SVM and kernel machines

Stéphane Canu, Gaëlle Loosli, Alain Rakotomamonjy

Recent advances in kernel machines

Julio 2011

Escuela de Ciencias Informáticas 2011

Laboratoire d'Informatique, de Traitement de l'Information et des Systèmes

Stéphane Canu, Gaëlle Loosli & Alain Rakotomamonjy

stephane.canu@litislab.eu

UNIVERSITÉ DE ROUEN

Le Havre Université

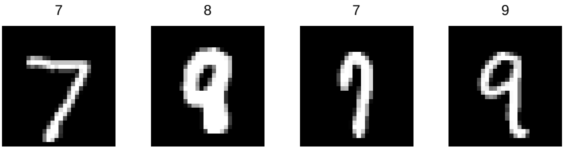
INSA ROUEN

Optical character recognition

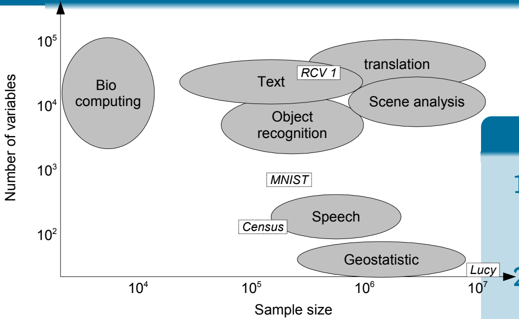
Example (The MNIST database)

- MNIST[a], data = « image-label »
- $n = 60,000$; $d = 700$; classes $= 10$
- Kernel error rate = 0.56 %,
- Best error rate = 0.4 % .

[a] http://yann.lecun.com/exdb/mnist/index.html



Learning challenges: the size effect



3 key issues

1. learn any problem:
   - functional universality
2. from data:
   - statistical consistency
3. with large data sets:
   - computational efficency

kernel machines adress these three issues
(up to a certain point regarding efficency)

L. Bottou, 2006

Historical perspective on kernel machines

statistics

- **1960** Parzen, Nadaraya Watson
- **1970** Splines
- **1980** Kernels: Silverman, Hardle...
- **1990** sparsity: Donoho (pursuit), Tibshirani (Lasso)...

Statistical learning

1985 Neural networks:

- non linear - universal
- structural complexity
- non convex optimization

1992 Vapnik et. al.

- theory - regularization - consistancy
- convexity - Linearity
- Kernel - universality
- sparsity
- results: MNIST

Notations

- inputs $\mathbf{x} \in \mathcal{X} = \mathbb{R}^d$, d feartures
- outputs y
- training set $(\mathbf{x}_i, y_i)$, $i = 1, n$, n exemples
- test set $(\mathbf{x}_j, y_j)$, $j = 1, \ell$, ℓ exemples
- kernel $k(\mathbf{x}_i, \mathbf{x}_j)$ $\quad \mathcal{X} \times \mathcal{X} \longmapsto \mathbb{R}$
- RKHS $\mathcal{H}$ (set of hypothesis associated with positive kernel k)
- RKKS $\mathcal{K}$ (set of hypothesis associated with kernel k) – Krein

Definition (Kernel machines)

$$\mathcal{A}\big((\mathbf{x}_i, y_i)_{i=1,n}\big)(\mathbf{x}) = \psi\Big(\sum_{i=1}^{n} \alpha_i k(\mathbf{x}, \mathbf{x}_i) + \sum_{j=1}^{p} \beta_j q_j(\mathbf{x})\Big)$$

$\boldsymbol{\alpha}$ et $\boldsymbol{\beta}$: parameters to be estimated

Road map

Multiple Kernel

The model

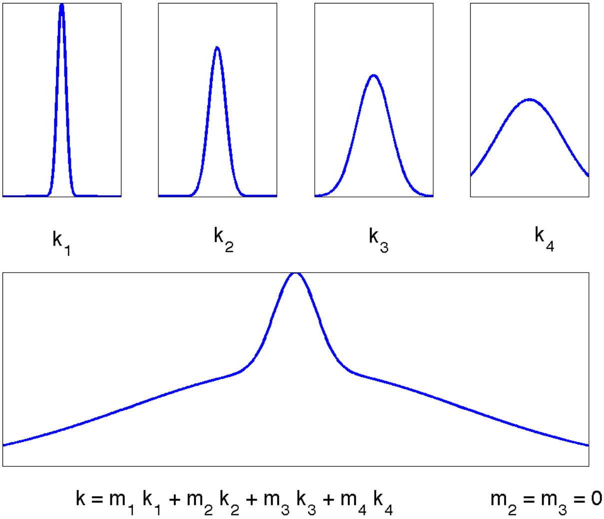
$$f(x) = \sum_{i=1}^{n} \alpha_i k(x, x_i) + b,$$

Given M kernel functions $K_1, \ldots, K_M$ that are potentially well suited for a given problem, find a positive linear combination of these kernels such that the resutling kernel k is "optimal"

$$k(\mathbf{x}, \mathbf{x}') = \sum_{m=1}^{M} d_m k_m(\mathbf{x}, \mathbf{x}'), \text{ with } d_m \geq 0, \sum_m d_m = 1$$

Need to learn together the kernel coefficients d_m and the SVR parameters α_i, b.

Multiple Kernel: illustration



Multiple Kernel functional Learning

The problem (for given C and t)

$$\begin{array}{ll}\min\limits_{\{f_m\},b,\xi,d} & \frac{1}{2}\sum\limits_m \frac{1}{d_m}\|f_m\|^2_{\mathcal{H}_m} + C\sum\limits_i \xi_i \\ \text{s.t.} & \Big|\sum\limits_m f_m(x_i) + b - y_i\Big| \leq t + \xi_i \quad \forall i \xi_i \geq 0 \quad \forall i \\ & \sum\limits_m d_m = 1 \ , \quad d_m \geq 0 \quad \forall m \ ,\end{array}$$

regularization formulation

$$\begin{array}{ll}\min\limits_{\{f_m\},b,d} & \frac{1}{2}\sum\limits_m \frac{1}{d_m}\|f_m\|^2_{\mathcal{H}_m} + C\sum\limits_i \max\Big(\Big|\sum\limits_m f_m(x_i) + b - y_i\Big| - t, 0\Big) \\ & \sum\limits_m d_m = 1 \ , \quad d_m \geq 0 \quad \forall m \ ,\end{array}$$

Equivalently

$$\min_{\{f_m\},b,\xi,d} \quad \sum_i \max\Big(\Big|\sum_m f_m(x_i) + b - y_i\Big| - t, 0\Big) + \frac{1}{2C}\sum_m \frac{1}{d_m}\|f_m\|^2_{\mathcal{H}_m} + \mu \sum_m |d_m|$$

Multiple Kernel functional Learning

The problem (for given C and t)

$$\begin{aligned} \min_{\{f_m\},b,\xi,d} \quad & \frac{1}{2}\sum_m \frac{1}{d_m}\|f_m\|^2_{\mathcal{H}_m} + C\sum_i \xi_i \\ \text{s.t.} \quad & \left|\sum_m f_m(x_i) + b - y_i\right| \le t + \xi_i \quad \forall i \xi_i \ge 0 \quad \forall i \\ & \sum_m d_m = 1 \,, \quad d_m \ge 0 \quad \forall m \,, \end{aligned}$$

Treated as a bi-level optimization task

$$\begin{aligned} \min_{d \in \mathbb{R}^M} \quad & \left\{ \begin{aligned} \min_{\{f_m\},b,\xi} \quad & \frac{1}{2}\sum_m \frac{1}{d_m}\|f_m\|^2_{\mathcal{H}_m} + C\sum_i \xi_i \\ \text{s.t.} \quad & \left|\sum_m f_m(x_i) + b - y_i\right| \ge t + \xi_i \quad \forall i \\ & \xi_i \ge 0 \quad \forall i \end{aligned} \right. \\ \text{s.t.} \quad & \sum_m d_m = 1 \,, \quad d_m \ge 0 \quad \forall m \,, \end{aligned}$$

Multiple Kernel Algorithm

Use a Reduced Gradient Algorithm[1]

$$\begin{aligned} \min_{d \in \mathbb{R}^M} \quad & J(d) \\ \text{s.t.} \quad & \sum_m d_m = 1 \ , \quad d_m \geq 0 \quad \forall m \ , \end{aligned}$$

SimpleMKL algorithm

set $d_m = \frac{1}{M}$ for $m = 1, \dots, M$
while stopping criterion not met **do**
compute $J(d)$ using an QP solver with $K = \sum_m d_m K_m$
compute $\frac{\partial J}{\partial d_m}$, Hessian and descent direction D
$\gamma \leftarrow$ compute optimal stepsize
$d \leftarrow d + \gamma D$
end while

$\longrightarrow$ Recent improvement reported using the Hessian

[1] Rakotomamonjy et al. JMLR 08

Complexity

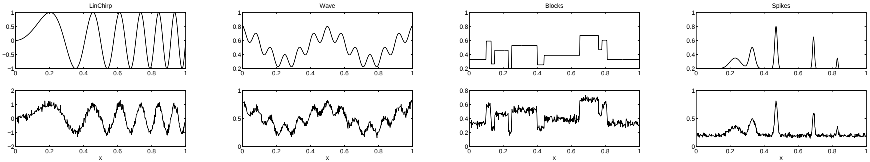
For each iteration:

- SVM training: $O(nn_{\mathrm{sv}} + n_{\mathrm{sv}}^3)$.
- Inverting $K_{\mathrm{sv,sv}}$ is $O(n_{\mathrm{sv}}^3)$, but might already be available as a by-product of the SVM training.
- Computing H: $O(Mn_{\mathrm{sv}}^2)$
- Finding d: $O(M^3)$.

The number of iterations is usually less than 10.

$\longrightarrow$ When $M < n_{\mathrm{sv}}$, computing d is not more expensive than QP.

Multiple Kernel experiments



	Single Kernel	Kernel *Dil*		Kernel *Dil-Trans*	
Data Set	Norm. MSE (%)	#Kernel	Norm. MSE	#Kernel	Norm. MSE
LinChirp	1.46 ± 0.28	7.0	1.00 ± 0.15	21.5	**0.92 ± 0.20**
Wave	0.98 ± 0.06	5.5	**0.73 ± 0.10**	20.6	0.79 ± 0.07
Blocks	1.96 ± 0.14	6.0	2.11 ± 0.12	19.4	**1.94 ± 0.13**
Spike	6.85 ± 0.68	6.1	6.97 ± 0.84	12.8	**5.58 ± 0.84**

Table: Normalized Mean Square error averaged over 20 runs.

Conclusion on multiple kernel (MKL)

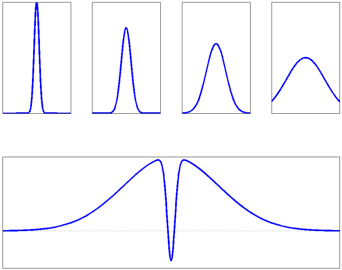
- MKL: Kernel tuning, variable selection. . .
  - extention to classification and one class SVM
- SVM KM: an efficient Matlab toolbox (available at MLOSS)[2]
- Multiple Kernels for Image Classification: Software and Experiments on Caltech-101[3]
- new trend: Multi kernel and Multi task

[2] http://mloss.org/software/view/33/

[3] http://www.robots.ox.ac.uk/~vgg/software/MKL/

Road map

Learning with non positive kernel: why?



$k = m_1 k_1 + m_2 k_2 + m_3 k_3 + m_4 k_4$ $\qquad$ $m_2 = m_3 = 0$ and $m_1 < 0$

- multiple non positive kernels

$$k(\mathbf{x}, \mathbf{x}') = \sum_{m=1}^{M} d_m k_m(\mathbf{x}, \mathbf{x}')$$

without $d_m \geq 0$,

- Biological non positive kernels
- Positive radial kernels are Localized
- $\tanh(\mathbf{w}^\top \mathbf{x})$

NON Positive kernels

Definition of the associated pre Krein space

- $\mathcal{K}_0 = \{f \in \mathbb{R}^{\mathcal{X}} | f(x) = \sum_{i=1}^{n} \alpha_i k(x, x_i), \alpha_i \in \mathbb{R}, x_i \in \mathcal{X}\}$
- inner product on $\mathcal{K}_0$:
  $f(x) = \sum_{i=1}^{n} \alpha_i k(x, x_i), \quad g(x) = \sum_{i=1}^{m} \beta_i k(x, \widetilde{\mathbf{x}}_i)$

$$\langle f(.), g(.)\rangle_{\mathcal{K}_0} = \sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i \beta_j \; k(x_i, \widetilde{\mathbf{x}}_j)$$

the continuity of the evaluation functional

- $A_x f = f(x) = \langle f(.), k(x, .)\rangle_{\mathcal{K}_\mathbf{0}}$ evaluation functional
- $\langle k(x, .), k(y, .)\rangle_{\mathcal{K}_\mathbf{0}} = k(x, y)$ reproducing property

no more norm: $\langle f, f\rangle_{\mathcal{K}_0}$ is NOT always positive

Reproducing Kernel Krein Spaces (RKKS)

Fundamental hypothesis

- There exist two positive kernels k_+ and k_- such that

$$k(x, y) = k_+(x, y) - k_-(x, y)$$

the RKKS space has to be complete

- define a topology using the positive kernels k_+ and k_-
- $\mathcal{K} = \widehat{\mathcal{K}}_0$
- remark about unicity

Theorem: 3 equivalent statements

- $\mathcal{K}$ is a RKKS with kernel k
- $k(x, y) = k_+(x, y) - k_-(x, y)$
- $k(x, y)$ is dominated by a positive kernel

Examples

- Minkowski space time

$$\langle (x,y,z,t),(\widetilde{\mathbf{x}},\widetilde{\mathbf{y}},\widetilde{z},\widetilde{t})\rangle_{\mathcal{K}} = x\widetilde{\mathbf{x}} + y\widetilde{\mathbf{y}} + z\widetilde{z} - t\widetilde{t}$$

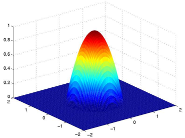
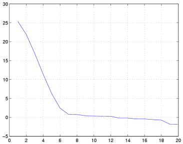
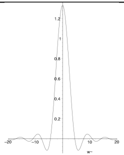
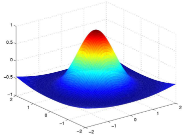
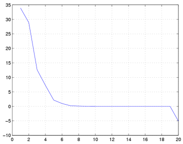
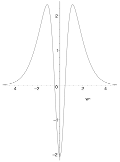
- Difference of two gaussians

$$k(s,t) := \alpha \exp^{-\frac{\|s-t\|^2}{b}} - \beta \exp^{-\frac{\|s-t\|^2}{c}}$$

- "Wavelets" kind: assume $\mathcal{H} = V \oplus W$

$$\mathcal{K} = V \ominus W$$

Examples

Kernel	2D kernel	Eigenvalues	Fourier
Epanechnikov kernel $\left(1 - \frac{\|\boldsymbol{s}-\boldsymbol{t}\|^2}{\sigma}\right)^p$, for $\frac{\|\boldsymbol{s}-\boldsymbol{t}\|^2}{\sigma} \leqslant 1$			
Gaussian Combination $\exp\left(\frac{-\|\boldsymbol{s}-\boldsymbol{t}\|^2}{\sigma_1}\right)$ $+\exp\left(\frac{-\|\boldsymbol{s}-\boldsymbol{t}\|^2}{\sigma_2}\right)$ $-\exp\left(\frac{-\|\boldsymbol{s}-\boldsymbol{t}\|^2}{\sigma_3}\right)$			

Examples of indefinite kernels. Column 2 shows the 2D surface of the kernel with respect to the origin, column 3 shows plots of the 20 eigenvalues with largest magnitude of uniformly spaced data from the interval $[-2, 2]$, column 4 shows plots of the Fourier spectra.

An other view on splines

Interpolation is an ill posed problem

Let $\mathcal{H}$ be a RKHS: Minimize $\|f\|^2_{\mathcal{H}}$ such that $f(\mathbf{x}_i) = y_i, i = 1, n$

In a Krein space

- NO more norm: how to regularize?
- project 0 on the set of constrains: $K\alpha = \mathbf{y}$

approximation is an ill posed problem

Let $\mathcal{H}$ be a RKHS: Minimize $\|f\|^2_{\mathcal{H}} + \frac{C}{2}\sum \xi_i^2$ such that $f(\mathbf{x}_i) - y_i = \xi_i, i = 1, n$ In a Krein space

- NO more norm: how to regularize?
- compute a path between 0 and the interpollating solution

Non positive SVM: related work

- considering that the indefinite kernel is a perturbation of a true Mercer kernel.
- finding a stationary point, which is not unique but each of those performs correct separation. Moreover, it is shown that the problem is then cannot be seen as a margin maximization although a notion of margin can be defined.
- Krein space instead of a Hilbert space.
- Applying this to SVM requires to interpret this stabilization setting.
- a (unconstraint) quadratic program in a Krein space has a unique solution (if the involved matrix is non singular) which is in general a stationary point.

SVM in a Krein space

$$\begin{cases} \min\limits_{f,\alpha_0} & \frac{1}{2}\|f\|^2_{\mathcal{H}} \\ \text{with} & y_i\big(f(\mathbf{x}_i)+\alpha_0\big) \geq 1 \end{cases} \Leftrightarrow \begin{cases} \text{stab}_{f,\alpha_0} & \frac{1}{2}\langle f, f\rangle_{\mathcal{K}} \\ \text{with} & y_i\big(f(\mathbf{x}_i)+\alpha_0\big) \geq 1 \end{cases}$$

The representer theorem holds: $\langle f, f\rangle_{\mathcal{K}} = \alpha^\top K\alpha$

solve the problem using normal residuals (ie. solving $Ax = b$ via $A^\top Ax = A^\top b$).

$$\begin{aligned} \alpha^\top G &= \mathbf{1} - \lambda\mathbf{y}^\top - \mu^\top + \eta^\top \\ \alpha^\top GG^\top &= (\mathbf{1} - \lambda\mathbf{y}^\top - \mu^\top + \eta^\top)G^\top \end{aligned}$$

This can be seen as least squares. All the other conditions remain identical.

SVM and KKT conditions of optimality

KKT conditions for SVM $\begin{cases} \min_\alpha & \frac{1}{2}\alpha^\top G\alpha - \alpha^\top \mathbf{1} \\ \text{subject to} & \alpha^\top \mathbf{y} = 0 \\ \text{and} & 0 \le \alpha_i \le C \qquad \forall i \in [1..n] \end{cases}$

The stationarity condition is as follows:

$$-\alpha^\top G + \mathbf{1} - \lambda \mathbf{y}^\top - \mu^\top + \eta^\top = \mathbf{0}$$

The primal admissibility is given by $\begin{array}{ll} -\alpha^\top \mathbf{y} = 0 & \\ \alpha_i \le C & \forall i \in [1..n] \\ \alpha_i \ge 0 & \forall i \in [1..n] \end{array}$

The dual admissibility is given by $\begin{array}{ll} \mu_i \ge 0 & \forall i \in [1..n] \\ \eta_i \ge 0 & \forall i \in [1..n] \end{array}$

The complementary conditions are $\begin{array}{ll} -\alpha_i \mu_i = 0 & \forall i \in [1..n] \\ (\alpha_i - C)\eta_i = 0 & \forall i \in [1..n] \end{array}$

Point of view 2 : a stabilization problem

Stabilizing $\mathcal{J}$ is equivalent to minimizing $\mathcal{M}$:

$$\mathcal{J} = \frac{1}{2}\alpha^\top G\alpha - \alpha^\top \mathbf{1} \qquad\qquad \mathcal{M}(\alpha) = \langle \alpha^\top G - \mathbf{1}^\top, \alpha^\top G - \mathbf{1}^\top \rangle$$

This provides $\begin{cases} \min_\alpha & \langle \alpha^\top G - \mathbf{1}^\top, \alpha^\top G - \mathbf{1}^\top \rangle \\ \text{with} & \alpha^\top \mathbf{y} = 0 \\ \text{and} & 0 \le \alpha_i \le C \quad \forall i \in [1..n] \end{cases}$

KKT conditions

The stationarity condition is as follows:

$$(\alpha^\top G - \mathbf{1}^\top + \lambda y^\top - \mu^\top + \eta^\top)G^\top = \mathbf{0}$$

The primal admissibility is given by
$$\begin{array}{ll} -\alpha^\top \mathbf{y} = 0 & \\ \alpha_i \le C & \forall i \in [1..n] \\ \alpha_i \ge 0 & \forall i \in [1..n] \end{array}$$

The dual admissibility is given by
$$\begin{array}{ll} \mu_i \ge 0 & \forall i \in [1..n] \\ \eta_i \ge 0 & \forall i \in [1..n] \end{array}$$

The complementary conditions are
$$\begin{array}{ll} -\alpha_i \mu_i = 0 & \forall i \in [1..n] \\ (\alpha_i - C)\eta_i = 0 & \forall i \in [1..n] \end{array}$$

Point of view 3 : The projection

Chasing the the most stable point, ie. the admissible point minimizing the gradient of the cost function (which is $\alpha^\top G - \mathbf{1}^\top$).

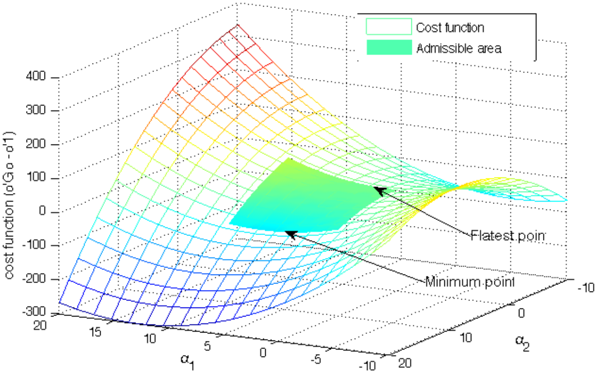


Figure: SVM cost function with sigmoid kernel, illustrated for 2 support vectors. The plain area shows the admissible solutions.

The solver

The proposed algorithm is derived from active set approach for SVM, The sets of points are defined according to the complementarity conditions (see table 2).

Table: Definition of groups for active set depending on the dual variable values

Group	α	η	μ
I_0	0	0	> 0
I_C	C	> 0	0
I_w	$0 < \alpha < C$	0	0

By default, all training points are in the non support vector set I_0 except for a couple with opposite labels which is in I_w. Any other initial situation based on warm-start or a priori does not change the algorithm.

Relaxing constraints in I_0 or I_C

If the current solution is admissible, we check the stationarity conditions for I_0 and I_C The most violating point is transferred from its group to I_w.

The NPSVM algorithm

Initialize (one random point for each class in I_w, all others in I_0)
while solution is not optimal **do**
  solve linear system
  if primal admissibility is not satisfied **then**
    project solution in the admissible domain : remove a support vector from I_w (to I_0 or I_C)
  else if stationarity condition is not satisfied **then**
    add new support vector to I_w (from I_0 or I_C)
  end if
end while

Experimental results

- sigmoid kernel (tanh) : $k(x_i, x_j = tanh(scale \times \langle x_i, x_j \rangle + bias)$
- the epanechnikov kernel: $k(x_i, x_j) = max(0, 1 - \gamma \langle x_i, x_j \rangle)$

Validation protocol

- split randomly the dataset, 2/3 for cross validation, 1/3 for test.
- perform 10 fold cross validation on the validation set ($C \in [0.01, 0.1, 1, 10, 100, 1000]$, $\sigma \in [0.10.5, 1, 5, 10, 15, 25, 50, 100, 250, 500] * \sqrt{n}]$ for rbf kernel, $scale = [pow2(-5 : 1.5 : 2), -pow2(-5 : 1.5 : 2)]$ and $bias = [pow2(-5 : 1.5 : 2), -pow2(-5 : 1.5 : 2)]$ for tanh kernel).
- train the svm on the full validation set with the parameters providing the best average performance during cross validation.
- test on the separate test set.

Experimental results

Solver	Kernel	Checkers	Checkers 10	Clown
C-SVM	rbf	87.1 % (51 sv)	79.5 % (151 sv)	99.93% (17 sv)
NPSVM	rbf	88.9 % (167 sv)	81.9 % (170 sv)	99.93% (53 sv)
Constraint-NPSVM	rbf	87.1 % (107 sv)	81.6 % (141 sv)	99.97% (54 sv)
NPSVM	tanh	74.9 % (35 sv)	71.8 % (41 sv)	99.97% (81 sv)
Constraint-NPSVM	tanh	86.6 % (114 sv)	81.6 % (145 sv)	99.87% (27 sv)
NPSVM	epanech	86.8 % (133 sv)	81.4 % (118 sv)	99.93% (32 sv)
Constraint-NPSVM	epanech	83.0 % (119 sv)	77.7 % (133 sv)	99.63% (56 sv)

Table: Results on synthetic dataset. Dataset sizes : 200 training points, 3000 testing points. Checkers 10 is a checker dataset with 10% of overlapping between classes. Clowns is also known as apple/banana.

Table: Results on some UCI dataset.

Solver	kernel	Heart	Sonar	Breast
C-SVM	rbf	82.22% (23.2 sv)	84.78% (90.3 sv)	97.47% (53.8 sv)
NPSVM	rbf	**83.44**% (35.9 sv)	**<u>86.09%</u>** (94.9 sv)	97.37% (56.6 sv)
NPSVM	tanh	**82.44**% (14.8 sv)	84.06% (70.1 sv)	**<u>97.76 %</u>** (116 sv)

Comparison with libSVM

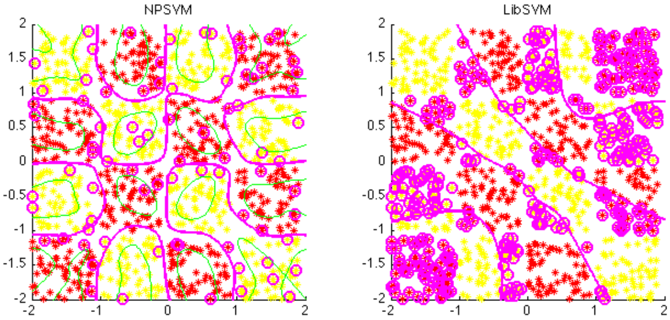


Figure: Results on checkers with NPSVM on the left and libSVM on the right, for an identical sigmoid kernel (scale = 2, bias = -2). Circles are support vectors.

Comparison to IndefiniteSVM

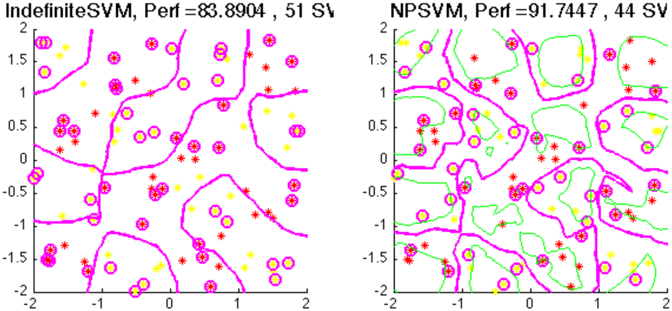


Figure: Results on checkers with IndefiniteSVM on the left and NPSVM on the right, for an identical epanech kernel. Circles are support vectors.

Discussion

SVM with non positive kernels is possible

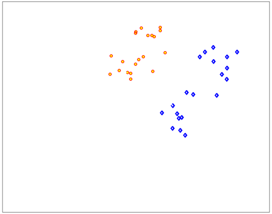
- representer theorm
- sparse solution
- efficient solver: NPSVM

SVM with non positive kernel is useful

- to be prove

Road map

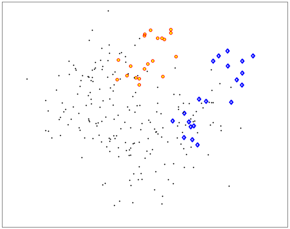
Distribution shift: the problem



Training data

$$(\mathbf{x}_i^A, y_i^A), i = 1, n$$

i.i.d. from
$\mathbb{P}_A(\mathbf{x}, y) = \mathbb{P}(y/\mathbf{x})\mathbb{P}_A(\mathbf{x})$

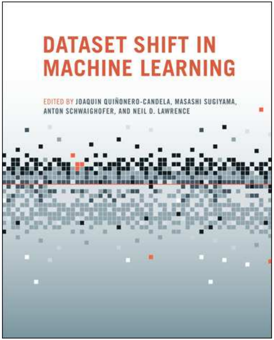


Test data

$$(\mathbf{x}_j^T, y_j^T), j = 1, \ell$$

i.i.d. from
$\mathbb{P}_T(\mathbf{x}, y) = \mathbb{P}(y/\mathbf{x})\mathbb{P}_T(\mathbf{x})$

Distribution shift: references

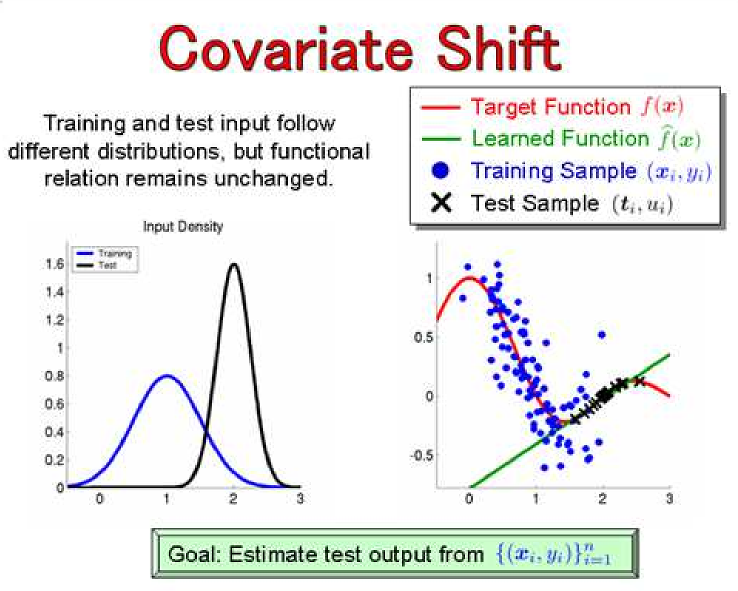


- Masashi Sugiyama (Tokyo Institute of Technology) Density ratio estimation methods: Tutorial in ACML2009[a]
- Arthur Greton (Max Planck Institute for Biological Cybernetics) Covariate Shift by Kernel Mean Matching Workshop at NIPS'09[b]
- Mahesan Niranjan (University of Southampton): application to intrusion detection

---

[a] http://lamda.nju.edu.cn/conf/acml09/files/invited_sugi.pdf

[b] http://videolectures.net/nipsworkshops09_gretton_cskm/



From M. Sugiyama Tutorial in ACML2009

Density ratio estimation principle

$$
\begin{aligned}
\min_{f\in\mathcal{H}} \mathbb{E}(J(X,Y)) &= \min_{f\in\mathcal{H}} \int_{\mathbf{x}}\int_{y} J(\mathbf{x},y)\mathbb{P}_T(\mathbf{x},y)\, d\mathbf{x}dy \\
&= \min_{f\in\mathcal{H}} \int_{\mathbf{x}}\int_{y} J(\mathbf{x},y)\mathbb{P}(y|\mathbf{x})\mathbb{P}_T(\mathbf{x})\, d\mathbf{x}dy && \text{factorize} \\
&= \min_{f\in\mathcal{H}} \int_{\mathbf{x}}\left(\int_{y} J(\mathbf{x},y)\mathbb{P}(y|\mathbf{x})dy\right)\mathbb{P}_T(\mathbf{x})\, d\mathbf{x} && \text{reorganize} \\
&= \min_{f\in\mathcal{H}} \int_{\mathbf{x}}\left(\int_{y} J(\mathbf{x},y)\mathbb{P}(y|\mathbf{x})dy\right)\mathbb{P}_T(\mathbf{x})\frac{\mathbb{P}_A(\mathbf{x})}{\mathbb{P}_A(\mathbf{x})}\, d\mathbf{x} && \mathbb{P}_A(\mathbf{x}) \neq 0 \\
&= \min_{f\in\mathcal{H}} \int_{\mathbf{x}}\left(\int_{y} J(\mathbf{x},y)\mathbb{P}(y|\mathbf{x})dy\right)\mathbb{P}_T(\mathbf{x})\frac{\mathbb{P}_A(\mathbf{x})}{\mathbb{P}_A(\mathbf{x})}\, d\mathbf{x} \\
&= \min_{f\in\mathcal{H}} \int_{\mathbf{x}}\left(\int_{y} J(\mathbf{x},y)\mathbb{P}(y|\mathbf{x})dy\right)w(\mathbf{x})\mathbb{P}_A(\mathbf{x})\, d\mathbf{x} && w(\mathbf{x}) = \frac{\mathbb{P}_T(\mathbf{x})}{\mathbb{P}_A(\mathbf{x})}
\end{aligned}
$$

Importance weighting

$$
\min_{f\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} J(\mathbf{x}_i, y_i)w(\mathbf{x}_i) \qquad\qquad w(\mathbf{x}_i) = \frac{\mathbb{P}_T(\mathbf{x}_i)}{\mathbb{P}_A(\mathbf{x}_i)}
$$

Density ratio estimation principle

the algorithm

1. estimate $w(\mathbf{x}_i) = \frac{\mathbb{P}_T(\mathbf{x}_i)}{\mathbb{P}_A(\mathbf{x}_i)}$

2. solve a weighted version of our favorite learning algorithm

$$(\mathcal{P}W) \begin{cases} \min\limits_{f\in\mathcal{H},\alpha_0,\xi\in\mathbb{R}^n} & \frac{1}{2}\|f\|^2 + \frac{C}{p}\sum\limits_{i=1}^{n} w(x_i)\xi_i^p \\ \text{with} & y_i\big(f(\mathbf{x}_i)+\alpha_0\big) \geq 1-\xi_i, \quad \xi_i \geq 0, \quad i=1,n \end{cases}$$

p = 1: L1 weighted SVM

$$\begin{cases} \max\limits_{\boldsymbol{\alpha}\in\mathbb{R}^n} & -\frac{1}{2}\boldsymbol{\alpha}^\top G\boldsymbol{\alpha} + \boldsymbol{\alpha}^\top \mathbb{1} \\ \text{with} & \boldsymbol{\alpha}^\top \mathbf{y} = 0 \\ \text{and} & 0 \leq \alpha_i \leq C w_i \quad i=1,n \end{cases}$$

p = 2: L2 weighted SVM

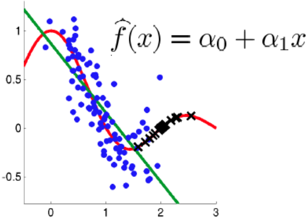
$$\begin{cases} \max\limits_{\boldsymbol{\alpha}\in\mathbb{R}^n} & -\frac{1}{2}\boldsymbol{\alpha}^\top \left(G + \frac{1}{C}W\right)\boldsymbol{\alpha} + \boldsymbol{\alpha}^\top \mathbb{1} \\ \text{with} & \boldsymbol{\alpha}^\top \mathbf{y} = 0 \\ \text{and} & 0 \leq \alpha_i \quad i=1,n \end{cases}$$

Adaptation Using Density Ratios

Shimodaira (JSPI2000), Sugiyama & Müller (ICANN2005, Stat&Deci2005)

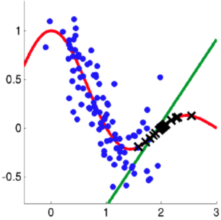
- Ordinary least-squares is not consistent.

$$\min_{\boldsymbol{\alpha}} \left[\sum_{i=1}^{n} \left(\hat{f}(\boldsymbol{x}_i) - y_i \right)^2 \right]$$



- Density-ratio weighted least-squares is consistent.

$$\min_{\boldsymbol{\alpha}} \left[\sum_{i=1}^{n} \frac{p_{\mathrm{test}}(\boldsymbol{x}_i)}{p_{\mathrm{train}}(\boldsymbol{x}_i)} \left(\hat{f}(\boldsymbol{x}_i) - y_i \right)^2 \right]$$



From M. Sugiyama Tutorial in ACML2009

Estimating the weights

$$w(\mathbf{x}_i) = \frac{\mathbb{P}_T(\mathbf{x}_i)}{\mathbb{P}_A(\mathbf{x}_i)}$$

- estimating the distributions –BAD–
  1. Parzen (or other) estimate on test data $\widehat{\mathbb{P}}_T(\mathbf{x})$
  2. Parzen (or other) estimate on training data $\widehat{\mathbb{P}}_A(\mathbf{x})$
  3. $\widehat{w}(\mathbf{x}) = \frac{\widehat{\mathbb{P}}_T(\mathbf{x})}{\widehat{\mathbb{P}}_A(\mathbf{x})}$
- Direct estimation: $\mathbf{w}(\mathbf{x}) = \frac{\mathbb{P}_T(\mathbf{x})}{\mathbb{P}_A(\mathbf{x})} \iff \mathbf{w}(\mathbf{x})\, \mathbb{P}_A(\mathbf{x}) = \mathbb{P}_T(\mathbf{x})$
  - Kullback-Leibler Importance Estimation Procedure

$$\min_{\widehat{w}} \; KL\big(\mathbb{P}_T(\mathbf{x}) \,||\, \widehat{w}(\mathbf{x})\, \mathbb{P}_A(\mathbf{x})\big)$$

  - Least-Squares Importance Fitting

$$\min_{\widehat{w}} \; \mathbb{E}_A \left(\widehat{w}(\mathbf{x}) - \frac{\mathbb{P}_T(\mathbf{x}_i)}{\mathbb{P}_A(\mathbf{x}_i)} \right)^2$$

Estimating the weights

$$w(\mathbf{x}_i) = \frac{\mathbb{P}_T(\mathbf{x}_i)}{\mathbb{P}_A(\mathbf{x}_i)}$$

- estimating the distributions –BAD–
  1. Parzen (or other) estimate on test data $\widehat{\mathbb{P}}_T(\mathbf{x})$
  2. Parzen (or other) estimate on training data $\widehat{\mathbb{P}}_A(\mathbf{x})$
  3. $\widehat{w}(\mathbf{x}) = \frac{\widehat{\mathbb{P}}_T(\mathbf{x})}{\widehat{\mathbb{P}}_A(\mathbf{x})}$
- Direct estimation: $\mathbf{w}(\mathbf{x}) = \frac{\mathbb{P}_T(\mathbf{x})}{\mathbb{P}_A(\mathbf{x})} \iff \mathbf{w}(\mathbf{x})\,\mathbb{P}_A(\mathbf{x}) = \mathbb{P}_T(\mathbf{x})$
  - Kullback-Leibler Importance Estimation Procedure

$$\min_{\widehat{w}} \; KL\big(\mathbb{P}_T(\mathbf{x}) \,||\, \widehat{w}(\mathbf{x})\,\mathbb{P}_A(\mathbf{x})\big)$$

  - Least-Squares Importance Fitting

$$\min_{\widehat{w}} \; \mathbb{E}_A \left(\widehat{w}(\mathbf{x}) - \frac{\mathbb{P}_T(\mathbf{x}_i)}{\mathbb{P}_A(\mathbf{x}_i)}\right)^2$$

Estimating the weights: empirical distribution

- Empirical distributions

$$\widehat{\mathbb{P}}_T(\mathbf{x}) = \frac{1}{\ell}\sum_{j=1}^{\ell} \delta_{\mathbf{x}_j} \qquad\qquad \widehat{\mathbb{P}}_A(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \delta_{\mathbf{x}_i}$$

- test data (T)

$$\min_{\widehat{w}} \int_{\mathbf{x}} \varphi\big(\widehat{w}(\mathbf{x})\big)\ \mathbb{P}_T(\mathbf{x})\ d\mathbf{x} \qquad \longrightarrow \qquad \min_{\widehat{w}} \frac{1}{\ell}\sum_{j=1}^{\ell} \varphi\big(\widehat{w}(\mathbf{x}_j)\big)$$

- training data (A)

$$\min_{\widehat{w}} \int_{\mathbf{x}} \varphi\big(\widehat{w}(\mathbf{x})\big)\ \mathbb{P}_A(\mathbf{x})\ d\mathbf{x} \qquad \longrightarrow \qquad \min_{\widehat{w}} \frac{1}{n}\sum_{i=1}^{n} \varphi\big(\widehat{w}(\mathbf{x}_i)\big)$$

Kullback-Leibler Importance Estimation

$$\begin{cases} \min\limits_{\widehat{w}} & KL\big(\mathbb{P}_T(\mathbf{x}) \,||\, \widehat{w}(\mathbf{x})\,\mathbb{P}_A(\mathbf{x})\big) \\ \text{with} & \displaystyle\int_{\mathbf{x}} \widehat{w}(x)\,\mathbb{P}_A(\mathbf{x})\,d\mathbf{x} = 1 \\ \text{and} & 0 \leq \widehat{w}(\mathbf{x}) \quad \forall \mathbf{x} \in \mathcal{X} \end{cases}$$

$$\begin{aligned} KL\big(\mathbb{P}_T(\mathbf{x}) \,||\, \widehat{w}(\mathbf{x})\,\mathbb{P}_A(\mathbf{x})\big) &= \int_{\mathbf{x}} \mathbb{P}_T(\mathbf{x}) \log \frac{\mathbb{P}_T(\mathbf{x})}{\widehat{w}(\mathbf{x})\,\mathbb{P}_A(\mathbf{x})}\,d\mathbf{x} \\ &= \underbrace{\int_{\mathbf{x}} \mathbb{P}_T(\mathbf{x}) \log \frac{\mathbb{P}_T(\mathbf{x})}{\mathbb{P}_A(\mathbf{x})}}_{\textit{constant}} - \int_{\mathbf{x}} \mathbb{P}_T(\mathbf{x}) \log \widehat{w}(\mathbf{x})\,d\mathbf{x} \end{aligned}$$

$$\begin{cases} \min\limits_{\widehat{w}} & -\displaystyle\int_{\mathbf{x}} \mathbb{P}_T(\mathbf{x}) \log \widehat{w}(\mathbf{x})\,d\mathbf{x} \\ \text{with} & \displaystyle\int_{\mathbf{x}} \widehat{w}(\mathbf{x})\,\mathbb{P}_A(\mathbf{x})\,d\mathbf{x} = 1 \\ \text{and} & 0 \leq \widehat{w}(\mathbf{x}) \quad \forall \mathbf{x} \in \mathcal{X} \end{cases} \qquad \begin{cases} \min\limits_{\widehat{w}} & -\displaystyle\sum_{j=1}^{\ell} \log \widehat{w}(\mathbf{x}_j) \\ \text{with} & \displaystyle\sum_{i=1}^{n} \widehat{w}(x_i) \quad = n \\ \text{and} & 0 \leq \widehat{w}(\mathbf{x}_i) \quad i = 1, n \end{cases}$$

Kullback-Leibler Importance Estimation

Use a kernel representation

$$\widehat{w}(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}, \mathbf{x}_i)$$

$$\left\{\begin{array}{ll} \displaystyle\min_{\widehat{w} \in \mathcal{H}} & \displaystyle -\sum_{j=1}^{\ell} \log \widehat{w}(\mathbf{x}_j) \\ \text{with} & \displaystyle\sum_{i=1}^{n} \widehat{w}(x_i) = n \\ \text{and} & 0 \leq \widehat{w}(\mathbf{x}_i) \quad i = 1, n \end{array}\right.
\qquad
\left\{\begin{array}{ll} \displaystyle\min_{\alpha \in \mathbb{R}^n} & \displaystyle -\sum_{j=1}^{\ell} \log K_T \alpha \\ \text{with} & \mathbf{e}^\top K_A \alpha = n \\ \text{and} & 0 \leq \alpha_i \quad i = 1, n \end{array}\right.$$

Convex (non linear, non quadratic) problem with a sparse solution

Least-Squares Importance Fitting

$$\begin{cases} \min_{\widehat{w}} & \mathbb{E}_A \left(\widehat{w}(\mathbf{x}) - \dfrac{\mathbb{P}_T(\mathbf{x}_i)}{\mathbb{P}_A(\mathbf{x}_i)} \right)^2 \\ \text{with} & \displaystyle\int_{\mathbf{x}} \widehat{w}(x)\, \mathbb{P}_A(\mathbf{x})\, d\mathbf{x} = 1 \\ \text{and} & 0 \leq \widehat{w}(\mathbf{x}) \quad \forall \mathbf{x} \in \mathcal{X} \end{cases}$$

$$\begin{aligned} \mathbb{E}_A \left(\widehat{w}(\mathbf{x}) - \tfrac{\mathbb{P}_T(\mathbf{x}_i)}{\mathbb{P}_A(\mathbf{x}_i)} \right)^2 &= \int_{\mathbf{x}} \left(\widehat{w}(\mathbf{x}) - \frac{\mathbb{P}_T(\mathbf{x}_i)}{\mathbb{P}_A(\mathbf{x}_i)} \right)^2 \mathbb{P}_A(\mathbf{x}) dx \\ &= \int_{\mathbf{x}} \widehat{w}(x)^2 \mathbb{P}_A(\mathbf{x}) dx - 2 \int \widehat{w}(\mathbf{x})\, \mathbb{P}_T(\mathbf{x}) d\mathbf{x} + \color{red}{constant} \end{aligned}$$

$$\begin{cases} \min_{\widehat{w}} & \frac{1}{2} \displaystyle\int_{\mathbf{x}} \widehat{w}(\mathbf{x})^2 \mathbb{P}_A(\mathbf{x}) - \int \widehat{w}(\mathbf{x}) \mathbb{P}_T(\mathbf{x}) \\ \text{with} & \displaystyle\int_{\mathbf{x}} \widehat{w}(\mathbf{x})\, \mathbb{P}_A(\mathbf{x})\, d\mathbf{x} = 1 \\ \text{and} & 0 \leq \widehat{w}(\mathbf{x}) \quad \forall \mathbf{x} \in \mathcal{X} \end{cases}$$

$$\begin{cases} \min_{\widehat{w}} & \frac{1}{2} \displaystyle\sum_{i=1}^{n} \widehat{w}(\mathbf{x}_i)^2 - \sum_{j=1}^{\ell} \widehat{w}(\mathbf{x}_j) \\ \text{with} & \displaystyle\sum_{i=1}^{n} \widehat{w}(x_i) = n \\ \text{and} & 0 \leq \widehat{w}(\mathbf{x}_i) \quad i = 1, n \end{cases}$$

Least-Squares Importance Fitting

Use a feature space representation in a RKHS

$$\widehat{w}(\mathbf{x}) = \sum_{k=1}^{\kappa} \alpha_k \phi_i(\mathbf{x})$$

$$\begin{cases} \min\limits_{\widehat{w}} & \frac{1}{2}\sum_{i=1}^{n} \widehat{w}(\mathbf{x}_i)^2 - \sum_{j=1}^{\ell} \widehat{w}(\mathbf{x}_j) \\ \text{with} & \sum_{i=1}^{n} \widehat{w}(x_i) \quad = n \\ \text{and} & 0 \leq \widehat{w}(\mathbf{x}_i) \quad i = 1, n \end{cases}$$

$$\begin{cases} \min\limits_{\widehat{w}} & \frac{1}{2}\alpha K \alpha - \mathbf{e}^\top \Phi_T \alpha \\ \text{with} & \mathbf{e}^\top \Phi_A \alpha \ = n \\ \text{and} & 0 \leq \alpha_i \quad i = 1, n \end{cases}$$

Convex quadratic program with a sparse solution

Conclusion

- use weights
- compute weights at a reasonable cost
- solve weighted SVM
- provides a full RKHS embedding with representer theorem

Road map

what's new since 1995

- Applications
  - kernlisation $w^\top \mathbf{x} \to \langle f, k(\mathbf{x}, .)\rangle_{\mathcal{H}} = f(\mathbf{x})$
  - kernel engineering
  - sturtured outputs
  - applications: image, text, signal, bio-info...
- Optimization
  - dual: mloss.org
  - regularization path
  - approximation
  - primal
- Statistic
  - proofs and bounds
  - model selection
    - span bound
    - multikernel: tuning (k and σ)

challenges: towards tough learning

- the size effect
  - ready to use: automatization
  - adaptative: on line context aware
  - beyond kenrels: deep learning
- Automatic and adaptive model selection
  - variable selection
  - kernel tuning: coarse-to-fine
  - hyperparametres: C, duality gap, λ
- $\mathbb{P}$ change
- Theory
  - non positive kernels
  - a more general representer theorem

biblio: kernel-machines.org

- John Shawe-Taylor and Nello Cristianini Kernel Methods for Pattern Analysis, Cambridge University Press, 2004
- Bernhard Schölkopf and Alex Smola. Learning with Kernels. MIT Press, Cambridge, MA, 2002.
- Trevor Hastie, Robert Tibshirani and Jerome Friedman, The Elements of Statistical Learning:. Data Mining, Inference, and Prediction, springer, 2001

- Léon Bottou, Olivier Chapelle, Dennis DeCoste and Jason Weston Large-Scale Kernel Machines (Neural Information Processing, MIT press 2007
- Olivier Chapelle, Bernhard Scholkopf and Alexander Zien, Semi-supervised Learning, MIT press 2006

- Vladimir Vapnik. Estimation of Dependences Based on Empirical Data. Springer Verlag, 2006, 2nd edition.
- Vladimir Vapnik. The Nature of Statistical Learning Theory. Springer, 1995.

- Grace Wahba. Spline Models for Observational Data. SIAM CBMS-NSF Regional Conference Series in Applied Mathematics vol. 59, Philadelphia, 1990
- Alain Berlinet and Christine Thomas-Agnan, Reproducing Kernel Hilbert Spaces in Probability and Statistics,Kluwer Academic Publishers, 2003
- Marc Atteia et Jean Gaches , Approximation Hilbertienne - Splines, Ondelettes, Fractales, PUG, 1999