Kernels and kernel based methods

Anestis Antoniadis (UJF)
e-mail:antonia@imag.fr

Outline

- Review of some facts on SVM (seen in Lecture 2).
- From linearity to nonlinearity.
- (semi-)Positive Definite Kernels.
- Kernel methods in Statistical Learning.
- Some examples.

Introduction

The computer scientist says: SVM is a linear classifier with large margin in a kernel space .

The statistician says: SVM is a nonparametric estimator. It is based on the minimization of a regularized empirical risk on a Hilbert space of real functions with a piecewise linear penalty function.

Binary Classification

Inputs $\mathbf{X} \in \mathcal{X}$

Class $Y \in \{-1, +1\}$

Goal: find a *classifier* $g : \mathcal{X} \to \{-1, +1\}$ presenting the smallest generalization error

$$L(g) = \mathbb{E}\left[1_{\{g(\mathbf{X} \neq Y\}}\right]$$

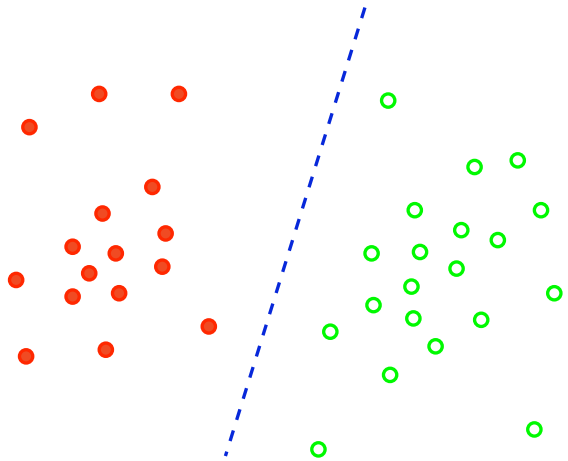
The best possible classifier is the *Bayes* one $g^\star$ and we aimed estimating $g^\star$ given an i.i.d. learning sample $(\mathbf{X}_i, Y_i)$, $i = 1, \ldots, n$.

Linear Classifier

The case $\mathcal{X} = \mathbb{R}^d$

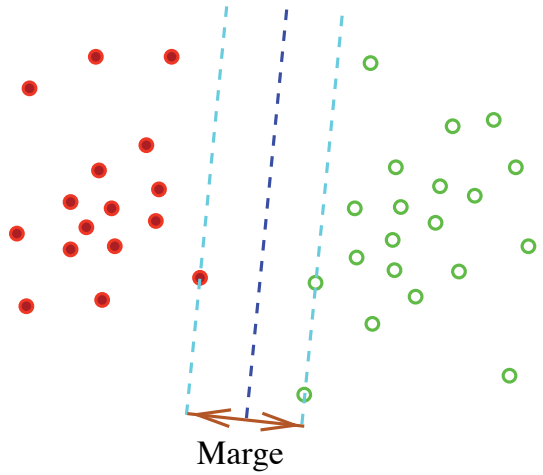
We would like to find a *linear* classifier, of the form

$$g_{w,b}(\mathbf{x}) = 1\{\langle \mathbf{w}, \mathbf{x} \rangle > b\} - 1\{\langle \mathbf{w}, \mathbf{x} \rangle \leq b\}$$



The principle of large margins

Assume that the two classes in the learning sample are perfectly separated by means of a linear classifier. We select the one for which the distance to the closest representatives in the two classes is maximum, i.e. the classes are separated with a "*maximum margin*".



Motivation: A larger margin allows a better control over the difference between the empirical error and the generalization error.

Optimization problem

Finding the best linear classifier $(\mathbf{w}, b)$ amounts in finding the maximum of

$$\min_{i=1,\ldots,n} |\langle \mathbf{w}, \mathbf{X}_i \rangle + b|,$$

under the constrains $\|\mathbf{w}\|^2 = 1$ et $\forall i$, $(\langle \mathbf{w}, \mathbf{X}_i \rangle + b) Y_i > 0$.

Trick : an equivalent way in tackling the problem is to search for the minimum of $\|\mathbf{w}\|^2$ under the constrains $\forall i$, $(\langle \mathbf{w}, \mathbf{X}_i \rangle + b) Y_i > 0$.

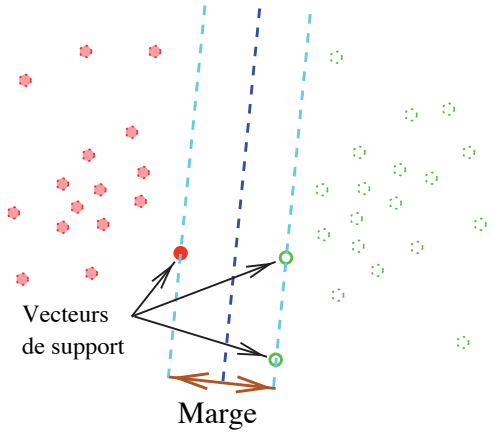
This is a *quadratic* optimization problem under linear constrains and there exists a large choice of good algorithms for solving it.

The maximal margin hyper plane

We have seen (duality arguments) that the optimal vector $\mathbf{w}$ can be written as

$$\mathbf{w} = \sum_{i \in SV} a_i \mathbf{X}_i$$

where the coefficients a_i are nonzero only for input points $\mathbf{X}_i$ that are located exactly “on the margin” (the *support vectors*).



Consequence

One may rewrite the optimization problem in terms of the a_i's:

$$\begin{aligned} \|\mathbf{w}\|^2 &= \sum_{i,j} a_i a_j \langle \mathbf{X}_i, \mathbf{X}_j \rangle, \\ \langle \mathbf{w}, \mathbf{X}_{i_0} \rangle &= \sum_{i} a_i \langle \mathbf{X}_i, \mathbf{X}_{i_0} \rangle. \end{aligned}$$

Note that from the $(\mathbf{X}_i)_1^n$ the pertinent information for solving the problem is the *Gram* matrix

$$G = [\langle \mathbf{X}_i, \mathbf{X}_j \rangle]_{i,j}.$$

The previous formulation has two disadvantages : it is not applicable for data that are not linearly separable and it is very sensitive to outliers.

Corresponding optimization

A softer version consists in minimizing

$$\|\mathbf{w}\|^2 + C \sum_{i=1}^{n} \xi_i,$$

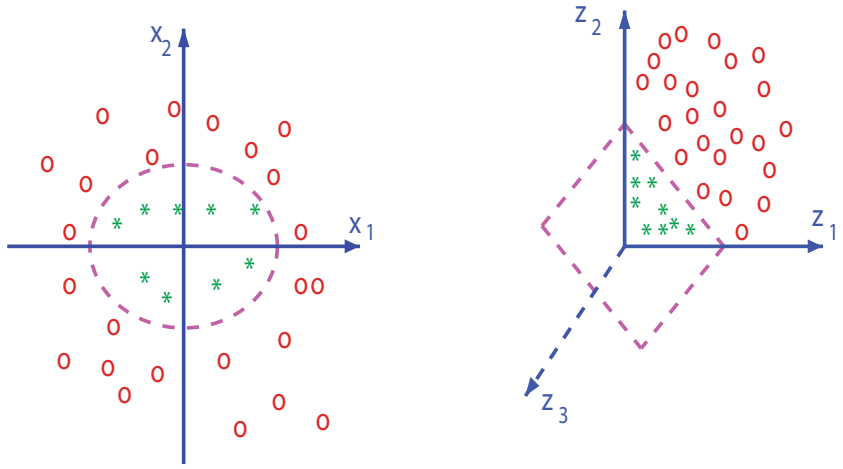
under the constrains $\forall i$, $(\langle \mathbf{w}, \mathbf{X}_i \rangle + b) Y_i > 1 - \xi_i$.

Note: constant C is a parameter of the algorithm. One has to choose it from the data.

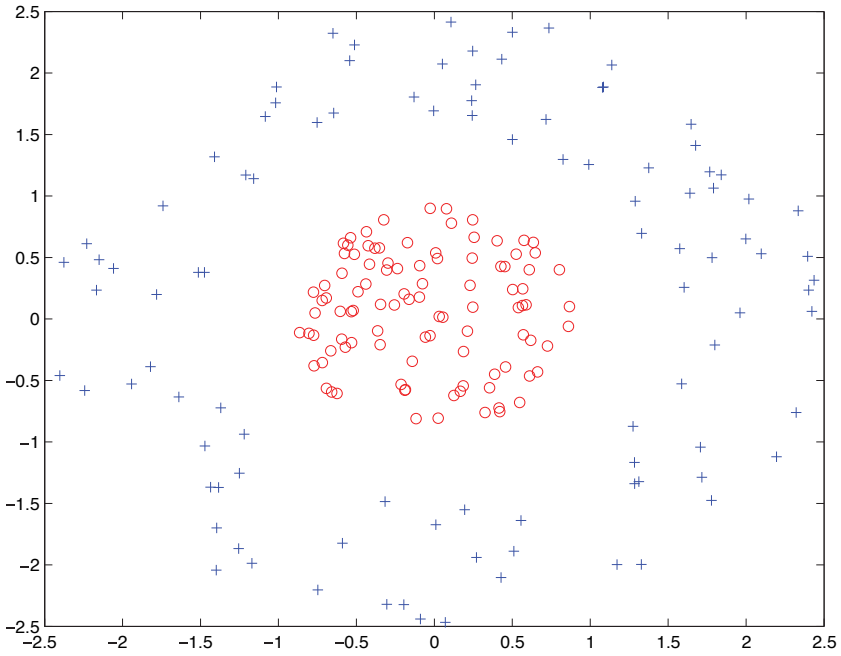
The solution $\mathbf{w}$ as a combination of support vectors is still valid in this case.

From linearity to nonlinearity

A second ingredient of the SVM's is totally independent and maybe more important than the first one. Moreover, it is not only limited to classification. It is based on the trivial remark that one may transform a nonlinear method to a linear one by sending the original data into a space of a larger dimension.



An example



The first class is made with i.i.d observations from a Uniform distribution on a disk of radius 0.9 and the second one from i.i.d. observations on the circular band.

- Consider $\Phi : \mathbb{R}^2 \to \mathbb{R}^3$ defined by

$$\mathbf{x} = (x_1, x_2) \to \mathbf{z} = \Phi(x_1, x_2) = (x_1^2, \sqrt{2}x_1x_2, x_2^2).$$

- Using a linear SVM classifier to separate the two classes from data transformed by Φ in $\mathbb{R}^3$ gives separating hyper planes of the form

$$\langle \mathbf{w}, \mathbf{z} \rangle_{\mathbb{R}^3} + b = 0$$

- As functions of $\mathbf{x}$ these are ellipses. One therefore may use linear SVM's on a transformed version of the data to get a nonlinear classifier with no much effort.

Data representation

We have an algorithm A (classification (or regression)) that is able to handle data in a space $\mathcal{X}$. To deal with a data sample S, we have used $\Phi : \mathcal{F} \to \mathcal{X}$ and worked with A on the set:

$$\Phi(\mathcal{S}) = \{\Phi(\mathbf{x}_1), \ldots, \Phi(\mathbf{x}_N)\} \in \mathcal{X}.$$

Finding such transforms Φ is not easy in general. A better way is to look at the problem using a *similarity* matrix.

Similarites

Let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ a *similarity* function. Using K, one may represent the data sample $\mathcal{S} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ by its similarity $N \times N$ matrix :

$$K_{i,j} = K(\mathbf{x}_i, \mathbf{x}_j)$$

Such a representation of the data is universal whatever the nature of the inputs is and the size is always $N \times N$.

We do not need anymore to define ad hoc algorithms for specific data: algorithms made for square matrices are enough. We only need to define appropriate similarity matrices.

The kernel trick

An important observation is that for many linear classification algorithms, including SVM's, knowing the inner products between the data points is sufficient for finding and computing the target function.

If one wishes to apply a method on the range of $\Phi(\mathbf{x})$ as before, he doesn't need to evaluate the $\Phi(\mathbf{x})$ explicitly. It is enough to compute the $K(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle$.

Conversely: under what conditions $K(\mathbf{x}, \mathbf{x}')$ may be written as above?

Positive definite kernels

Definition . A positive definite kernel (p.d.k.) on a set $\mathcal{X}$ is a function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that

- K is symmetric

$$K(\mathbf{x}, \mathbf{x}') = K(\mathbf{x}', \mathbf{x}), \text{ for all } (\mathbf{x}, \mathbf{x}') \in \mathcal{X} \times \mathcal{X}.$$

- K is positive definite i.e. for all integer N, $(\mathbf{x}_1, \ldots, \mathbf{x}_N) \in \mathcal{X}^N$, $(a_1, \ldots, a_N) \in \mathbb{R}^N$,

$$\sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j K(\mathbf{x}_i, \mathbf{x}_j) \geq 0.$$

Namely for any sample $\mathcal{S}$ the similarity matrix K is positive definite.

Examples of p.d. kernels

Let $\mathcal{X} = H$ a Hilbert space with scalar product $\langle \cdot, \cdot \rangle$. Let the following function from H^2 into $\mathbb{R}$ defined by

$$K(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle, \quad \forall (\mathbf{x}, \mathbf{x}') \in H \times H.$$

Then K is a p.d.k on H.

More generally let Φ be a function from $\mathcal{X}$ with values in a Hilbert space H. Then K defined by

$$K(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle, \quad \forall (\mathbf{x}, \mathbf{x}') \in \mathcal{X} \times \mathcal{X},$$

is a p.d.k.

Elementary properties

- for all $\mathbf{x} \in \mathcal{X}$, $K(\mathbf{x}, \mathbf{x}) \geq 0$.
- for all $(\mathbf{x}, \mathbf{x}') \in \mathcal{X} \times \mathcal{X}$,

$$|K(\mathbf{x}, \mathbf{x}')| \leq \sqrt{K(\mathbf{x}, \mathbf{x})}\sqrt{K(\mathbf{x}', \mathbf{x}')}.$$

- for all $(\mathbf{x}, \mathbf{x}') \in \mathcal{X} \times \mathcal{X}$,

$$|K(\mathbf{x}, \mathbf{x}')| \leq K(\mathbf{x}, \mathbf{x}) + K(\mathbf{x}', \mathbf{x}')$$

- If K_1 and K_2 are two p.d.k., then for any $a_1 \geq 0$ and any $a_2 \geq 0$, $a_1 K_1 + a_2 K_2$ is a p.d.k.

- If K_1 and K_2 are two p.d.k., then K defined by

$$K(\mathbf{x}, \mathbf{x}') = K_1(\mathbf{x}, \mathbf{x}') K_2(\mathbf{x}, \mathbf{x}')$$

is a p.d.k.

- If L is a p.d.k., then K defined by

$$K(\mathbf{x}, \mathbf{x}') = \exp(K(\mathbf{x}, \mathbf{x}'))$$

is a p.d.k.

Examples

- *Radial basis function (RBF)*

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

- *Polynomial kernel*

$$K(\mathbf{x}, \mathbf{x}') = \left(\langle \mathbf{x}, \mathbf{x}' \rangle + \theta\right)^d, \quad d \in \mathbb{N}, \theta \in \mathbb{R}.$$

- *Sigmoidal kernel*

$$K(\mathbf{x}, \mathbf{x}') = \tanh\left(\kappa \langle \mathbf{x}, \mathbf{x}' \rangle + \theta\right), \quad \kappa \in \mathbb{R}^+, \theta \in \mathbb{R}.$$

Conversely . . .

Theorem. If K is p.d.k on a arbitrary set $\mathcal{X}$, then *there exists a Hilbert space* H with scalar product $\langle \cdot, \cdot \rangle_H$ and a map $\Phi : \mathcal{X} \to H$ such that:

$$K(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle_H, \quad \forall (\mathbf{x}, \mathbf{x}') \in \mathcal{X} \times \mathcal{X}.$$

Many proofs of this Theorem exist. The proof is easy when $\mathcal{X}$ is finite; Mercer (1909) has proved the Theorem for $\mathcal{X} = [a, b]$ and K continuous, Kolmogorov (1941) for $\mathcal{X}$ countable, and Aronsjan (1944, 1950) in the general case.

Proofs

• The case $\mathcal{X}$ finite. Assume that $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and let K be a p.d.k on$\mathcal{X}$. The proof follows from the fact that the corresponding similarity matrix K is positive definite and relies upon a SVD of K.

• Consider now the case where $\mathcal{X}$ is a compact metric space (typically a bounded closed set of $\mathbb{R}^d$) and let K be a continuous p.d.k on $\mathcal{X} \times \mathcal{X}$. Such a kernel is called a Mercer kernel. The proof follows from several lemmas that we are going to examine.

Let μ the Borel measure on $\mathcal{X}$ and $\mathcal{H} = L_2(\mathcal{X}, d\mu)$.

For any function $K : \mathcal{X}^{\times 2} \to \mathbb{R}$ set (when it is defined):

$$(L_K f)(\mathbf{x}) = \int K(\mathbf{x}, \mathbf{t}) f(\mathbf{t}) d\mu(\mathbf{t}).$$

We then have:

Lemma 1 If K is a Mercer kernel, then L_K is a compact bounded linear operator on $L_2(\mathcal{X}, d\mu)$,self-adjoint and positive

Proof of Lemma 1

- L_K is obviously linear from $L_2(\mathcal{X}, d\mu)$ into $L_2(\mathcal{X}, d\mu)$
- L_K is bounded
- L_K is compact (Ascoli)
- L_K is self-adjoint
- L_K is positive

Spectral Theorem

Lemma 2 *Let L a compact linear operator on a Hilbert space $\mathcal{H}$. There exists in $\mathcal{H}$ a complete orthonormal system $\{\psi_1, \psi_2, \dots\}$ of eigenfunctions of L. The corresponding eigenvalues $\{\lambda_1, \lambda_2, \dots\}$ are real if L is self-adjoint, and positive if L is positive.*

For L_K, the eigenfunctions ψ_k corresponding to the eigenvalues $\lambda_k \neq 0$ are continuous functions, since:

$$\psi_k = \frac{1}{\lambda_k} L_K \psi_k.$$

Mercer's Theorem

Lemma 3 *Let $\mathcal{X}$ a compact normed space, ν a Borel measure on $\mathcal{X}$ and K a Mercer kernel. Let $\{\lambda_1, \lambda_2, \dots\}$ be the eigenvalues of L_K (in decreasing order) and $\{\psi_1, \psi_2, \dots\}$ the o.n.s of corresponding eigenfunctions. Then, for any $\mathbf{x}$, $\mathbf{x}'$ in $\mathcal{X}$:*

$$K(\mathbf{x}, \mathbf{x}') = \sum_k \lambda_k \psi_k(\mathbf{x}) \psi_k(\mathbf{x}'),$$

(the convergence is absolute and uniform on $\mathcal{X}^{\times 2}$)
From the above, it follows that $\Phi : \mathcal{X} \to \ell^2$ given by $\Phi(\mathbf{x}) = \left\{\sqrt{\lambda_k}\psi_k(\mathbf{x})\right\}$ is well defined, continuous and such that

$$K(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle_{\ell^2}.$$

A new construction

Let $\mathcal{X}$ be an arbitrary set, and $(H, \langle \cdot, \cdot \rangle_H)$ a Hilbert space of functions on $\mathcal{X}$ ($H \subset \mathbb{R}^{\mathcal{X}}$).

A function $K : \mathcal{X}^{\times 2} \to \mathbb{R}$ is a reproducing kernel (r.k.) iff:

- H contains all functions of the form

$$\forall \mathbf{x} \in \mathcal{X}, \quad K_{\mathbf{x}} : \mathbf{x}' \to K(\mathbf{x}, \mathbf{x}')$$

- $\forall \mathbf{x} \in \mathcal{X}$ et $f \in H$, we have:

$$f(\mathbf{x}) = \langle f, K_{\mathbf{x}} \rangle_H$$

If a r.k.. exists, H is called a reproducing kernel Hilbert space (rkhs).

Properties of r.k.'s and rkhs's

- if a r.k. exists, it is unique
- a r.k. exists iff the evaluation functional is continuous
- a r.k is a p.d.k .
- if K is a p.d.k there exists a rkhs having K for r.k.
- if K is a r.k., it has the reproducing property.

Mercer kernel and rkhs

Assume that $\lambda > 0$ for all $k \geq 1$. Let the Hilbert space:

$$H_K = \left\{ f \in L_2(\mathcal{X}, d\mu); f = \sum_{i=1}^{n} a_i \psi_i, \sum \frac{a_k^2}{\lambda_k} < \infty \right\}$$

with the scalar product $\langle f, g \rangle_K = \sum_{k=1}^{\infty} \frac{a_k b_k}{\lambda_k}$.

To show that H_K is the rkhs associated to K, we must show that:

- it is a space of functions from $\mathcal{X}$ into $\mathbb{R}$,
- for any $\mathbf{x} \in \mathcal{X}$, $K_{\mathbf{x}} \in H_K$,
- for any $\mathbf{x} \in \mathcal{X}$ et $f \in H_K$, $f(\mathbf{x}) = \langle f, K_{\mathbf{x}} \rangle_K$.

Representation Theorem

Theorem Le $\mathcal{X}$ be a set with a p.d.k. K, H_K the corresponding rkhs, and $\mathcal{S} \subset \mathcal{X}$ a finite subset. Let $\Psi : \mathbb{R}^{n+1} \to R$ a function with $n+1$ arguments, strictly increasing with respect to its las argument. Then any solution to the problem:

$$\min_{f \in H_K} \Psi(f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n), \|f\|_{H_K}) = \min_{f \in H_K} \xi(f, \mathcal{S})$$

can be represented as:

$$\forall \mathbf{x} \in \mathcal{X}, f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}_i, \mathbf{x}).$$

Often Ψ has the following form:

$$\Psi(f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n), \|f\|_{H_K})) = c(f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n) + \nu \|f\|_{H_K})$$

kernel SVM for classification

- $\mathcal{S} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}, \mathbf{x}_i \in \mathcal{X}$
- $\{y_1, y_2, \ldots, y_n\}, y_i \in \{-1, +1\}$ corresponding labels
- Classification : find $f : \mathcal{X} \to \{-1, +1\}$ to predict Y by $f(\mathbf{x})$.
- K kernel on $\mathcal{X} \times \mathcal{X}$, H Hilbert space and Φ such that $K(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle_H$.
- A linear classifier on H : perfect classification,

$$(\langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle_H + b) y_i \geq 1, \quad i = 1, \ldots, n.$$

kernel SVM

The maximum margin linear classifier in the input space H is the solution of the following quadratic problem

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2}\|\mathbf{w}\|_H^2 \\ \text{under the constrains} \quad & (\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle_H + b) y_i \geq 1, \quad i = 1, \ldots, n. \end{aligned}$$

Maximum Margin SVM

Using the kernel trick and writing the dual problem we do not need the explicit expression of $\mathbf{w}$.

Using the Lagrange multipliers λ_i, $i = 1, \ldots, n$ associated to each of the constrains respecter, the Lagrange formulation is:

$$L(\mathbf{w}, b, \boldsymbol{\lambda}) = \frac{1}{2}\|\mathbf{w}\|_H^2 - \sum_{i=1}^{n} \lambda_i[(\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle_H + b)y_i - 1)]$$

Dual program (1)

The dual problem is obtained by minimizing, for any fixed value of the vector of multipliers $\boldsymbol{\lambda}$, the Lagrangian

$$(\mathbf{w}, b) \rightarrow L(\mathbf{w}, b, \boldsymbol{\lambda}).$$

The vector of multipliers $\boldsymbol{\lambda}$ is admissible (for the dual) if $\lambda_i \geq 0$ for $i = 1, \ldots, n$ et $\sum_{i=1}^{n} \lambda_i y_i = 0$. For an admissible vector $\boldsymbol{\lambda}$, the dual is given by:

$$\sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i,j=1}^{n} \lambda_i \lambda_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

computable using only the kernel.

Dual program (2)

The dual program is

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i,j=1}^{n} \lambda_i \lambda_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{under the constrains} \quad & \lambda_i \geq 0, \quad i = 1, \ldots, n. \\ & \sum_{i=1}^{n} \lambda_i y_i = 0. \end{aligned}$$

which only depends on the similarity!! If the initial problem admits a solution then, the KKT conditions show that the solution $\mathbf{w}^\star$ is given by

$$\mathbf{w}^* = \sum_{i=1}^{n} \lambda_i^* y_i \Phi(\mathbf{x}_i),$$

where the λ_i^*'s are the optimal Lagrange multipliers.

The optimal Lagrange multipliers are nonzero if the corresponding inputs $\mathbf{x}_i$ are on the margin, i.e.

$$(\langle \mathbf{w}^\star, \Phi(\mathbf{x}_i) \rangle_H + b^*) y_i = 1,$$

outlining the important property of SVM's : generally the solutions are sparse!!!

The optimal value of b, b^*, is obtained by averaging the support vectors indexed by $I = \{i \in \{1, \ldots, n\}, \lambda_i^* > 0\}$,

$$b^\star = \frac{1}{|I|} \sum_{i \in I} \left(y_i - \sum_{j=1}^{n} y_j \lambda_j^* K(\mathbf{x}_i, \mathbf{x}_j) \right).$$

Classification rule

The classifier is given by :

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

with

$$b^\star = \frac{1}{|I|} \sum_{i \in I} \left(y_i - \sum_{j=1}^{n} y_j \lambda_j^* K(\mathbf{x}_i, \mathbf{x}_j) \right),$$

computable by using solely the kernel K!!

The nonseparable case

To relax the separability conditions we introduce again the slack variables ξ_i and the constrains become

$$(\langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle_H + b) y_i \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, n.$$

The primal problem then is

$$\begin{array}{rl} \text{minimize} & \frac{1}{2}\|\mathbf{w}\|_H^2 + \frac{C}{n}\sum_{i=1}^n \xi_i \\ \text{under the constrains} & (\langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle_H + b) y_i \geq 1 - \xi_i, \quad i = 1, \ldots, n \\ & \xi_i \geq 0, \quad i = 1, \ldots, n, \end{array}$$

with a regularization constant $C > 0$.

The dual

The dual problem is

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i,j=1}^{n} \lambda_i \lambda_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{under the constrains} \quad & 0 \leq \lambda_i \leq \frac{C}{n}, \quad i = 1, \ldots, n, \\ & \sum_{i=1}^{n} \lambda_i y_i = 0. \end{aligned}$$

KKT conditions

The KKT conditions are

$$\begin{aligned}
\lambda_i = 0 &\Rightarrow (\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle_H + b) y_i = 1 \quad \xi_i = 0 \\
0 < \lambda_i < \frac{C}{n} &\Rightarrow (\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle_H + b) y_i = 1 \quad \xi_i = 0 \\
\lambda_i = \frac{C}{n} &\Rightarrow (\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle_H + b) y_i \leq 1 \quad \xi_i \geq 0.
\end{aligned}$$

and the optimal b^* is computed by:

$$b^\star = \frac{1}{|I|} \sum_{i \in I} \left(y_i - \sum_{j=1}^{n} y_j \lambda_j^* K(\mathbf{x}_i, \mathbf{x}_j) \right),$$

where $I = \{i \in \{1, \ldots, n\}, 0 < \lambda_i^* < \frac{C}{n}\}$.

A statistical view

When the data is embedded in a rkhs,the optimization problem

$$\begin{array}{rc} \text{minimize} & \frac{1}{2}\|\mathbf{w}\|_H^2 + \frac{C}{n}\sum_{i=1}^n \xi_i \\ \text{under the constrains} & (\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle_H + b)y_i \geq 1 - \xi_i, \quad i = 1, \ldots, n \\ & \xi_i \geq 0, \quad i = 1, \ldots, n, \end{array}$$

is equivalent to

$$\min_{f \in H_K, b \in \mathbb{R}} \left(\frac{1}{n}\sum_{i=1}^n (1 - (f(\mathbf{x}_i) + b)y_i)_+ + \lambda \|f\|_H^2 \right),$$

and the solution appears as a penalized nonparametric estimator with a loss function of the form

$$\gamma(f, b, \mathbf{x}, y) = (1 - (f(\mathbf{x}) + b)y)_+.$$

The loss function γ

Recall that

$$\eta(\mathbf{x}) = \mathbb{P}(Y = 1 | \mathbf{X} = \mathbf{x}) = \mathbb{E}^{\mathbf{x}}(Y|\mathbf{X}).$$

For any binary classifier $g : \mathcal{X} \to \{-1, +1\}$, the classification error

$$1\{g(\mathbf{X}) \neq Y\} = (1 - g(\mathbf{X})Y)_+/2 = |Y - g(\mathbf{X})|/2$$

and the risk

$$L(g) = \mathbb{E}\left((1 - g(\mathbf{X})Y)_+\right)/2$$

Bayes rule, minimizes this risk is

$$g^*(\mathbf{x}) = \mathsf{sgn}(\eta(\mathbf{x}) - 1/2).$$

Examples of loss functions

- Squared hinge loss

$$\gamma(f, b, \mathbf{x}, y) = [(1 - (f(\mathbf{x}) + b)y)_+]^2$$

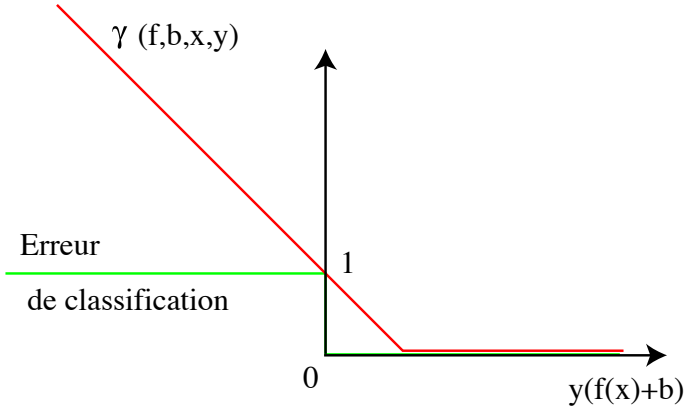
- Squared loss

$$\gamma(f, b, \mathbf{x}, y) = [(1 - (f(\mathbf{x}) + b)y]^2$$

- Exponential loss

$$\gamma(f, b, \mathbf{x}, y) = \exp[-(1 - (f(\mathbf{x}) + b)y)]$$

Hinge loss



γ is **convex** upper bound of the classification error.

Remark

If the r.k.h.s corresponding to the kernel K contains the constant functions on $\mathcal{X}$ then

$$\min_{f \in H_K, b \in \mathbb{R}} \left(\frac{1}{n} \sum_{i=1}^{n} (1 - (f(\mathbf{x}_i) + b) y_i)_+ + \lambda \| f \|_H^2 \right),$$

is equivalent to

$$\min_{f \in H_K} \left(\frac{1}{n} \sum_{i=1}^{n} (1 - f(\mathbf{x}_i) y_i)_+ + \lambda \| Pf \|_H^2 \right),$$

where Pf is the projection of f on the subspace of H_K spanned by the constant functions. If $\hat{f}$ is a solution of such a problem, then $\text{sgn}(\hat{f})$ is the SVM corresponding classification rule.

Solving the penalized problem

Since the loss is convex, we get an efficient algorithm for solving the optimization problem. Using the representation Theorem, and if H_K contains the constants, the solution $\hat{f}$ can be written as

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

The solution belongs to H_K.

Controlling the error

If the kernel K is bounded above by a constant M, then for any $\delta > 0$,

$$\forall f, \quad \|f\|_H \leq R,$$

$$L(f) \leq \mathbb{E}(\gamma(f)) \leq \frac{1}{n}\sum_{i=1}^{n} \max(\gamma(f, b, \mathbf{x}_i, y_i), 1) + 2M\,\frac{R + \sqrt{\log \delta^{-1}}}{\sqrt{n}},$$

which in terms of machine learning can be written as

$$L(f) \leq \frac{1}{n}\sum_{i=1}^{n} \max(\xi_i, 1) + 2M\frac{(1/\rho) + \sqrt{\log \delta^{-1}}}{\sqrt{n}},$$

where ρ is the margin.

What is important

- In practice one may concentrate on the construction of an appropriate kernel, the optimization being solved by a black box.
- The kernel trick allows to deal with several kinds of data.
- Research: choose the amount of regularization; model selection.

Another classification method

The SVM algorithm may be formulated as

$$\min_{f \in H_K} \left(\frac{1}{n} \sum_{i=1}^{n} (1 - f(\mathbf{x}_i) y_i)_+ + \lambda \| Pf \|_{H_K}^2 \right),$$

which is very similar with Tikhonov regularization for inverse problems.

But for such inverse problems one may show that projection based methods have more adaptive properties. This is the key remark in the algorithm KPM of Blanchard and Zwald (2004).

KPM (1)

Let K be a Mercer kernel on $\mathcal{X} \times \mathcal{X}$and let L_K be the corresponding operator

$$L_k : f(\cdot) \in L_2(\mathcal{X}, \mu) \to \int_{\mathcal{X}} K(x, \cdot) f(x) d\mu(x) \in L_2(\mathcal{X}, \mu)$$

Denote by $\psi_1, \psi_2, \ldots$ the normalized eigenfunctions of L_K, ranked in decreasing order of corresponding eigenvalues $(\lambda_i)_{i \geq 1}$. For any integer D, the subspace $\mathcal{F}_D$ spanned by $\{1, \psi_1, \ldots, \psi_D\}$ is a subspace of H_K and

$$H_K = \overline{\cup_{D=1}^{\infty} \mathcal{F}_D}.$$

KPM (2)

Instead of selecting the "best" ball in H_K as for kernel SVM, consider projection estimators $\hat{f}_D$ defined by

$$\hat{f}_D = \arg \min_{f \in \mathcal{F}_D} \sum_{i=1}^{n} (1 - f(\mathbf{x}_i) y_i)_+$$

$$\hat{f}_D(\cdot) = \sum_{j=1}^{D} \beta_j^\star \psi_j(\cdot) + b^\star,$$

with

$$(\boldsymbol{\beta}^\star, b^\star) = \arg \min_{\boldsymbol{\beta} \in \mathbb{R}^D, b \in \mathbb{R}} \sum_{i=1}^{n} \left(1 - y_i \left(\sum_{j=1}^{D} \beta_j \psi_j(x_i) + b \right) \right)_+$$

KPM (3)

Since neither μ or the eigenfunctions are known such a method cannot be applied as such.

When $\mu = P_{\mathbf{X}}$ is the marginal distribution of X, the idea is to replace $\mathcal{F}_D$ defined through L_K by the corresponding ones defined through the similarity matrix K.

It is known that the svd of the matrix K is a good approximation of the svd of the operator L_K at the points $\mathbf{x}_i$, i.e., if $V_1, \ldots, V_D$ are the first D eigenvectors of K corresponding to the eigenvalues $\hat{\lambda}_1 \geq \cdots \geq \hat{\lambda}_D$, then

$$V_i = (V_i^{(1)}, \ldots, V_i^{(n)})^T \simeq (\psi_i(\mathbf{x}_1), \ldots, \psi_i(\mathbf{x}_n))^T.$$

The algorithm (1)

The empirical version of the algorithm is then

$$(\boldsymbol{\beta}^\star, b^\star) = \arg \min_{\boldsymbol{\beta}\in\mathbb{R}^D, b\in\mathbb{R}} \sum_{i=1}^{n} \left(1 - y_i \left(\sum_{j=1}^{D} \beta_j V_j^{(i)} + b\right)\right)_+$$

and the solution has the form $\hat{f}_D(\cdot) = \sum_{j=1}^{D} \beta_j^\star \psi_j(\cdot) + b^\star$. SInce the ψ_j are unknown we use instead

$$\hat{f}_D(\cdot) = \sum_{j=1}^{D} \alpha_j^\star K(\mathbf{x}_i, \cdot) + b^\star$$

The algorithm (2)

Restricting the equations to the points in $\mathcal{S}$, we must solve

$$\beta_1^\star V_1 + \cdots + \beta_D^\star V_D = K\boldsymbol{\alpha}^\star$$

and the solution is (if the first D eigenvalues of K are > 0):

$$\alpha^* = \sum_{j=1}^{D} \frac{\beta_j^\star}{\hat{\lambda}_j} V_j$$

The algorithm (3)

• Given $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$ and an p.d.k K on $\mathcal{X} \times \mathcal{X}$, compute K, the eigenvectors $V_1, \ldots, V_n$ and the eigenvalues $\hat{\lambda}_1 \geq \cdots \hat{\lambda}_n$.

• For each dimension D such that $\hat{\lambda}_D > 0$ solve

$$(\boldsymbol{\beta}^\star, b^\star) = \arg \min_{\boldsymbol{\beta} \in \mathbb{R}^D, b \in \mathbb{R}, \boldsymbol{\xi}} \sum_{i=1}^{n} \xi_i$$

under the constrains $\forall i, \xi_i \geq 0, y_i \left(\sum_{j=1}^{D} \beta_j V_j^{(i)} + b \right) \geq 1 - \xi_i$.

• Compute $\boldsymbol{\alpha}^\star = \sum_{j=1}^{D} \frac{\beta_j^\star}{\hat{\lambda}_j} V_j$ and $\hat{f}_D(\cdot) = \sum_{j=1}^{D} \alpha_j^\star K(\mathbf{x}_i, \cdot) + b^\star$.

• Choose the dimension $\hat{D}$ giving the best performance to $\hat{f}_{\hat{D}}$ (model selection with a penalty on the dimension).

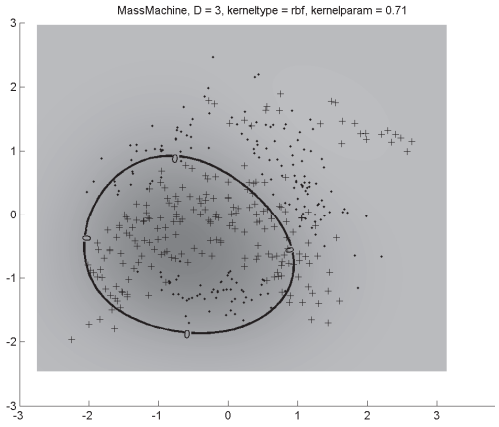
Example

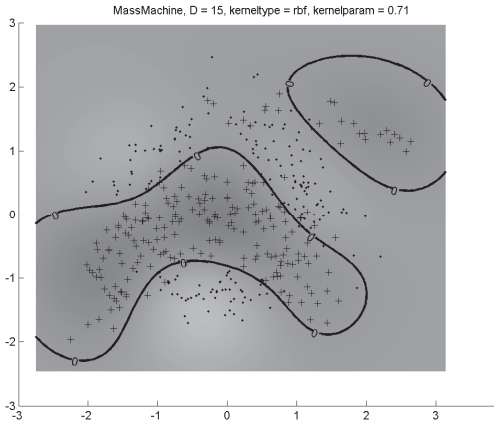
Benchmark:

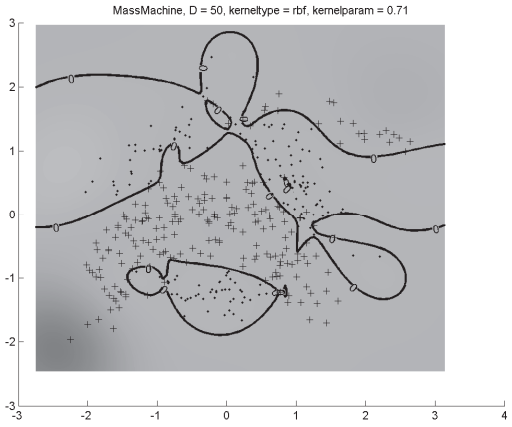
`http://ida.first.gmd.de/~raetsch/data/benchmarks.htm`

Fichier	KPM	D	SVM
Banana	10.73 (0.42)	15	11.53 (0.66)
Breast Cancer	26.51 (4.75)	24	26.04 (4.74)
Diabetis	23.37 (1.92)	11	23.53 (1.73)

D=3,15,50


MassMachine, D = 3, kerneltype = rbf, kernelparam = 0.71


MassMachine, D = 15, kerneltype = rbf, kernelparam = 0.71


MassMachine, D = 50, kerneltype = rbf, kernelparam = 0.71

Remarks

Zwald (thesis 2005) shows that a penalty proportional to the dimension is good under the assumption of a low noise ($\forall \mathbf{x}, \quad |\eta(\mathbf{x}) - 0.5| > \rho$) The optimality depends on the way the margin ρ is allowed to tend to zero. For a VC class the rate is of the order $\sqrt{VC/n}$.

Some examples

Look

$$\begin{aligned} \text{maximize} \quad & \textstyle\sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i,j=1}^{n} \lambda_i \lambda_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{under the constrains} \quad & 0 \leq \lambda_i \leq \frac{C}{n}, \quad i = 1, \ldots, n, \\ & \textstyle\sum_{i=1}^{n} \lambda_i y_i = 0. \end{aligned}$$

$$\mathbf{w}^* = \sum_{i=1}^{n} \lambda_i^* y_i K(\mathbf{x}_i, \cdot), \quad K(\mathbf{x}, \mathbf{x}') = \exp(-\|\mathbf{x} - \mathbf{x}'\|^2/(2\sigma^2))$$

Effects of σ and of C.

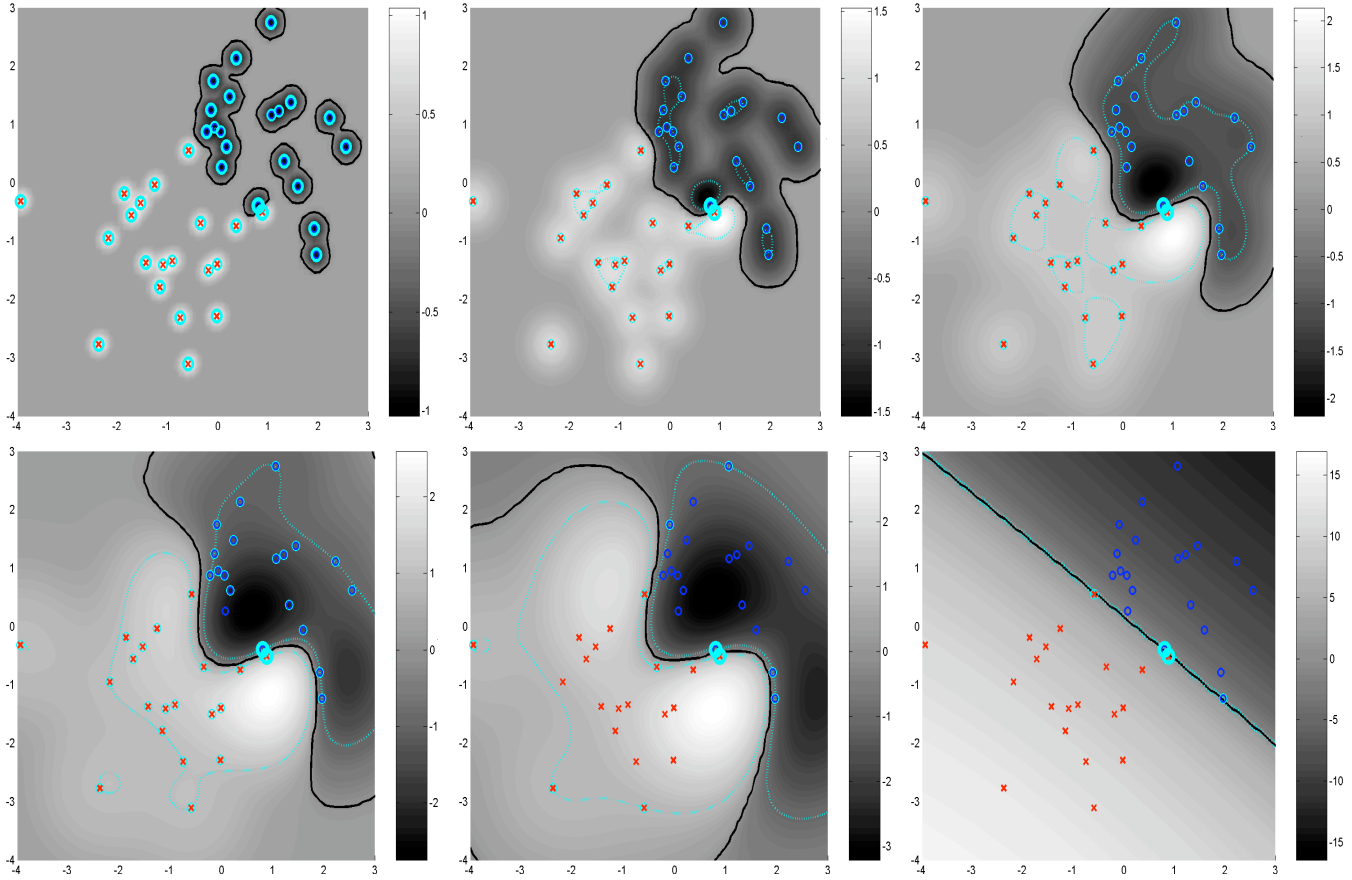
Effect of σ

- 2-D data with two classes made with 20 samples each from $N((-1,-1),I_2)$ and $N(1,1),I_2)$.
- Learning an SVM on theses data
- plot the results for several values of σ

code

```
randn('seed',1);
m=20;
d=1;
s=1;
x1=[randn(m,1)*s-d randn(m,1)*s-d]
x2=[randn(m,1)*s+d randn(m,1)*s+d]
d=data([x1;x2],[ones(m,1);-ones(m,1)]);
a=svm;
sigma=10;
a.child=kernel('rbf',sigma);
[r a]=train(a,d);
plot(a)
```

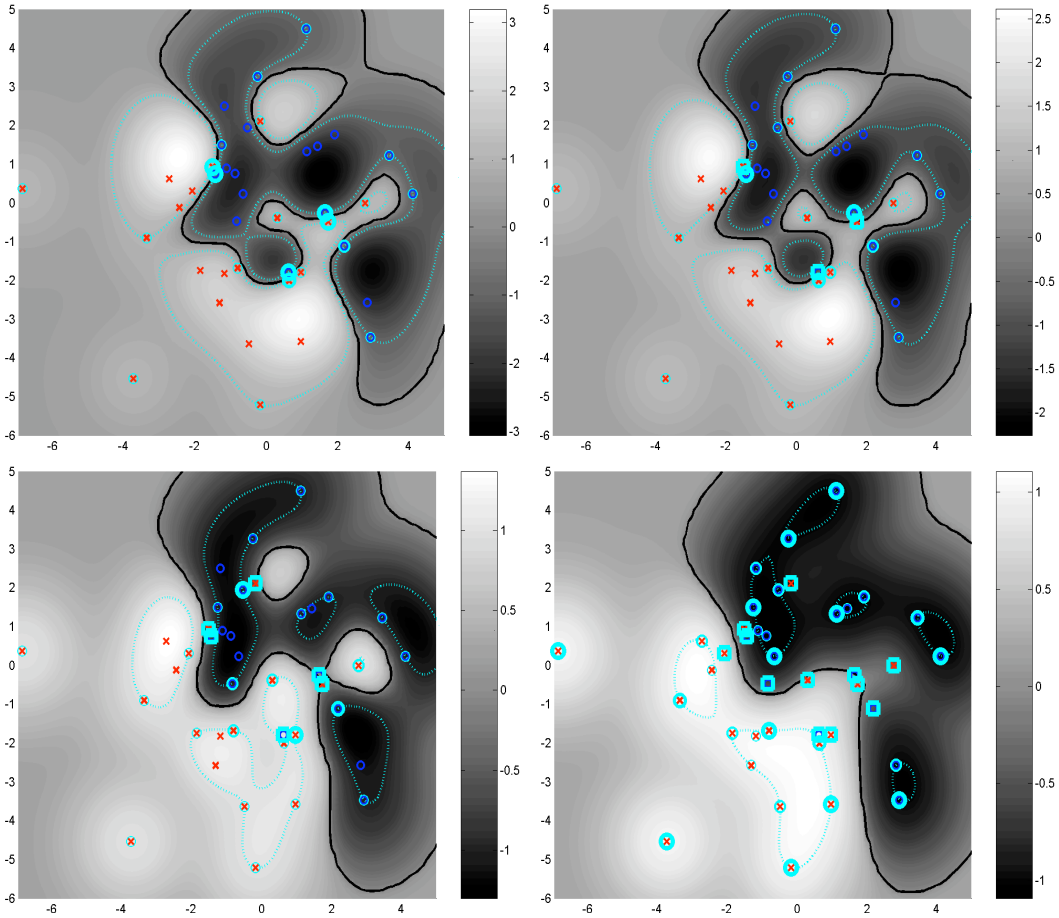
$\sigma =$ 0.1,0.25.0.5.0.75.1,10



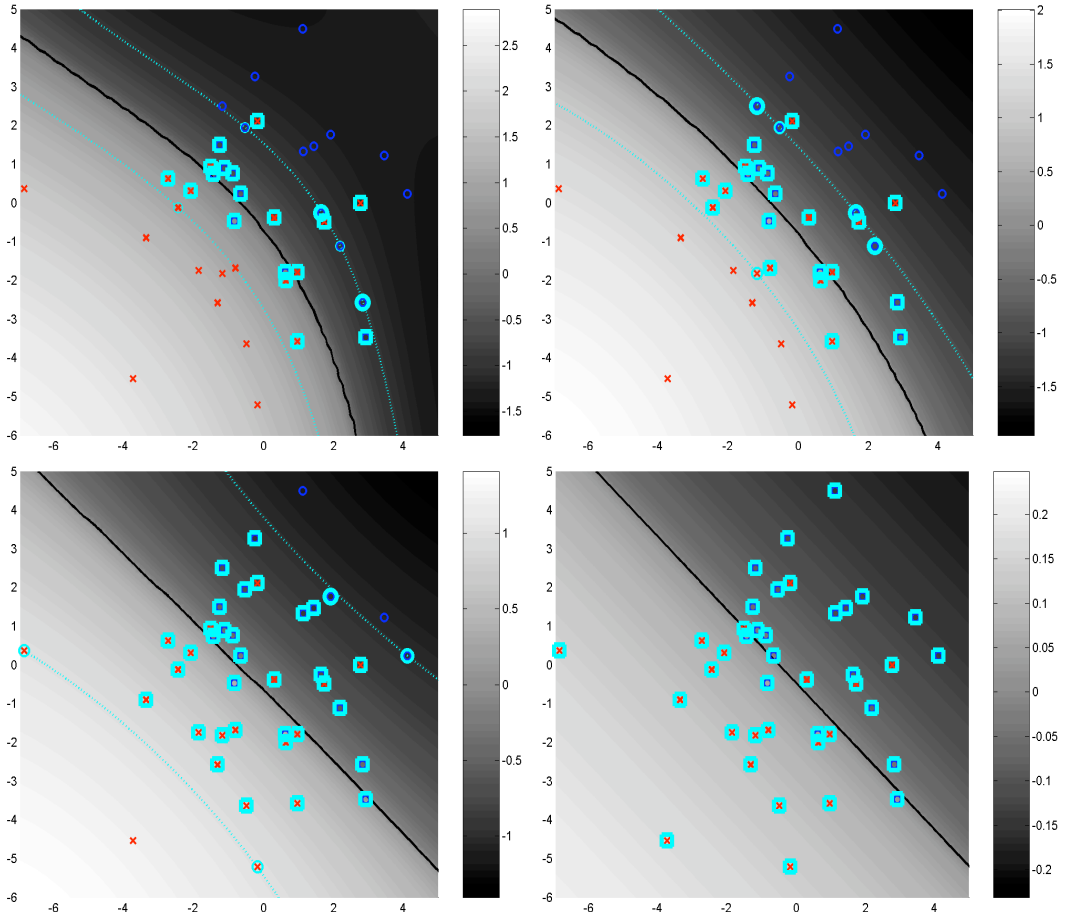
code

```
randn('seed',1);
m=20;
d=1;
s=2;
x1=[randn(m,1)*s-d randn(m,1)*s-d]
x2=[randn(m,1)*s+d randn(m,1)*s+d]
d=data([x1;x2],[ones(m,1);-ones(m,1)]);
a=svm;
sigma=10; (ou sigma=1)
a.C=1;
a.child=kernel('rbf',sigma);
[r a]=train(a,d);
plot(a)
```

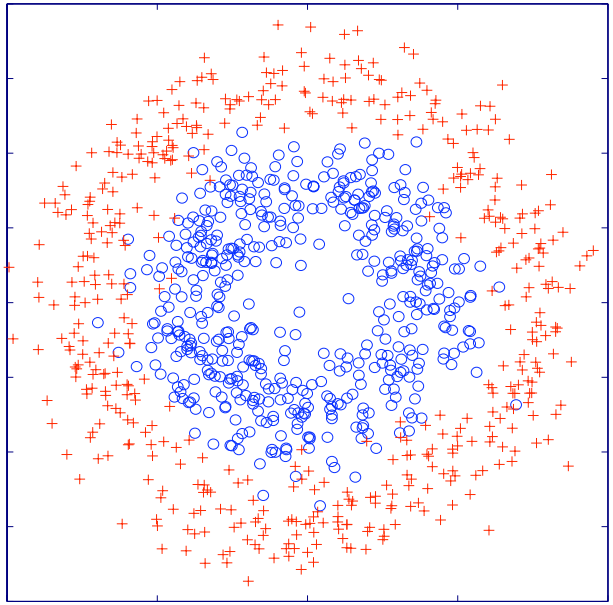
$C =$ inf, 100,10,1

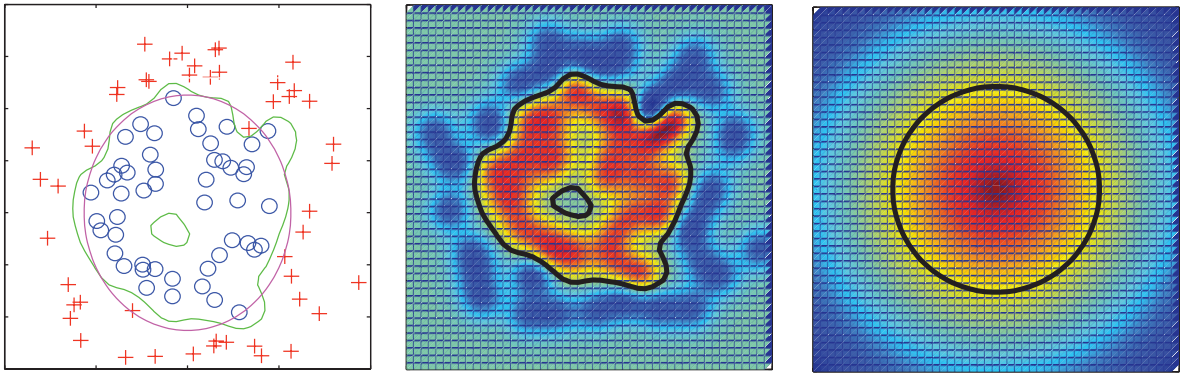


$C =$ **100,10,1,** $\sigma = 10$**;** $\sigma = 100, C = 1$



Choosing the kernel





References

Books

J. Shawe-Taylor, N. Christianini : An introduction to Support Vector Machines (2000)
Kernel based methods for Pattern Analysis (2004)
B. Schölkopf, A. Smola: Learning with Kernels (2002)
Vladimir Vapnik, Statistical Learning Theory (1998)
L. Devroye, L. Györfi, and G. Lugosi. A Probabilistic Theory of Pattern Recognition. Number 31 in Applications of mathematics. Springer, New York, 1996.
Hastie, T., Tibshirani, R. & Friedman, J. (2001), The Elements of Statistical Learning; Data mining, Inference and Prediction, Springer Verlag, New York.

Papers

N. Aronszan. Theory of reproducing kernels. Transactions of the American Mathematical Society, 68:337–404, 1950.

G. Blanchard, P. Massart, R. Vert, and L. Zwald. Kernel projection machine: a new tool for pattern recognition. NIPS 2004.

G. Blanchard, O. Bousquet, and L. Zwald. Statistical properties of kernel princi- pal component analysis. In Proceedings of the 17th. Conference on Learning The- ory (COLT 2004), pages 594–608, 2004.

O. Bousquet, S. Boucheron, and G. Lugosi (2005). Theory of classification: a survey of recent advances. ESAIM: Probability and Statistics.

C.J.C. Burges. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2(2):955-974, 1998. http://citeseer.nj.nec.com/burges98tutorial.html

Girosi F. 1998. An equivalence between sparse approximation and support vector machines. Neural Computation 10(6): 1455–1480.
Lin, Y. (2002), 'Support vector machines and the bayes rule in classification', Data Mining and Knowledge Discovery 6, 259–275.
Lee, Y., Lin, Y. & Wahba, G. (2004), 'Multicategory support vector machines, theory,and application to the classification of microarray data and satellite radiance data', Journal of the American Statistical Association 99, 67–81.
T. Zhang (2004). Statistical behavior and consistency of classification methods based on convex risk minimization (with discussion). Annals of Statistics, 32, pp. 56–85.

Implementations

- libSVM
- SVMlight
- Spider (Matlab)
- kernlab (R)