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 *Unsupervised learning, J. Gama*
- Learn a hypothesis on each cluster
- Aggregate hypotheses

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 $\mathcal{E}$

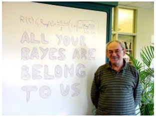
$$\begin{cases} \text{Min.} & \frac{1}{2} \sum_{i,j,\in \mathcal{F}} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle + \; C \sum_{\ell=1}^{n} \xi_\ell \\ \text{t.q.} & \forall \ell = 1 \ldots n, \\ & \quad (\langle w, \mathbf{x}_\ell \rangle + b) \geq 1 - \xi_\ell \\ & \quad \xi_\ell \geq 0 \end{cases}$$

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
- Videolectures + ML Summer Schools
- Large scale Machine Learning challenge,
  ICML 2008 wshop:
  http://largescale.ml.tu-berlin.de/workshop/

Overview

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

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

$\ell =$ 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 $h^* = argmin\{Err(h), h \in \mathcal{H}\}$.

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", $Err_{emp,n} \to Err$ when $n \to \infty$

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 — here, Learning
- Predictions about its behavior — 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 — here, Learning
- Predictions about its behavior — 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

- A set of data breast cancer
- $\bar{x}^+$: 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. $\widehat{h_n}$

Results

- deviation of the empirical error

$$Err(\widehat{h_n}) \leq Err_{emp,n}(\widehat{h_n}) + \mathcal{B}_1(n, \mathcal{H})$$

- bias-variance

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

Approaches

Minimization of the empirical risk

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

beware of overfitting

Minimization of the structual risk

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

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

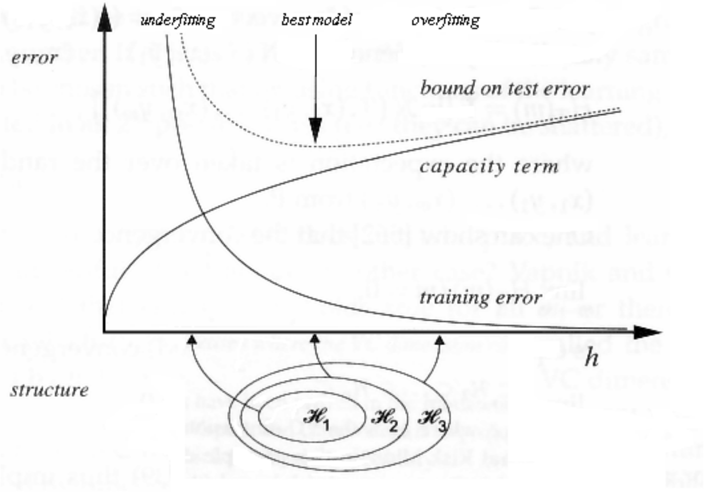
Which penalization?

Regularization

$$\text{Find } \widehat{h_n} = argmin\{Err_n(h) + \lambda||h||,\ h \in \mathcal{H}\}$$

λ is identified by cross-validation

Structural Risk Minimization



Tool 1. Hoeffding bound

Hoeffing 1963

Let $X_1 \ldots, X_n$ be independent random variables, and assume X_i takes values in $[a_i, b_i]$
Let $\overline{X} = (X_1 + \cdots + X_n)/n$ be their empirical mean.

Theorem

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

where $\mathrm{E}[\overline{X}]$ is the expectation of $\overline{X}$.

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

Statistical Learning. Definitions

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

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

Growth Function

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

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** $(x_1, \ldots x_n)$ iff

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

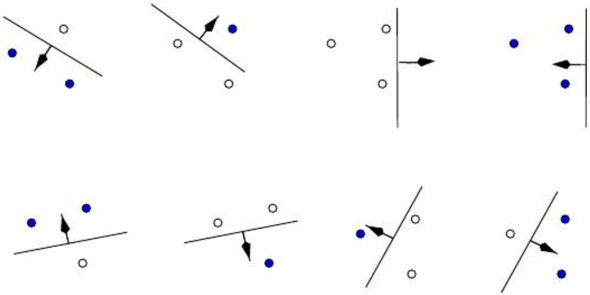
Vapnik Cervonenkis Dimension

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

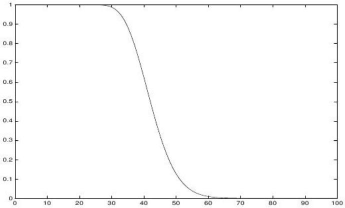
$$VC(\mathcal{H}) = max\{n \; / \; S(\mathcal{H}, n) = 2^n\}$$

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 exponental 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: $Err_{\mathcal{E}'} \geq \frac{\varepsilon}{2}$

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

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

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