Apprentissage, Noyaux et parcimonie

Janvier, 2009
Stat Im'09

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

Stéphane Canu
stephane.canu@litislab.eu

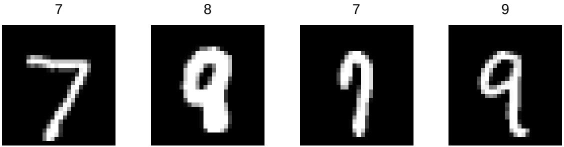
Roadmap

1. **Kernels and the learning problem**
   - Two learning problems
   - Kernelizing the linear regression
   - Kernel machines: a definition
2. **Tools: the functional framework**
   - In the beginning was the kernel
   - Kernel and hypothesis set
3. **Kernel machines**
   - Non sparse kernel machines
   - sparse kernel machines: SVM
   - practical SVM
4. **Conclusion**

Optical character recognition

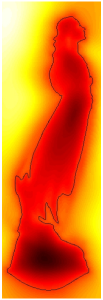
Example (The MNIST database)

- MNIST[1], data = « image-label »
- $n = 60,000$; $d = 700$; classes $= 10$
- Kernel error rate = 0.56 %,
- Best error rate = 0.4 % .



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

Implicit Surface Modelling

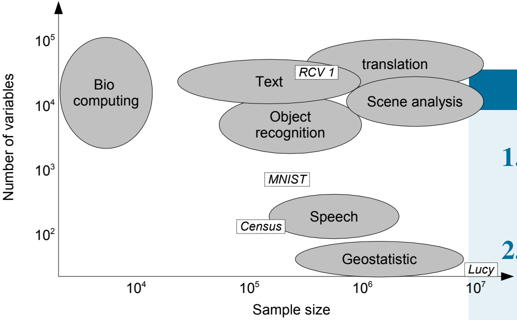


Example (*Lucy*[2])

- 14 million points with normals.
- 364,982 compactly supported basis function centres.
- kernel regression with *thin plate splines* kernels

[2] http://www.kyb.mpg.de/~walder

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)

the example of the linear least mean square

the linear model

$$y_i = \sum_{j=1}^{d} \beta_j x_{ij} + \varepsilon_i \quad , \qquad i = 1, n$$

n observations and d variables; $d < n$

$$\min_{\beta} = \sum_{i=1}^{n} \left(\sum_{j=1}^{d} x_{ij}\beta_j - y_i \right)^2 = \|X\beta - Y\|^2$$

Solution: $\widehat{\beta} = (X^\top X)^{-1} X^\top Y$

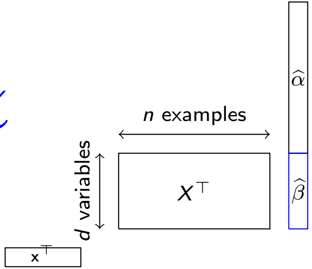
$$f(\mathbf{x}) = \mathbf{x}^\top \underbrace{(X^\top X)^{-1} X^\top Y}_{\widehat{\beta}}$$

What is the influence of each example (X rows)?

The influence of the examples

for a new input $\mathbf{x}$

$$
\begin{aligned}
f(\mathbf{x}) &= \mathbf{x}^\top (X^\top X)(X^\top X)^{-1} \underbrace{(X^\top X)^{-1} X^\top Y}_{\widehat{\beta}} \\
&= \mathbf{x}^\top X^\top \underbrace{X(X^\top X)^{-1}(X^\top X)^{-1} X^\top Y}_{\widehat{\alpha}}
\end{aligned}
$$



$$
f(\mathbf{x}) = \sum_{j=1}^{d} \widehat{\beta}_j x_j
$$

from variables to examples

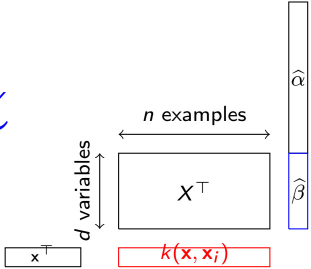
$$
\underbrace{\widehat{\alpha} = X(X^\top X)^{-1}\widehat{\beta}}_{n \text{ examples}} \quad \text{and} \quad \underbrace{\widehat{\beta} = X^\top \widehat{\alpha}}_{d \text{ variables}}
$$

What if $d \geq n$?

The influence of the examples

for a new input $\mathbf{x}$

$$
\begin{aligned}
f(\mathbf{x}) &= \mathbf{x}^\top (X^\top X)(X^\top X)^{-1} \underbrace{(X^\top X)^{-1} X^\top Y}_{\widehat{\beta}} \\
&= \mathbf{x}^\top X^\top \underbrace{X(X^\top X)^{-1}(X^\top X)^{-1} X^\top Y}_{\widehat{\alpha}}
\end{aligned}
$$



$$
f(\mathbf{x}) = \sum_{j=1}^{d} \widehat{\beta}_j x_j = \sum_{i=1}^{n} \widehat{\alpha}_i \, (\mathbf{x}^\top \mathbf{x}_i)
$$

from variables to examples

$$
\underbrace{\widehat{\alpha} = X(X^\top X)^{-1} \widehat{\beta}}_{n \text{ examples}} \qquad \text{and} \qquad \underbrace{\widehat{\beta} = X^\top \widehat{\alpha}}_{d \text{ variables}}
$$

What if $d \geq n$?

Non linear case: dictionnary *vs.* kernel

in the non linear case: use a dictionary of functions

$$\phi_j(\mathbf{x}), j = 1, p \qquad \text{with possibly} \qquad p = \infty$$

for instance polynomials, wavelets... (assume orthogonality)

$$f(\mathbf{x}) = \sum_{j=1}^{p} \widehat{\beta}_j \phi_j(\mathbf{x}) \qquad \text{with} \qquad \widehat{\beta}_j = \sum_{i=1}^{n} y_i \phi_j(\mathbf{x}_i)$$

using linearity

$$f(\mathbf{x}) = \sum_{i=1}^{n} y_i \underbrace{\sum_{j=1}^{p} \phi_j(\mathbf{x}_i)\phi_j(\mathbf{x})}_{k(\mathbf{x}_i,\mathbf{x})}$$

$p \geq n$ so what since $k(\mathbf{x}_i, \mathbf{x}) = \sum_{j=1}^{p} \phi_j(\mathbf{x}_i)\phi_j(\mathbf{x})$

Non linear case: dictionnary *vs.* kernel

in the non linear case: use a dictionary of functions

$$\phi_j(\mathbf{x}), j = 1, p \qquad \text{with possibly} \qquad p = \infty$$

for instance polynomials, wavelets... (assume orthogonality)

$$f(\mathbf{x}) = \sum_{j=1}^{p} \widehat{\beta}_j \phi_j(\mathbf{x}) \qquad \text{with} \qquad \widehat{\beta}_j = \sum_{i=1}^{n} y_i \phi_j(\mathbf{x}_i)$$

using linearity

$$f(\mathbf{x}) = \sum_{i=1}^{n} y_i \underbrace{\sum_{j=1}^{p} \phi_j(\mathbf{x}_i)\phi_j(\mathbf{x})}_{k(\mathbf{x}_i,\mathbf{x})}$$

$p \geq n$ so what since $k(\mathbf{x}_i, \mathbf{x}) = \sum_{j=1}^{p} \phi_j(\mathbf{x}_i)\phi_j(\mathbf{x})$

closed form kernel: the quadratic kernel

The quadratic dictionary in $\mathbb{R}^d$:

$$\begin{array}{rccl} \boldsymbol{\Phi}: & \mathbb{R}^d & \to & \mathbb{R}^{p=1+d+\frac{d(d+1)}{2}} \\ & \mathbf{s} & \mapsto & \boldsymbol{\Phi} = \left(1, s_1, s_2, ..., s_d, s_1^2, s_2^2, ..., s_d^2, ..., s_i s_j, ...\right) \end{array}$$

in this case

$$\boldsymbol{\Phi}(\mathbf{s})^\top \boldsymbol{\Phi}(\mathbf{t}) = 1 + s_1 t_1 + s_2 t_2 + ... + s_d t_d + s_1^2 t_1^2 + ... + s_d^2 t_d^2 + ... + s_i s_j t_i t_j + ...$$

The quadratic kenrel: $\mathbf{s}, \mathbf{t} \in \mathbb{R}^d, \quad k(\mathbf{s}, \mathbf{t}) = \left(\mathbf{s}^\top \mathbf{t} + 1\right)^2 = 1 + 2\mathbf{s}^\top \mathbf{t} + \left(\mathbf{s}^\top \mathbf{t}\right)^2$

computes the dot product of the reweighted dictionary:

$$\begin{array}{rccl} \boldsymbol{\Phi}: & \mathbb{R}^d & \to & \mathbb{R}^{p=1+d+\frac{d(d+1)}{2}} \\ & \mathbf{s} & \mapsto & \boldsymbol{\Phi} = \left(1, \sqrt{2}s_1, \sqrt{2}s_2, ..., \sqrt{2}s_d, s_1^2, s_2^2, ..., s_d^2, ..., \sqrt{2}s_i s_j, ...\right) \end{array}$$

$p = 1 + d + \frac{d(d+1)}{2}$ multiplications *vs.* $d + 1$

use kernel to save computration

closed form kernel: the quadratic kernel

The quadratic dictionary in $\mathbb{R}^d$:

$$\begin{array}{rccl} \mathbf{\Phi}: & \mathbb{R}^d & \to & \mathbb{R}^{p=1+d+\frac{d(d+1)}{2}} \\ & \mathbf{s} & \mapsto & \mathbf{\Phi} = \left(1, s_1, s_2, ..., s_d, s_1^2, s_2^2, ..., s_d^2, ..., s_i s_j, ...\right) \end{array}$$

in this case

$$\mathbf{\Phi}(\mathbf{s})^\top \mathbf{\Phi}(\mathbf{t}) = 1 + s_1 t_1 + s_2 t_2 + ... + s_d t_d + s_1^2 t_1^2 + ... + s_d^2 t_d^2 + ... + s_i s_j t_i t_j + ...$$

The quadratic kenrel: $\mathbf{s}, \mathbf{t} \in \mathbb{R}^d, \quad k(\mathbf{s}, \mathbf{t}) \begin{array}{l} = \left(\mathbf{s}^\top \mathbf{t} + 1\right)^2 \\ = 1 + 2\mathbf{s}^\top \mathbf{t} + \left(\mathbf{s}^\top \mathbf{t}\right)^2 \end{array}$

computes the dot product of the reweighted dictionary:

$$\begin{array}{rccl} \mathbf{\Phi}: & \mathbb{R}^d & \to & \mathbb{R}^{p=1+d+\frac{d(d+1)}{2}} \\ & \mathbf{s} & \mapsto & \mathbf{\Phi} = \left(1, \sqrt{2}s_1, \sqrt{2}s_2, ..., \sqrt{2}s_d, s_1^2, s_2^2, ..., s_d^2, ..., \sqrt{2}s_i s_j, ...\right) \end{array}$$

$p = 1 + d + \frac{d(d+1)}{2}$ multiplications *vs.* $d + 1$

use kernel to save computration

closed form kernel: the quadratic kernel

The quadratic dictionary in $\mathbb{R}^d$:

$$\begin{array}{rccl} \boldsymbol{\Phi}: & \mathbb{R}^d & \to & \mathbb{R}^{p=1+d+\frac{d(d+1)}{2}} \\ & \mathbf{s} & \mapsto & \boldsymbol{\Phi} = \left(1, s_1, s_2, ..., s_d, s_1^2, s_2^2, ..., s_d^2, ..., s_i s_j, ...\right) \end{array}$$

in this case

$$\boldsymbol{\Phi}(\mathbf{s})^\top \boldsymbol{\Phi}(\mathbf{t}) = 1 + s_1 t_1 + s_2 t_2 + ... + s_d t_d + s_1^2 t_1^2 + ... + s_d^2 t_d^2 + ... + s_i s_j t_i t_j + ...$$

The quadratic kenrel: $\mathbf{s}, \mathbf{t} \in \mathbb{R}^d, \quad k(\mathbf{s}, \mathbf{t}) \begin{array}{l} = \left(\mathbf{s}^\top \mathbf{t} + 1\right)^2 \\ = 1 + 2\mathbf{s}^\top \mathbf{t} + \left(\mathbf{s}^\top \mathbf{t}\right)^2 \end{array}$

computes the dot product of the reweighted dictionary:

$$\begin{array}{rccl} \boldsymbol{\Phi}: & \mathbb{R}^d & \to & \mathbb{R}^{p=1+d+\frac{d(d+1)}{2}} \\ & \mathbf{s} & \mapsto & \boldsymbol{\Phi} = \left(1, \sqrt{2}s_1, \sqrt{2}s_2, ..., \sqrt{2}s_d, s_1^2, s_2^2, ..., s_d^2, ..., \sqrt{2}s_i s_j, ...\right) \end{array}$$

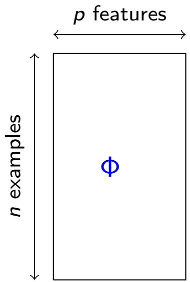
$p = 1 + d + \frac{d(d+1)}{2}$ multiplications *vs.* $d + 1$

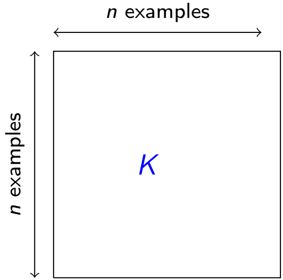
use kernel to save computration

kernel: features throught pairwise comparizons

$\mathbf{x}$ $\qquad$ $\phi(\mathbf{x})$

e.g. a text $\qquad$ *e.g.* BOW





$$k(\mathbf{x}_i, \mathbf{x}_j) = \sum_{j=1}^{p} \phi_j(\mathbf{x}_i)\phi_j(\mathbf{x}_j)$$

$\boxed{K \text{ The matrix of } \textit{pairwise} \text{ comparizons } (\mathcal{O}(n^2))}$

Kernel machines

use a kernel as a dictionary

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}, \mathbf{x}_i)$$

- α_i influence of example i — depends on y_i
- $k(\mathbf{x}, \mathbf{x}_i)$ the kernel — do NOT depend on y_i

Definition (Kernel)

a function k from $\mathcal{X} \times \mathcal{X}$ onto $\mathbb{R}$.

semi-parametric version: given the family $q_j(\mathbf{x}), j = 1, p$

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}, \mathbf{x}_i) + \sum_{j=1}^{p} \beta_j q_j(\mathbf{x})$$

Kernel machines

use a kernel as a dictionary

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}, \mathbf{x}_i)$$

- α_i influence of example i — depends on y_i
- $k(\mathbf{x}, \mathbf{x}_i)$ the kernel — do NOT depend on y_i

Definition (Kernel)

a function k from $\mathcal{X} \times \mathcal{X}$ onto $\mathbb{R}$.

semi-parametric version: given the family $q_j(\mathbf{x}),\ j = 1, p$

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}, \mathbf{x}_i) + \sum_{j=1}^{p} \beta_j q_j(\mathbf{x})$$

Kernel machines

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.

Exemples

$$\mathcal{A}(x) = \sum_{i=1}^{n} \alpha_i (x - x_i)_+^3 + \beta_0 + \beta_1 x \qquad \text{splines}$$

$$\mathcal{A}(\mathbf{x}) = \operatorname{sign}\Big(\sum_{i \in I} \alpha_i \exp^{-\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{b}} + \beta_0\Big) \qquad \text{SVM}$$

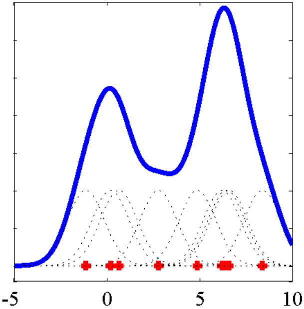
$$\mathbb{P}(y|\mathbf{x}) = \frac{1}{Z} \exp\Big(\sum_{i \in I} \alpha_i \mathbb{1}_{\{y=y_i\}} (\mathbf{x}^\top \mathbf{x}_i + b)^2\Big) \qquad \text{exponential family}$$

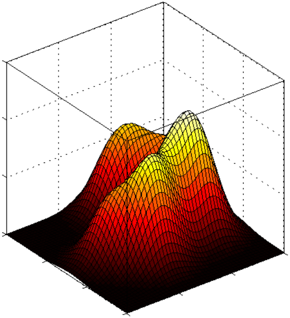
example of kernel machine: the parzen estimate (1960)

assume the kernel is normalized: $\forall \mathbf{s} \in \mathcal{X}, \quad \int_{\mathcal{X}} k(\mathbf{x}, \mathbf{s}) d\mathbf{x} = 1$

for a given data set $(\mathbf{x}_1, ..., \mathbf{x}_i, ..., \mathbf{x}_n)$ the parzen window estimate is

$$\widehat{\mathbb{P}}(x) = \frac{1}{n} \sum_{i=1}^{n} k(\mathbf{x}, \mathbf{x}_i)$$





derive:
- potential based classification rule
- the Nadaraya-Watson regression estimator

Different kernel machines

- different kernels

  kernel machines are defined for (almost) ANY kernel
  kernel machines are defined for (almost) ANY input
- kernel machines are defined throught the cost functions
  - task dependent criterion:
    - classif (SVM, K Log. reg), regression (SVR, splines)
    - ranking, clustering (OCSVM), semi supervised (Trans. SVM)
    - dim. reduction (KPCA, KPLS), sources separation (KICA)...
  - penalty term
  - sparse / non sparse : $I_0 = \{\alpha_i = 0\}$

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}, \mathbf{x}_i) + \sum_{j=1}^{p} \beta_j q_j(\mathbf{x})$$

- different implementations (algorithms)

  introducing sparsity

linear algorithm $\rightarrow$ *kernelization* $\rightarrow$ sparsity

Different kernel machines

- different kernels
  kernel machines are defined for (almost) ANY kernel
  kernel machines are defined for (almost) ANY input
- kernel machines are defined throught the cost functions
  - task dependent criterion:
    - classif (SVM, K Log. reg), regression (SVR, splines)
    - ranking, clustering (OCSVM), semi supervised (Trans. SVM)
    - dim. reduction (KPCA, KPLS), sources separation (KICA)...
  - penalty term
  - sparse / non sparse : $I_0 = \{\alpha_i = 0\}$

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}, \mathbf{x}_i) + \sum_{j=1}^{p} \beta_j q_j(\mathbf{x})$$

- different implementations (algorithms)
  introducing sparsity

linear algorithm $\longrightarrow$ *kernelization* $\longrightarrow$ sparsity

Let's summarize

- kernel k
  - defines a relation between inputs
  - provides genericity
  - pairwise comparizons
- the definition of a kernel machine require also:
  - Hypothesis set $\mathcal{H}$
  - loss function ℓ
  - a way to get universal consistency
- efficent algorithm

Roadmap

1. **Kernels and the learning problem**
   - Two learning problems
   - Kernelizing the linear regression
   - Kernel machines: a definition
2. **Tools: the functional framework**
   - In the beginning was the kernel
   - Kernel and hypothesis set
3. **Kernel machines**
   - Non sparse kernel machines
   - sparse kernel machines: SVM
   - practical SVM
4. **Conclusion**

In the beginning was the kenrel...

Definition (Kernel)

a function of two variable k from $\mathcal{X} \times \mathcal{X}$ to $\mathbb{R}$

Definition (Positive kernel)

A kernel $k(s,t)$ on $\mathcal{X}$ is said to be positive

- if it is symetric: $k(s,t) = k(t,s)$
- an if for any finite positive interger n:

$$\forall \{\alpha_i\}_{i=1,n} \in \mathbb{R}, \forall \{\mathbf{x}_i\}_{i=1,n} \in \mathcal{X}, \quad \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) \geq 0$$

it is strictly positive if for $\alpha_i \neq 0$

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) > 0$$

Examples of positive kernels

the linear kernel: $\mathbf{s}, \mathbf{t} \in \mathbb{R}^d, \quad k(\mathbf{s}, \mathbf{t}) = \mathbf{s}^\top \mathbf{t}$

symetric: $\mathbf{s}^\top \mathbf{t} = \mathbf{t}^\top \mathbf{s}$

positive:
$$\sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) = \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j \mathbf{x}_i^\top \mathbf{x}_j = \left(\sum_{i=1}^n \alpha_i \mathbf{x}_i\right)^\top \left(\sum_{j=1}^n \alpha_j \mathbf{x}_j\right) = \left\|\sum_{i=1}^n \alpha_i \mathbf{x}_i\right\|^2$$

the product kernel: $k(\mathbf{s}, \mathbf{t}) = g(\mathbf{s}) g(\mathbf{t})$ **for some** $g : \mathbb{R}^d \to \mathbb{R}$,

symetric by construction

positive:
$$\sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) = \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j g(\mathbf{x}_i) g(\mathbf{x}_j) = \left(\sum_{i=1}^n \alpha_i g(\mathbf{x}_i)\right) \left(\sum_{j=1}^n \alpha_j g(\mathbf{x}_j)\right) = \left(\sum_{i=1}^n \alpha_i g(\mathbf{x}_i)\right)^2$$

k is positive $\Leftrightarrow$ (its square root exists) $\Leftrightarrow$ $k(\mathbf{s}, \mathbf{t}) = \langle \phi_\mathbf{s}, \phi_\mathbf{t} \rangle$

positive definite Kernel (PDK) algebra (closure)

if $k_1(\mathbf{s},\mathbf{t})$ and $k_2(\mathbf{s},\mathbf{t})$ are two positive kernels

- DPK are a convex cone: $\forall a_1 \in \mathbb{R}^+ \quad a_1 k_1(\mathbf{s},\mathbf{t}) + k_2(\mathbf{s},\mathbf{t})$
- for any measurable function ψ from $\mathcal{X}$ to $\mathbb{R}$ $\quad k(\mathbf{s},\mathbf{t}) = \psi(\mathbf{s})\psi(\mathbf{t})$
- product kernel $\quad k_1(\mathbf{s},\mathbf{t})k_2(\mathbf{s},\mathbf{t})$

proofs

- by linearity:

$$\sum_{i=1}^n\sum_{j=1}^n \alpha_i\alpha_j\big(a_1k_1(i,j)k_2(i,j)\big) = a_1\sum_{i=1}^n\sum_{j=1}^n \alpha_i\alpha_j k_1(i,j) + \sum_{i=1}^n\sum_{j=1}^n \alpha_i\alpha_j k_2(i,j)$$

- by linearity: $\displaystyle\sum_{i=1}^n\sum_{j=1}^n \alpha_i\alpha_j\big(\psi(\mathbf{x}_i)\psi(\mathbf{x}_j)\big) = \Big(\sum_{i=1}^n \alpha_i\psi(\mathbf{x}_i)\Big)\Big(\sum_{j=1}^n \alpha_j\psi(\mathbf{x}_j)\Big)$

- assuming $\exists\psi_\ell$ s.t. $k_1(\mathbf{s},\mathbf{t}) = \sum_\ell \psi_\ell(\mathbf{s})\psi_\ell(\mathbf{t})$

$$\begin{aligned}\sum_{i=1}^n\sum_{j=1}^n \alpha_i\alpha_j\; k_1(\mathbf{x}_i,\mathbf{x}_j)k_2(\mathbf{x}_i,\mathbf{x}_j) &= \sum_{i=1}^n\sum_{j=1}^n \alpha_i\alpha_j\Big(\sum_\ell \psi_\ell(\mathbf{x}_i)\psi_\ell(\mathbf{x}_j)k_2(\mathbf{x}_i,\mathbf{x}_j)\Big)\\ &= \sum_\ell\sum_{i=1}^n\sum_{j=1}^n \big(\alpha_i\psi_\ell(\mathbf{x}_i)\big)\;\big(\alpha_j\psi_\ell(\mathbf{x}_j)\big)\;k_2(\mathbf{x}_i,\mathbf{x}_j)\end{aligned}$$

N. Cristianini and J. Shawe Taylor, kernel methods for pattern analysis, 2004

Kernel engineering: building PDK

- for any polynomial with positive coef. ϕ from $\mathbb{R}$ to $\mathbb{R}$ $\qquad \phi\big(k(\mathbf{s},\mathbf{t})\big)$
- if $\Psi$is a function from $\mathbb{R}^d$ to $\mathbb{R}^d$ $\qquad k\big(\Psi(\mathbf{s}),\Psi(\mathbf{t})\big)$
- if φ from $\mathbb{R}^d$ to $\mathbb{R}^+$, is minimum in 0 $\qquad k(\mathbf{s},\mathbf{t}) = \varphi(\mathbf{s}+\mathbf{t}) - \varphi(\mathbf{s}-\mathbf{t})$
- convolution of two positive kernels is a positive kernel $\qquad K_1 \star K_2$

the Gaussian kernel is a PDK

$$\begin{aligned}\exp(-\|\mathbf{s}-\mathbf{t}\|^2) &= \exp(-\|\mathbf{s}\|^2 - \|\mathbf{t}\|^2 - 2\mathbf{s}^\top\mathbf{t}) \\ &= \exp(-\|\mathbf{s}\|^2)\exp(-\|\mathbf{t}\|^2)\exp(2\mathbf{s}^\top\mathbf{t})\end{aligned}$$

- $\mathbf{s}^\top\mathbf{t}$ is a PDK and function $\exp$ as the limit of positive series expansion, so $\exp(2\mathbf{s}^\top\mathbf{t})$ is a PDK
- $\exp(-\|\mathbf{s}\|^2)\exp(-\|\mathbf{t}\|^2)$ is a PDK as a product kernel
- the product of two PDK is a PDK

some examples of PD kernels...

type	name	$k(s,t)$
radial	gaussian	$\exp\left(-\frac{r^2}{b}\right), \quad r = \|s-t\|$
radial	laplacian	$\exp(-r/b)$
radial	rationnal	$1 - \frac{r^2}{r^2+b}$
radial	loc. gauss.	$\max\left(0, 1-\frac{r}{3b}\right)^d \exp(-\frac{r^2}{b})$
non stat.	χ^2	$\exp(-r/b), \ r = \sum_k \frac{(s_k-t_k)^2}{s_k+t_k}$
projective	polynomial	$(s^\top t)^p$
projective	affine	$(s^\top t + b)^p$
projective	cosine	$s^\top t / \|s\|\|t\|$
projective	correlation	$\exp\left(\frac{s^\top t}{\|s\|\|t\|} - b\right)$

Most of the kernels depends on a quantity b called the bandwidth

kernels for objects and structures

kernels on histograms and probability distributions

$$k(p(x), q(x)) = \int k_i\big(p(x), q(x)\big)\mathbb{P}(x)dx$$

kernel on strings

- spectral string kernel $k(\mathbf{s}, \mathbf{t}) = \sum_u \phi_u(\mathbf{s})\phi_u(\mathbf{t})$
- using sub sequences
- similarities by alignements $k(\mathbf{s}, \mathbf{t}) = \sum_\pi \exp(\beta(\mathbf{s}, \mathbf{t}, \pi))$

kernels on graphs

- the pseudo inverse of the (regularized) graph Laplacian

  $L = D - A$ $\quad$ A is the adjency matrix D the degree matrix

- diffusion kernels $\frac{1}{Z(b)} \exp^{bL}$
- subgraph kernel convolution (using random walks)

and kernels on heterogeneous data (image), HMM, automata...

different point of view about kernels

kernel and scalar product

$$k(\mathbf{s},\mathbf{t}) = \langle \phi(\mathbf{s}), \phi(\mathbf{t}) \rangle_{\mathcal{H}}$$

kernel and distance

$$d(\mathbf{s},\mathbf{t})^2 = k(\mathbf{s},\mathbf{s}) + k(\mathbf{t},\mathbf{t}) - 2k(\mathbf{s},\mathbf{t})$$

kernel and covariance: a positive matrix is a covariance matrix

$$\mathbb{P}(\mathbf{f}) = \frac{1}{Z}\exp\left(-\frac{1}{2}(\mathbf{f}-\mathbf{f}_0)^\top K^{-1}(\mathbf{f}-\mathbf{f}_0)\right)$$

if $f_0 = 0$ and $f = K\boldsymbol{\alpha}$, $\mathbb{P}(\boldsymbol{\alpha}) = \frac{1}{Z}\exp -\frac{1}{2}\boldsymbol{\alpha}^\top K\boldsymbol{\alpha}$

Kernel and regularity (green's function)

$k(\mathbf{s},\mathbf{t}) = P^*P\delta_{\mathbf{s}-\mathbf{t}}$ for some operator P (*e.g.* some differential)

Roadmap

From kernel to functions

$$\mathcal{H}_0 = \left\{ f \,\middle|\, m_f < \infty; f_j \in \mathbb{R}; t_j \in \mathcal{X},\ f(\mathbf{x}) = \sum_{j=1}^{m_f} f_j k(\mathbf{x}, t_j) \right\}$$

let define the bilinear form $\left(g(\mathbf{x}) = \sum_{i=1}^{m_g} g_i k(\mathbf{x}, s_i) \right)$:

$$\forall f, g \in \mathcal{H}_0,\ \langle f, g \rangle_{\mathcal{H}_0} = \sum_{j=1}^{m_f} \sum_{i=1}^{m_g} f_j\ g_i\ k(t_j, s_i)$$

Evaluation functional: $\forall \mathbf{x} \in \mathcal{X}$

$$f(\mathbf{x}) = \langle f(.), k(\mathbf{x}, .) \rangle_{\mathcal{H}_0}$$

from k to $\mathcal{H}$

with any postive kernel, a hypothesis set can be constructed $\mathcal{H}$ with its metric

RKHS

Definition (reproducing kernel Hibert space (RKHS))

a Hilbert space $\mathcal{H}$ embeded with the inner product $\langle ., . \rangle_{\mathcal{H}}$ is said to be with reproduicing kernel if it exists a positive kernel k such that

$$\forall s \in \mathcal{X}, k(., s) \in \mathcal{H} \text{ et } \forall f \in \mathcal{H}, \quad f(s) = \langle f(.), k(s, .) \rangle_{\mathcal{H}}$$

positive kernel $\Leftrightarrow$ RKHS

- any function is pointwise defined
- defines the inner product
- it defines the regularity (smoothness) of the hypothesis set

functional differentiation in RKHS

Let J be a functional

$$J: \begin{array}{ccc} \mathcal{H} & \rightarrow & \mathbb{R} \\ f & \mapsto & J(f) \end{array} \qquad \text{examples:} \qquad J_1(f) = \|f\|^2, J_2(f) = f(\mathbf{x}),$$

J directional derivative in direction g at point f

$$dJ(f,g) = \lim_{\varepsilon \rightarrow 0} \frac{J(f+\varepsilon g) - J(f)}{\varepsilon}$$

Gradient $\nabla_J(f)$

$$\nabla_J: \begin{array}{ccc} \mathcal{H} & \rightarrow & \mathcal{H} \\ f & \mapsto & \nabla_J(f) \end{array} \qquad \text{si} \qquad dJ(f,g) = \langle \nabla_J(f), g \rangle_{\mathcal{H}}$$

exercice: find out $\nabla_{J_1}(f)$ et $\nabla_{J_2}(f)$

other kernels (what realy matters)

- finite kernels
$$k(\mathbf{s},\mathbf{t}) = \big(\phi_1(\mathbf{s}), ..., \phi_p(\mathbf{s})\big)^\top \big(\phi_1(\mathbf{t}), ..., \phi_p(\mathbf{t})\big)$$
- Mercer kernels
  positive on a compact set $\Leftrightarrow$ $k(\mathbf{s},\mathbf{t}) = \sum_{j=1}^{p} \lambda_j \phi_j(\mathbf{s})\phi_j(\mathbf{t})$
- positive kernels
- positive semi-definite
- conditionnaly positive (for some functions p_j)
$$\forall \{\mathbf{x}_i\}_{i=1,n}, \forall \alpha_i, \sum_{i}^{n} \alpha_i p_j(\mathbf{x}_i) = 0; \quad j = 1, p, \quad \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) \geq 0$$
- symetric non positive
$$k(\mathbf{s},\mathbf{t}) = \tanh(\mathbf{s}^\top \mathbf{t} + \alpha_0)$$
- non symetric – non positive

the key property: $\nabla_{J_t}(f) = k(t, .)$ holds

C. Ong *et al*, ICML , 2004

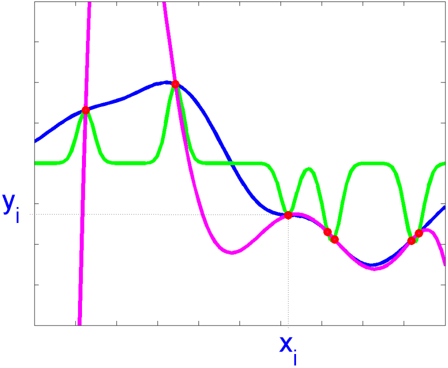
Let's summarize

- positive kernels $\Leftrightarrow$ RKHS $= \mathcal{H}$ $\Leftrightarrow$ regularity $\|f\|^2_{\mathcal{H}}$
- the key property: $\nabla_{J_t}(f) = k(t,.)$ holds not only for positive kernels $f(\mathbf{x}_i)$ exists (pointwise defined functions)
- universal consistency in RKHS
- the Gram matrix summarize the pairwise comparizons

Plan

Interpolation splines

find out $f \in \mathcal{H}$ such that $f(\mathbf{x}_i) = y_i, \qquad i = 1, ..., n$



It is an ill posed problem

Interpolation splines: minimum norm interpolation

$$\begin{cases} \min\limits_{f \in \mathcal{H}} & \frac{1}{2}\|f\|^2_{\mathcal{H}} \\ \text{such that} & f(\mathbf{x}_i) = y_i, \qquad i = 1, \dots, n \end{cases}$$

The lagrangian (α_i Lagrange multipliers)

$$L(f, \boldsymbol{\alpha}) = \frac{1}{2}\|f\|^2 - \sum_{i=1}^{n} \alpha_i \big(f(\mathbf{x}_i) - y_i\big)$$

optimality for f: $\quad \nabla_f L(f, \boldsymbol{\alpha}) = 0 \quad \Leftrightarrow \quad f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x})$

dual formulation (remove f from the lagrangian):

$$Q(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) + \sum_{i=1}^{n} \alpha_i y_i \qquad \text{solution:} \qquad \max_{\boldsymbol{\alpha} \in \mathbb{R}^n} Q(\boldsymbol{\alpha})$$

$$\boxed{K\boldsymbol{\alpha} = \mathbf{y}}$$

Interpolation splines: minimum norm interpolation

$$\begin{cases} \min\limits_{f \in \mathcal{H}} & \frac{1}{2}\|f\|^2_{\mathcal{H}} \\ \text{such that} & f(\mathbf{x}_i) = y_i, \qquad i = 1, \ldots, n \end{cases}$$

The lagrangian (α_i Lagrange multipliers)

$$L(f, \boldsymbol{\alpha}) = \frac{1}{2}\|f\|^2 - \sum_{i=1}^{n} \alpha_i \big(f(\mathbf{x}_i) - y_i\big)$$

optimality for f: $\quad \nabla_f L(f, \boldsymbol{\alpha}) = 0 \quad \Leftrightarrow \quad f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x})$

dual formulation (remove f from the lagrangian):

$$Q(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) + \sum_{i=1}^{n} \alpha_i y_i \qquad \text{solution:} \quad \max_{\boldsymbol{\alpha} \in \mathbb{R}^n} Q(\boldsymbol{\alpha})$$

$$\boxed{K\boldsymbol{\alpha} = \mathbf{y}}$$

Interpolation splines: minimum norm interpolation

$$\begin{cases} \min\limits_{f \in \mathcal{H}} & \frac{1}{2}\|f\|^2_{\mathcal{H}} \\ \text{such that} & f(\mathbf{x}_i) = y_i, \qquad i = 1, \ldots, n \end{cases}$$

The lagrangian (α_i Lagrange multipliers)

$$L(f, \boldsymbol{\alpha}) = \frac{1}{2}\|f\|^2 - \sum_{i=1}^{n} \alpha_i \big(f(\mathbf{x}_i) - y_i\big)$$

optimality for f: $\nabla_f L(f, \boldsymbol{\alpha}) = 0 \quad \Leftrightarrow \quad f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x})$

dual formulation (remove f from the lagrangian):

$$Q(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) + \sum_{i=1}^{n} \alpha_i y_i$$

solution: $\max\limits_{\boldsymbol{\alpha} \in \mathbb{R}^n} Q(\boldsymbol{\alpha})$

$$\boxed{K\boldsymbol{\alpha} = y}$$

Representer theorem

Theorem (epresenter theorem)

Let $\mathcal{H}$ be a RKHS with kernel $k(s,t)$. Let ℓ be a function from $\mathcal{X}$ to $\mathbb{R}$ (loss function) and Φ a non decreasing function from $\mathbb{R}$ to $\mathbb{R}$. If there exists a function f^*minimizing:

$$f^* = \underset{f \in \mathcal{H}}{argmin} \sum_{i=1}^{n} \ell\big(y_i, f(\mathbf{x}_i)\big) + \Phi\big(\|f\|_{\mathcal{H}}^2\big)$$

then there exists a vector $\boldsymbol{\alpha} \in \mathbb{R}^n$ such that:

$$f^*(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}, \mathbf{x}_i)$$

it can be generalized to the semi parametric case: $+\sum_{j=1}^{m} \beta_j \phi_j(\mathbf{x})$

Smooting splines

introducing the error (the slack) $\xi = f(x_i) - y_i$

$$(\mathcal{S}) \quad \begin{cases} \min\limits_{f \in \mathcal{H}} & \frac{1}{2}\|f\|_{\mathcal{H}}^2 + \frac{1}{2\lambda}\sum\limits_{i=1}^{n} \xi_i^2 \\ \text{such that} & f(x_i) = y_i + \xi_i, \qquad i = 1, n \end{cases}$$

three equivalents definitions

$$(\mathcal{S}') \quad \min_{f \in \mathcal{H}} \frac{1}{2}\sum_{i=1}^{n}\big(f(x_i) - y_i\big)^2 + \frac{\lambda}{2}\|f\|_{\mathcal{H}}^2$$

$$\begin{cases} \min\limits_{f \in \mathcal{H}} & \frac{1}{2}\|f\|_{\mathcal{H}}^2 \\ \text{such that} & \sum\limits_{i=1}^{n}\big(f(\mathbf{x}_i) - y_i\big)^2 \le C' \end{cases} \qquad \begin{cases} \min\limits_{f \in \mathcal{H}} & \sum\limits_{i=1}^{n}\big(f(\mathbf{x}_i) - y_i\big)^2 \\ \text{such that} & \|f\|_{\mathcal{H}}^2 \le C'' \end{cases}$$

using the representer theorem

$$(\mathcal{S}'') \quad \min_{\boldsymbol{\alpha} \in \mathbb{R}^n} \frac{1}{2}\|K\boldsymbol{\alpha} - \mathbf{y}\|^2 + \frac{\lambda}{2}\boldsymbol{\alpha}^\top K \boldsymbol{\alpha}$$

solution:

$$(\mathcal{S}) \Leftrightarrow (\mathcal{S}') \Leftrightarrow (\mathcal{S}'') \Leftrightarrow (K + \lambda I)\boldsymbol{\alpha} = \mathbf{y}$$

Kernel logistic regression

inspiration: the Bayes rule

$$D(\mathbf{x}) = \text{sign}\big(f(\mathbf{x}) + \alpha_0\big) \quad \Longrightarrow \quad \log\left(\frac{\mathbb{P}(Y=1|\mathbf{x})}{\mathbb{P}(Y=-1|\mathbf{x})}\right) = f(\mathbf{x}) + \alpha_0$$

probabilities:

$$\mathbb{P}(Y=1|\mathbf{x}) = \frac{\exp^{f(\mathbf{x})+\alpha_0}}{1+\exp^{f(\mathbf{x})+\alpha_0}} \qquad \mathbb{P}(Y=-1|\mathbf{x}) = \frac{1}{1+\exp^{f(\mathbf{x})+\alpha_0}}$$

Rademacher distribution

$$\mathcal{L}(x_i, y_i, f, \alpha_0) = \mathbb{P}(Y=1|\mathbf{x}_i)^{\frac{y_i+1}{2}} \left(1 - \mathbb{P}(Y=1|\mathbf{x}_i)\right)^{\frac{1-y_i}{2}}$$

penalized likelihood

$$\begin{aligned} J(f, \alpha_0) &= -\sum_{i=1}^{n} \log\big(\mathcal{L}(x_i, y_i, f, \alpha_0)\big) + \frac{\lambda}{2}\|f\|_{\mathcal{H}}^2 \\ &= \sum_{i=1}^{n} \log\left(1 + \exp^{-y_i(f(\mathbf{x}_i)+\alpha_0)}\right) + \frac{\lambda}{2}\|f\|_{\mathcal{H}}^2 \end{aligned}$$

Kernel logistic regression (2)

$$(\mathcal{R}) \quad \begin{cases} \min\limits_{f \in \mathcal{H}} & \frac{1}{2}\|f\|^2_{\mathcal{H}} + \frac{1}{\lambda}\sum\limits_{i=1}^{n} \log\left(1+\exp^{-\xi_i}\right) \\ \text{with} & \xi_i = y_i\left(f(\mathbf{x}_i) + \alpha_0\right), \qquad i = 1, n \end{cases}$$

Representer theorem

$$J(\alpha, \alpha_0) = \mathbb{I}^\top \log\left(\mathbb{I} + \exp^{\text{diag}(\mathbf{y})K\boldsymbol{\alpha}+\alpha_0\mathbf{y}}\right) + \frac{\lambda}{2}\boldsymbol{\alpha}^\top K\boldsymbol{\alpha}$$

gradient vector anf Hessian matrix:

$$\nabla_{\boldsymbol{\alpha}} J(\alpha, \alpha_0) = K\left(\mathbf{y} - (2\mathbf{p} - \mathbb{I})\right) + \lambda K\boldsymbol{\alpha}$$

$$H_{\boldsymbol{\alpha}} J(\alpha, \alpha_0) = K\text{diag}\left(\mathbf{p}(\mathbb{I} - \mathbf{p})\right)K + \lambda K$$

solve the problem using Newton iterations

$$\boldsymbol{\alpha}^{\text{new}} = \boldsymbol{\alpha}^{\text{old}} + \left(K\text{diag}\left(\mathbf{p}(\mathbb{I} - \mathbf{p})\right)K + \lambda K\right)^{-1} K\left(\mathbf{y} - (2\mathbf{p} - \mathbb{I}) + \lambda\boldsymbol{\alpha}\right)$$

Let's summarize

- pros
  - Universality
  - from $\mathcal{H}$ to $\mathbb{R}^n$ using the representer theorem
  - no (explicit) curse of dimensionality
- splines $\mathcal{O}(n^3)$ (can be reduced to $\mathcal{O}(n^2)$)
- logistic regression $\mathcal{O}(kn^3)$ (can be reduced to $\mathcal{O}(kn^2)$)
- no scalability!

sparsness comes to the rescue!

SVM: the separable case (no noise)

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

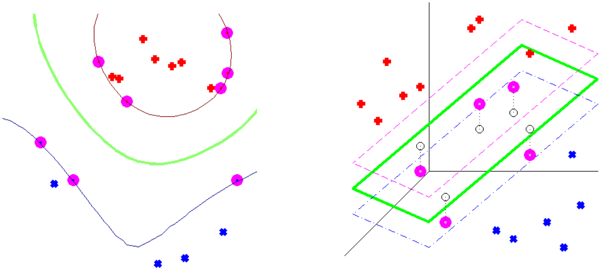
3 ways to represent function f

$$\underbrace{f(x)}_{\text{in the RKHS } \mathcal{H}} = \underbrace{\sum_{j=1}^{d} w_j\, \phi_j(\mathbf{x})}_{d \text{ features}} = \underbrace{\sum_{i=1}^{n} \alpha_i\, y_i\, k(\mathbf{x},\mathbf{x}_i)}_{n \text{ data points}}$$

$$\begin{cases} \min\limits_{\mathbf{w},\alpha_0} & \frac{1}{2}\|\mathbf{w}\|^2_{\mathbb{R}^d} = \frac{1}{2}\,\mathbf{w}^\top\mathbf{w} \\ \text{with} & y_i\big(\mathbf{w}^\top\phi(\mathbf{x}_i)+\alpha_0\big) \geq 1 \end{cases} \Leftrightarrow \begin{cases} \min\limits_{\boldsymbol{\alpha},\alpha_0} & \frac{1}{2}\,\boldsymbol{\alpha}^\top K\boldsymbol{\alpha} \\ \text{with} & y_i\big(\boldsymbol{\alpha}^\top K(:,i)+\alpha_0\big) \geq 1 \end{cases}$$

using relevant features...

a data point becomes a function $\mathbf{x} \longrightarrow k(\mathbf{x}, .)$



input space representation: x　　　feature space: k(x,.)

SVM dual formulation

Dual formulation

$$\begin{cases} \max\limits_{\alpha \in \mathbb{R}^n} & -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j k(\mathbf{x}_i,\mathbf{x}_j) + \sum_{i=1}^{n}\alpha_i \\ \text{with} & \sum_{i=1}^{n}\alpha_i y_i = 0 \quad \text{and } 0 \leq \alpha_i, \quad i = 1, n \end{cases}$$

The dual formulation gives a quadratic program (QP)

$$\begin{cases} \min\limits_{\boldsymbol{\alpha} \in \mathbb{R}^n} & \frac{1}{2}\boldsymbol{\alpha}^\top G\boldsymbol{\alpha} - \mathbb{1}^\top\boldsymbol{\alpha} \\ \text{with} & \boldsymbol{\alpha}^\top\mathbf{y} = 0 \quad \text{and } 0 \leq \boldsymbol{\alpha} \end{cases}$$

with $G_{ij} = y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)$

with the linear kernel $f(\mathbf{x}) = \sum_{i=1}^{n}\alpha_i y_i(\mathbf{x}^\top\mathbf{x}_i) = \sum_{j=1}^{d}\beta_j x_j$
when d is small wrt. n primal may be interesting.

the general case: C-SVM

Primal formulation

$$(\mathcal{P}) \begin{cases} \displaystyle\min_{f\in\mathcal{H},\alpha_0,\xi\in\mathbb{R}^n} & \frac{1}{2}\|f\|^2 + \frac{C}{p}\sum_{i=1}^n \xi_i^p \\ \text{such that} & y_i\big(f(\mathbf{x}_i)+\alpha_0\big) \geq 1-\xi_i, \quad \xi_i \geq 0, \quad i=1,n \end{cases}$$

C is the *regularization path* parameter (to be tuned)

$p=1$, L_1 SVM

$$\begin{cases} \displaystyle\max_{\boldsymbol{\alpha}\in\mathbb{R}^n} & -\frac{1}{2}\boldsymbol{\alpha}^\top H\boldsymbol{\alpha} + \boldsymbol{\alpha}^\top \mathbb{1} \\ \text{such that} & \boldsymbol{\alpha}^\top\mathbf{y} = 0 \text{ and } 0 \leq \alpha_i \leq C \quad i=1,n \end{cases}$$

$p=2$, L_2 SVM

$$\begin{cases} \displaystyle\max_{\boldsymbol{\alpha}\in\mathbb{R}^n} & -\frac{1}{2}\boldsymbol{\alpha}^\top\left(H+\frac{1}{C}I\right)\boldsymbol{\alpha} + \boldsymbol{\alpha}^\top \mathbb{1} \\ \text{such that} & \boldsymbol{\alpha}^\top\mathbf{y} = 0 \text{ and } 0 \leq \alpha_i \quad i=1,n \end{cases}$$

the regularization path: is the set of solutions $\boldsymbol{\alpha}(C)$ when C varies

The importance of being support

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x})$$

data point	α	constraint value
$\mathbf{x}_i$ *useless*	$\alpha_i = 0$	$y_i\big(f(\mathbf{x}_i) + \alpha_0\big) > 1$
$\mathbf{x}_i$ *support*	$\alpha_i > 0$	$y_i\big(f(\mathbf{x}_i) + \alpha_0\big) = 1$

Table: When a data point is « support » it lies exacty on the margin.

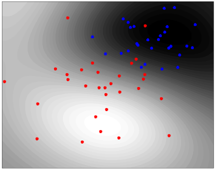
here lies the efficency of the algorithm (and its complexity)!

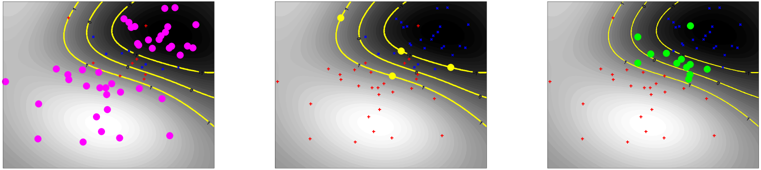
sparsness: $\alpha_i = 0$

Data groups: illustration

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}, \mathbf{x}_i) + \alpha_0$$

$$D(x) = \text{sign}\big(f(\mathbf{x})\big)$$





useless data	important data	suspicious data
well classified	support	
$\alpha = 0$	$0 < \alpha < C$	$\alpha = C$

Two more ways to derivate SVM

Using the hinge loss

$$\min_{f \in \mathcal{H}, \alpha_0 \in \mathbb{R}} \frac{1}{p} \sum_{i=1}^{n} \max\big(0, 1 - y_i(f(\mathbf{x}_i) + \alpha_0)\big)^p + \frac{1}{2C}\|f\|^2$$

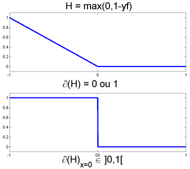
Minimizing the distance between the convex hulls

$$\begin{cases} \min\limits_{\alpha} & \|u - v\|_{\mathcal{H}}^2 \\ \text{with} & u(\mathbf{x}) = \sum\limits_{\{i|y_i=1\}} \alpha_i k(\mathbf{x}_i, \mathbf{x}), \; v(\mathbf{x}) = \sum\limits_{\{i|y_i=-1\}} \alpha_i k(\mathbf{x}_i, \mathbf{x}) \\ \text{and} & \sum\limits_{\{i|y_i=1\}} \alpha_i = 1, \; \sum\limits_{\{i|y_i=-1\}} \alpha_i = 1, 0 \le \alpha_i \quad i = 1, n \end{cases}$$

$$f(\mathbf{x}) = \frac{2}{\|u - v\|_{\mathcal{H}}^2}\big(u(\mathbf{x}) - v(\mathbf{x})\big) \text{ and } \alpha_0 = \frac{\|u\|_{\mathcal{H}}^2 - \|v\|_{\mathcal{H}}^2}{\|u - v\|_{\mathcal{H}}^2}$$

Regularization path for SVM

$$\min_{f \in \mathcal{H}} \sum_{i=1}^{n} \max(1 - y_i f(\mathbf{x}_i), 0) + \frac{\lambda}{2} \|f\|^2_{\mathcal{H}}$$



I_α is the set of support vectors s.t. $y_i f(\mathbf{x}_i) = 1$;

$$\nabla_f J(f) = \sum_{i \in I_\alpha} \alpha_i y_i K(\mathbf{x}_i, .) + \sum_{i \in I_1} y_i K(\mathbf{x}_i, .) + \lambda\, f(.) \quad \text{with} \quad \alpha_i = \partial H(\mathbf{x}_i)$$

in particular at point $\mathbf{x}_j \in I_\alpha$ $\left(f_o(\mathbf{x}_j) = f_n(\mathbf{x}_j) = y_j\right)$

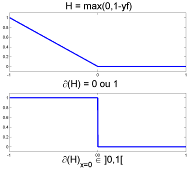
$$\begin{array}{rcl} \sum_{i \in I_\alpha} \alpha_{io} y_i K(\mathbf{x}_i, \mathbf{x}_j) & = & -\sum_{i \in I_1} y_i K(\mathbf{x}_i, \mathbf{x}_j) - \lambda_o\, y_j \\ \sum_{i \in I_\alpha} \alpha_{in} y_i K(\mathbf{x}_i, \mathbf{x}_j) & = & -\sum_{i \in I_1} y_i K(\mathbf{x}_i, \mathbf{x}_j) - \lambda_n\, y_j \\ \hline G(\alpha_n - \alpha_o) & = & (\lambda_o - \lambda_n)\mathbf{y} \qquad \text{avec} \qquad G_{ij} = y_i K(\mathbf{x}_i, \mathbf{x}_j) \end{array}$$

$$\alpha_n = \alpha_o + (\lambda_o - \lambda_n)\mathbf{w}$$

$$\mathbf{w} = (G)^{-1}\mathbf{y}$$

Regularization path for SVM

$$\min_{f \in \mathcal{H}} \sum_{i=1}^{n} \max(1 - y_i f(\mathbf{x}_i), 0) + \frac{\lambda}{2} \|f\|^2_{\mathcal{H}}$$



I_α is the set of support vectors s.t. $y_i f(\mathbf{x}_i) = 1$;

$$\nabla_f J(f) = \sum_{i \in I_\alpha} \alpha_i y_i K(\mathbf{x}_i, .) + \sum_{i \in I_1} y_i K(\mathbf{x}_i, .) + \lambda\, f(.) \quad \text{with} \quad \alpha_i = \partial H(\mathbf{x}_i)$$

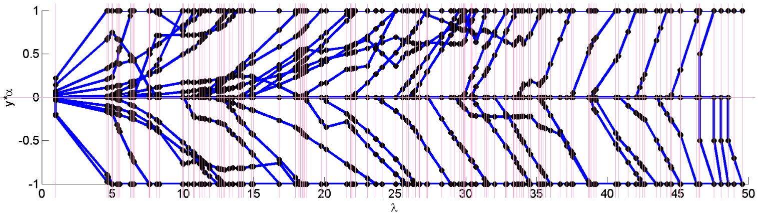
in particular at point $\mathbf{x}_j \in I_\alpha$ $\left(f_o(\mathbf{x}_j) = f_n(\mathbf{x}_j) = y_j\right)$

$$\begin{array}{rcl} \sum_{i \in I_\alpha} \alpha_{io} y_i K(\mathbf{x}_i, \mathbf{x}_j) & = & -\sum_{i \in I_1} y_i K(\mathbf{x}_i, \mathbf{x}_j) - \lambda_o\, y_j \\ \sum_{i \in I_\alpha} \alpha_{in} y_i K(\mathbf{x}_i, \mathbf{x}_j) & = & -\sum_{i \in I_1} y_i K(\mathbf{x}_i, \mathbf{x}_j) - \lambda_n\, y_j \\ \hline G(\alpha_n - \alpha_o) & = & (\lambda_o - \lambda_n)\mathbf{y} \qquad \text{avec} \qquad G_{ij} = y_i K(\mathbf{x}_i, \mathbf{x}_j) \end{array}$$

$$\alpha_n = \alpha_o + (\lambda_o - \lambda_n)\mathbf{w}$$

$$\mathbf{w} = (G)^{-1}\mathbf{y}$$

Example of regularization path



α estimation and data selection

How to choose ℓ and P to get linear *regularization path*?

the *path* is piecewise linear $\Leftrightarrow$ one is piecewise quadratic and the other is piecewise linear

the convex case [Rosset & Zhu, 07]

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^d} \ \ell(\boldsymbol{\beta}) + \lambda P(\boldsymbol{\beta})$$

1. piecewise linearity: $\displaystyle\lim_{\varepsilon \to 0} \frac{\beta(\lambda+\varepsilon) - \beta(\lambda)}{\varepsilon} = \text{constant}$
2. optimality
$$\nabla \ell(\boldsymbol{\beta}(\lambda)) + \lambda \nabla P(\boldsymbol{\beta}(\lambda)) = 0$$
$$\nabla \ell(\boldsymbol{\beta}(\lambda+\varepsilon)) + (\lambda+\varepsilon) \nabla P(\boldsymbol{\beta}(\lambda+\varepsilon)) = 0$$
3. Taylor expension
$$\lim_{\varepsilon \to 0} \frac{\beta(\lambda+\varepsilon) - \beta(\lambda)}{\varepsilon} = \left[\nabla^2 \ell(\boldsymbol{\beta}(\lambda)) + \lambda \nabla^2 P(\boldsymbol{\beta}(\lambda))\right]^{-1} \nabla P(\boldsymbol{\beta}(\lambda))$$

$\nabla^2 \ell(\boldsymbol{\beta}(\lambda)) =$ **constant** **and** $\nabla^2 P(\boldsymbol{\beta}(\lambda)) = 0$

Problems with Piecewise linear regularization path

L	P	*regression*	*classification*	*clustering*
L_2	L_1	Lasso/LARS	L1 L2 SVM	PCA L1
L_1	L_2	SVR	SVM	OC SVM
L_1	L_1	L1 LAD	L1 SVM	
		Danzig Selector		

Table: example of piecewise linear regularization path algorithms.

$P: \quad L_p = \sum_{j=1}^{d} |\beta_j|^p$

$L: \quad L_p : |f(\mathbf{x}) - y|^p \quad \text{hinge } (yf(\mathbf{x}) - 1)_+^p$

ε-insensitive
$$\begin{cases} 0 & \text{if } |f(\mathbf{x}) - y| < \varepsilon \\ |f(\mathbf{x}) - y| - \varepsilon & \text{else} \end{cases}$$

Huber's loss:
$$\begin{cases} |f(\mathbf{x}) - y|^2 & \text{if } |f(\mathbf{x}) - y| < t \\ 2t|f(\mathbf{x}) - y| - t^2 & \text{else} \end{cases}$$

standart formulation

- portfolio optimization (Markovitz, 1952)
  - return *vs.* risk
    $$\begin{cases} \min\limits_{\beta} & \frac{1}{2}\beta^\top Q\beta \\ \text{with} & \mathbf{e}^\top\beta = C \end{cases}$$
  - *efficiency frontier*: piecewise linear (*Critical path Algo.*)
- Sensitivity analysis: standart formulation (Heller, 1954)
  $$\begin{cases} \min\limits_{\beta} & \frac{1}{2}\beta^\top Q\beta + (\mathbf{c} + \lambda\,\Delta\mathbf{c})^\top\beta \\ \text{with} & A\beta = \mathbf{b} + \mu\,\Delta\mathbf{b} \end{cases}$$
- Parametric programming (see T. Gal's book 1968)
  - in the general case of PLP: the reg. path is piecewise linear
  - ... and PQP is piecewise quadratic
  - multiparametric programming

ν-SVM and other formulations...

$\nu \in [0,1]$

$$
(\nu)\begin{cases}
\min\limits_{f,\alpha_0,\xi,m} & \frac{1}{2}\|f\|^2_{\mathcal{H}} + \frac{1}{np}\sum\limits_{i=1}^{n}\xi_i^p - \nu m \\
\text{with} & y_i\big(f(\mathbf{x}_i)+\alpha_0\big) \geq m - \xi_i, \quad i=1,n, \\
\text{and} & m \geq 0, \quad \xi_i \geq 0, \quad i=1,n,
\end{cases}
$$

for $p=1$ the dual formulation is:

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

$$C = \frac{1}{m}$$

SVM with non symetric costs

problem in the primal

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

for $p=1$ the dual formulation is the following:

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

Generalized SVM

$$\min_{f \in \mathcal{H}, \alpha_0 \in \mathbb{R}} \sum_{i=1}^{n} \max\big(0, 1 - y_i(f(\mathbf{x}_i) + \alpha_0)\big) + \frac{1}{C}\varphi(f) \qquad \varphi \text{ convex}$$

in particular $\varphi(f) = \|f\|_p^p$ with $p = 1$ leads to L1 SVM.

$$\begin{cases} \min\limits_{\alpha \in \mathbb{R}^n, \alpha_0, \xi} & \mathbb{1}^\top \boldsymbol{\beta} + C\mathbb{1}^\top \xi \\ \text{with} & y_i\Big(\sum_{j=1}^{n} \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) + \alpha_0\Big) \geq 1 - \xi_i, \\ \text{and} & -\beta_i \leq \alpha_i \leq \beta_i, \quad \xi_i \geq 0, \quad i = 1, n \end{cases}$$

with $\boldsymbol{\beta} = |\boldsymbol{\alpha}|$. the dual is:

$$\begin{cases} \max\limits_{\gamma, \delta, \delta^* \in \mathbb{R}^{3n}} & \mathbb{1}^\top \gamma \\ \text{with} & \mathbf{y}^\top \gamma = 0, \quad \delta_i + \delta_i^* = 1 \\ & \sum_{j=1}^{n} \gamma_j k(\mathbf{x}_i, \mathbf{x}_j) = \delta_i - \delta_i^*, \quad i = 1, n \\ \text{and} & 0 \leq \delta_i, 0 \leq \delta_i^*, \ 0 \leq \gamma_i \leq C, \quad i = 1, n \end{cases}$$

Mangassarian, 2001

SVM reduction (reduced set method))

- objective: compile the model
- $f(x) = \sum_{i=1}^{n_s} \alpha_i k(\mathbf{x}_i, \mathbf{x}), \; n_s \ll n,$ $\quad n_s$ too big
- compiled model as the solution of:

$$g(\mathbf{x}) = \sum_{i=1}^{n_c} \beta_i k(\mathbf{z}_i, \mathbf{x}), \; n_c \ll n_s$$

- $\beta, \mathbf{z}_i$ and c are tuned by minimizing:

$$\min_{\beta, \mathbf{z}_i} \|g - f\|_H^2$$

where

$$\min_{\beta, \mathbf{z}_i} \|g - f\|_H^2 = \boldsymbol{\alpha}^\top K_x \boldsymbol{\alpha} + \boldsymbol{\beta}^\top K_z \boldsymbol{\beta} - 2\boldsymbol{\alpha}^\top K_{xz} \boldsymbol{\beta}$$

some authors advice $0,03 \leq \frac{n_c}{n_s} \leq 0,1$

- solve it by using use (stochastic) gradient (its a RBF problem)

Burges 1996, Ozuna 1997, Romdhani 2001

SVM and probabilities (1/2)

$$\log \frac{\mathbb{P}(Y=1|\mathbf{x})}{\mathbb{P}(Y=-1|\mathbf{x})} \text{ as (almost) the same sign as } f(\mathbf{x})$$

$$\log \frac{\mathbb{P}(Y=1|\mathbf{x})}{\mathbb{P}(Y=-1|\mathbf{x})} = a_1 f(\mathbf{x}) + a_2 \quad \mathbb{P}(Y=1|\mathbf{x}) = 1 - \frac{1}{1+\exp^{a_1 f(\mathbf{x})+a_2}}$$

a_1 et a_2 estimated using maximum likelihood

some facts

- SVM is universaly consistent (coverges towards the Bayes risk)
- SVM asymptotically implements the bayes rule
- but theoreticaly: no consistency towards conditional probabilities (due to the nature of sparsity)
- to estimate conditional probabilities on an interval (typicaly$[\frac{1}{2}-\eta, \frac{1}{2}+\eta]$) to spasness in this interval (all data points have to be support vectors)

Platt, 99 ; Bartlett & Tewari, JMLR, 07

SVM and probabilities (2/2)

An alternative approach

$$g(\mathbf{x}) - \varepsilon^-(\mathbf{x}) \ \leq \ \mathbb{P}(Y=1|\mathbf{x}) \ \leq \ g(\mathbf{x}) + \varepsilon^+(\mathbf{x})$$

with $g(\mathbf{x}) = \frac{1}{1+4^{-f(\mathbf{x})-\alpha_0}}$
non parametric functions ε^- and ε^+ have to verify:

$$\begin{aligned} g(\mathbf{x}) + \varepsilon^+(\mathbf{x}) &= \exp^{-a_1(1-f(\mathbf{x})-\alpha_0)_+ + a_2} \\ 1 - g(\mathbf{x}) - \varepsilon^-(\mathbf{x}) &= \exp^{-a_1(1+f(\mathbf{x})+\alpha_0)_+ + a_2} \end{aligned}$$

with $a_1 = \log 2$ and $a_2 = 0$

Grandvalet *et al.*, 07

logistic regression and the import vector machine

- Logistic regression is NON sparse
- kernalize it using the *dictionary* strategy
- Algorithm:
  - find the solution of the KLR using only a subset $\mathcal{S}$ of the data
  - build $\mathcal{S}$ iteratively using active constraint approach
- this trick brings sparsity
- it estimates probability
- it can naturally be generalized to the multiclass case
- efficent when uses:
  - a few import vectors
  - component-wise update procedure
- extention using L_1 KLR

Zhu & Hastie, 01 ; Keerthi *et. al.*, 02

Multiclass SVM

- one *vs* all: winner takes all
- one *vs* one:
  - max-wins voting
  - pairwise coupling: use probability
- global approach (size $c \times n$),
  - formal (differents variations)

$$\begin{cases} \displaystyle\min_{f \in \mathcal{H}, \alpha_0, \xi \in \mathbb{R}^n} & \displaystyle\frac{1}{2}\sum_{\ell=1}^{c} \|f_\ell\|_{\mathcal{H}}^2 + \frac{C}{p}\sum_{i=1}^{n}\sum_{\ell=1, \ell \neq y_i}^{c} \xi_{i\ell}^p \\ \text{with} & y_i\big(f_{y_i}(\mathbf{x}_i) + b_{y_i} - f_\ell(\mathbf{x}_i) - b_\ell\big) \geq 1 - \xi_{i\ell} \\ \text{and} & \xi_{i\ell} \geq 0 \text{ for } i = 1, \ldots, n; \quad \ell = 1, \ldots, c; \quad \ell \neq y_i \end{cases}$$

  non consistent estimator but practicaly usefull
  - structured outputs

approach	problem size	number of sub problems
all together	$n \times c$	1
1 vs. all	n	c
1 vs. 1	$\frac{2n}{c}$	$\frac{c(c-1)}{2}$

Duan & Keerti, 05 ; Bartlett 06

Multiclass SVM

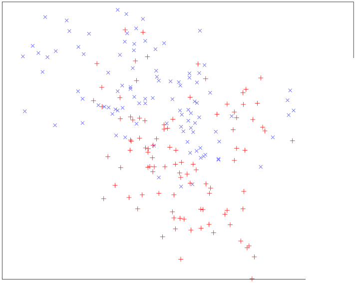
- one *vs* all: winner takes all
- one *vs* one:
  - max-wins voting
  - pairwise coupling: use probability – best results
- global approach (size $c \times n$),
  - formal (differents variations)

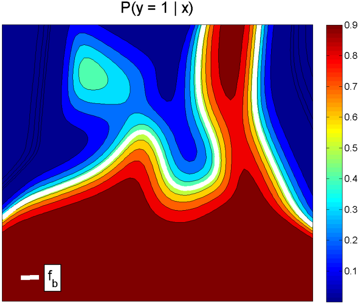
$$\begin{cases} \min_{f \in \mathcal{H}, \alpha_0, \xi \in \mathbb{R}^n} & \frac{1}{2} \sum_{\ell=1}^{c} \|f_\ell\|_{\mathcal{H}}^2 + \frac{C}{p} \sum_{i=1}^{n} \sum_{\ell=1, \ell \neq y_i}^{c} \xi_{i\ell}^p \\ \text{with} & y_i \big(f_{y_i}(\mathbf{x}_i) + b_{y_i} - f_\ell(\mathbf{x}_i) - b_\ell \big) \geq 1 - \xi_{i\ell} \\ \text{and} & \xi_{i\ell} \geq 0 \text{ for } i = 1, ..., n; \quad \ell = 1, ..., c; \quad \ell \neq y_i \end{cases}$$

  non consistent estimator but practicaly usefull
  - structured outputs

approach	problem size	number of sub problems
all together	$n \times c$	1
1 vs. all	n	c
1 vs. 1	$\frac{2n}{c}$	$\frac{c(c-1)}{2}$

Mixture data

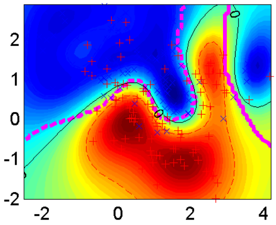




- x : 200×2
- y : 100 for each class
- mixturee model with 10 gaussians
- the bayes error is known

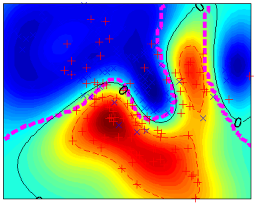
the kernel effect

C = 7.3; σ = 1.75



Noyau rationnel

C = 1.7; σ = 0.8



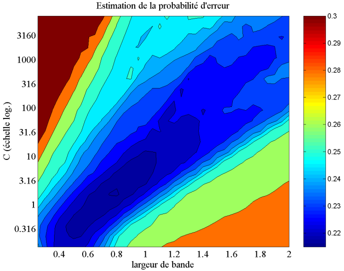
Noyau gaussien

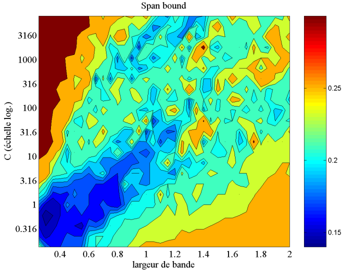
tuning C and σ : *grid search*

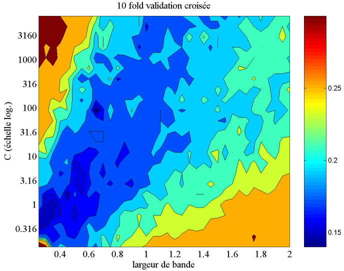
for $\sigma = 0.5 : 0.25 : 2$

for $C = 0.1$ à 10000

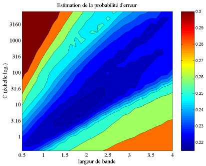
3 different error estimate

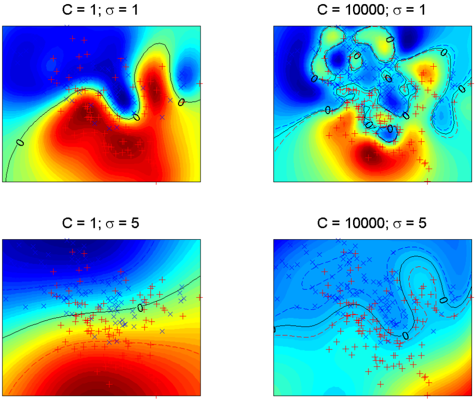






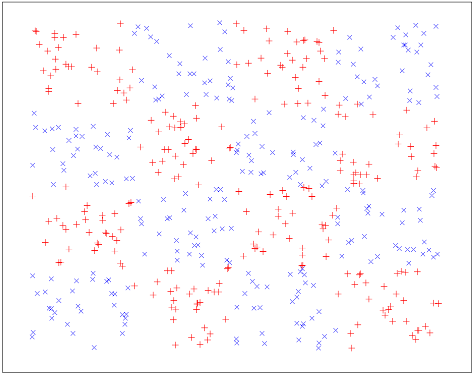
C and σ influence



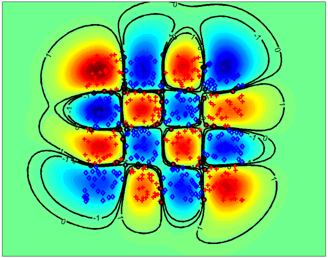


checker board

- 2 classes
- 500 examples
- separable

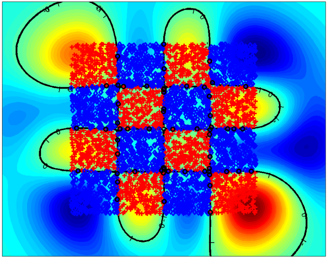


a separable case

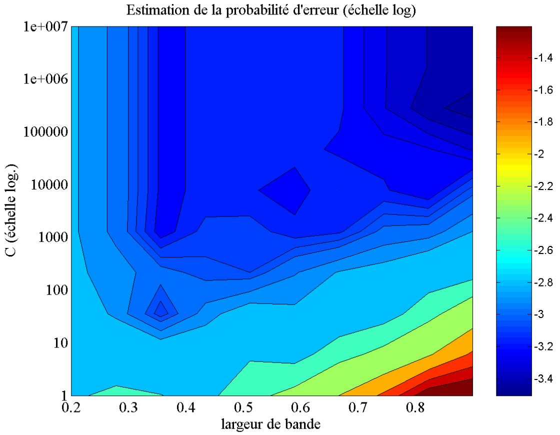


$n = 500$ data points

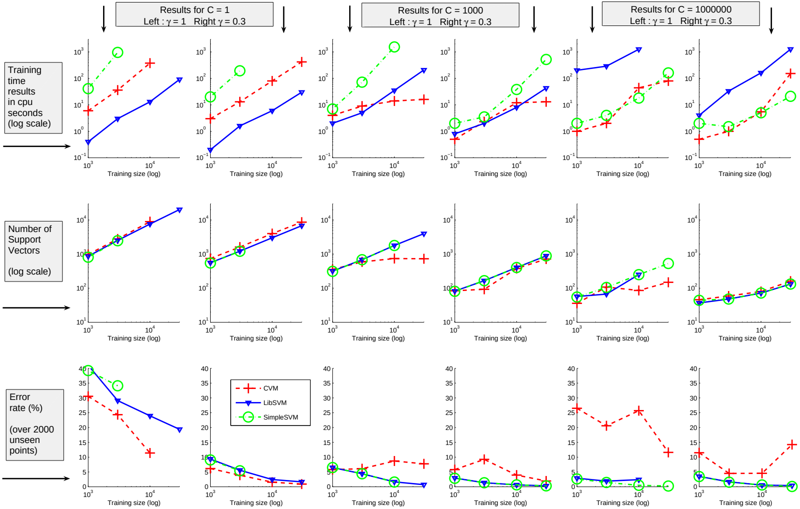
$n = 5000$ data points



tuning C and σ : *grid search*



empirical complexity



G. Loosli *et al* JMLR, 2007

Conclusion

- nonlinearity throught kernel: using examples influence
- universality: kernel to functions (R.K.H.S.)
- representer theorem: from functions to vectors
- $L1$ provides sparsity

a question of vocabulary

- margin: regularization
- Mercer kenrel: positive kernel
- SVM: a method among others

- kernels (RKHS) — universality
- regularization — univ. consistency
- convexity — efficiency
- sparsity — efficiency

no (explicit) model

but a kernel, a cost and a regularity

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

what's new since 1995

- Applications
  - kernlisation $w^\top \mathbf{x} \rightarrow \langle f, k(\mathbf{x}, .)\rangle_{\mathcal{H}}$
  - 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
- Automatic and adaptive model selection
  - variable selection
  - kernel tuning (k et σ)
  - 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