

# Statistics and learning

## Support Vector Machines

Emmanuel Rachelson and Matthieu Vignes

ISAE SupAero

Friday 29<sup>th</sup> November 2013

# Linearly separable data

Intuition: How would you separate whites and blacks?



images/lin\_sep0.pdf

# Separation hyperplane

images/lin\_sep0.pdf

# Separation hyperplane



# Separation hyperplane



## Separation hyperplane

images/lin\_sep3.pdf

Any separation hyperplane can be written  $(\beta, \beta_0)$  such that:

$$\forall i = 1..N, \beta^T x_i + \beta_0 \geq 0 \text{ if } y_i = +1$$

$$\forall i = 1..N, \beta^T x_i + \beta_0 \leq 0 \text{ if } y_i = -1$$

This can be written:

$$\forall i = 1..N, y_i (\beta^T x_i + \beta_0) \geq 0$$

# Separation hyperplane



But...

$y_i (\beta^T x_i + \beta_0)$  is the  
*signed distance* between  
point  $i$  and  
the hyperplane  $(\beta, \beta_0)$

Margin of a separating hyperplane:  $\min_i y_i (\beta^T x_i + \beta_0) ?$

## Separation hyperplane



images/lin\_sep3.pdf

## Optimal separating hyperplane

Maximize the *margin* between the hyperplane and the data.

$$\max_{\beta, \beta_0} M$$

such that  $\forall i = 1..N, y_i (\beta^T x_i + \beta_0) \geq M$  and  $\|\beta\| = 1$

## Separation hyperplane

images/lin\_sep3.pdf

Let's get rid of  $\|\beta\| = 1$ :

$$\begin{aligned} \forall i = 1..N, \frac{1}{\|\beta\|} y_i (\beta^T x_i + \beta_0) &\geq M \\ \Rightarrow \forall i = 1..N, y_i (\beta^T x_i + \beta_0) &\geq M \|\beta\| \end{aligned}$$

## Separation hyperplane



$$\forall i = 1..N, y_i (\beta^T x_i + \beta_0) \geq M \|\beta\|$$

If  $(\beta, \beta_0)$  satisfies this constraint, then  $\forall \alpha > 0, (\alpha\beta, \alpha\beta_0)$  does too.

Let's choose to have  $\forall i = 1..N, y_i (\beta^T x_i + \beta_0) \geq 1$

then we need to set  $\|\beta\| = \frac{1}{M}$

# Separation hyperplane

images/lin\_sep3.pdf

Now  $M = \frac{1}{\|\beta\|}$ . Geometrical interpretation?

So

$$\max_{\beta, \beta_0} M \Leftrightarrow \min_{\beta, \beta_0} \|\beta\| \Leftrightarrow \min_{\beta, \beta_0} \|\beta\|^2$$

## Separation hyperplane

images/lin\_sep3.pdf

### Optimal separating hyperplane (continued)

$$\min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2$$

such that  $\forall i = 1..N, y_i (\beta^T x_i + \beta_0) \geq 1$

Maximize the *margin*  $M = \frac{1}{\|\beta\|}$  between the hyperplane and the data.

# Optimal separating hyperplane

$$\min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2$$

such that  $\forall i = 1..N, y_i (\beta^T x_i + \beta_0) \geq 1$

It's a QP problem!

# Optimal separating hyperplane

$$\min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2$$

such that  $\forall i = 1..N, y_i (\beta^T x_i + \beta_0) \geq 1$

It's a QP problem!

$$L_P(\beta, \beta_0, \alpha) = \frac{1}{2} \|\beta\|^2 - \sum_{i=1}^N \alpha_i (y_i (\beta^T x_i + \beta_0) - 1)$$

# Optimal separating hyperplane

$$\min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2$$

such that  $\forall i = 1..N, y_i (\beta^T x_i + \beta_0) \geq 1$

It's a QP problem!

$$L_P(\beta, \beta_0, \alpha) = \frac{1}{2} \|\beta\|^2 - \sum_{i=1}^N \alpha_i (y_i (\beta^T x_i + \beta_0) - 1)$$

KKT conditions

$$\begin{cases} \frac{\partial L_P}{\partial \beta} = 0 \Rightarrow \beta = \sum_{i=1}^N \alpha_i y_i x_i \\ \frac{\partial L_P}{\partial \beta_0} = 0 \Rightarrow 0 = \sum_{i=1}^N \alpha_i y_i \\ \forall i = 1..N, \alpha_i (y_i (\beta^T x_i + \beta_0) - 1) = 0 \\ \forall i = 1..N, \alpha_i \geq 0 \end{cases}$$

# Optimal separating hyperplane

$$\min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2$$

such that  $\forall i = 1..N, y_i (\beta^T x_i + \beta_0) \geq 1$

It's a QP problem!

$$\forall i = 1..N, \alpha_i (y_i (\beta^T x_i + \beta_0) - 1) = 0$$

Two possibilities:

- $\alpha_i > 0$ , then  $y_i (\beta^T x_i + \beta_0) = 1$ :  $x_i$  is on the margin's boundary
- $\alpha_i = 0$ , then  $x_i$  is anywhere on the boundary or further  
... but does not participate in  $\beta$ .

$$\beta = \sum_{i=1}^N \alpha_i y_i x_i$$

The  $x_i$  for which  $\alpha_i > 0$  are called *Support Vectors*.

# Optimal separating hyperplane

$$\min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2$$

such that  $\forall i = 1..N, y_i (\beta^T x_i + \beta_0) \geq 1$

It's a QP problem!

Dual problem:  $\max_{\alpha \in \mathbb{R}^{+N}} L_D(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j x_i^T x_j$

such that  $\sum_{i=1}^N \alpha_i y_i = 0$

Solving the dual problem is a maximization in  $\mathbb{R}^N$ , rather than a (constrained) minimization in  $\mathbb{R}^n$ . Usual algorithm: SMO=Sequential Minimal Optimization.

# Optimal separating hyperplane

$$\min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2$$

such that  $\forall i = 1..N, y_i (\beta^T x_i + \beta_0) \geq 1$

It's a QP problem!

And  $\beta_0$ ?

Solve  $\alpha_i (y_i (\beta^T x_i + \beta_0) - 1) = 0$  for any  $i$  such that  $\alpha_i > 0$

# Optimal separating hyperplane

images/lin\_sep3.pdf

Overall:

$$\beta = \sum_{i=1}^N \alpha_i y_i x_i$$

With  $\alpha_i > 0$  only for  $x_i$  support vectors.

Prediction:  $f(x) = \text{sign} (\beta^T x + \beta_0) = \text{sign} \left( \sum_{i=1}^N \alpha_i y_i x_i^T x + \beta_0 \right)$

# Non-linearly separable data?

images/non\_lin\_sep0.pdf

# Non-linearly separable data?

images/non\_lin\_sep1.pdf

# Non-linearly separable data?

images/non\_lin\_sep2.pdf

# Non-linearly separable data?

images/non\_lin\_sep3.pdf

Slack variables  $\xi = (\xi_1, \dots, \xi_N)$

$$\left. \begin{array}{l} y_i(\beta^T x_i + \beta_0) \geq M - \xi_i \\ \text{or} \\ y_i(\beta^T x_i + \beta_0) \geq M(1 - \xi_i) \end{array} \right\} \text{and } \xi_i \geq 0 \text{ and } \sum_{i=1}^N \xi_i \leq K$$

## Non-linearly separable data?

images/non\_lin\_sep3.pdf

$y_i(\beta^T x_i + \beta_0) \geq M(1 - \xi_i) \Rightarrow$  misclassification if  $\xi_i \geq 1$

$$\sum_{i=1}^N \xi_i \leq K \Rightarrow$$
 maximum  $K$  misclassifications

# Non-linearly separable data?

images/non\_lin\_sep3.pdf

## Optimal separating hyperplane

$$\min_{\beta, \beta_0} \|\beta\|$$

such that  $\forall i = 1..N, \begin{cases} y_i (\beta^T x_i + \beta_0) \geq 1 - \xi_i, \\ \xi_i \geq 0, \sum_{i=1}^N \xi_i \leq K \end{cases}$

# Non-linearly separable data?

images/non\_lin\_sep3.pdf

## Optimal separating hyperplane

$$\min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^N \xi_i$$

such that  $\forall i = 1..N, \begin{cases} y_i (\beta^T x_i + \beta_0) \geq 1 - \xi_i, \\ \xi_i \geq 0 \end{cases}$

# Optimal separating hyperplane

Again a QP problem.

$$L_P = \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i (y_i (\beta^T x_i + \beta_0) - (1 - \xi_i)) - \sum_{i=1}^N \mu_i \xi_i$$

KKT conditions

$$\left\{ \begin{array}{l} \frac{\partial L_P}{\partial \beta} = 0 \Rightarrow \beta = \sum_{i=1}^N \alpha_i y_i x_i \\ \frac{\partial L_P}{\partial \beta_0} = 0 \Rightarrow 0 = \sum_{i=1}^N \alpha_i y_i \\ \frac{\partial L_P}{\partial \xi} = 0 \Rightarrow \alpha_i = C - \mu_i \\ \forall i = 1..N, \alpha_i (y_i (\beta^T x_i + \beta_0) - (1 - \xi_i)) = 0 \\ \forall i = 1..N, \mu_i \xi_i = 0 \\ \forall i = 1..N, \alpha_i \geq 0, \mu_i \geq 0 \end{array} \right.$$

# Optimal separating hyperplane

Dual problem:  $\max_{\alpha \in \mathbb{R}^{+N}} L_D(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j x_i^T x_j$

such that  $\sum_{i=1}^N \alpha_i y_i = 0$

and  $0 \leq \alpha_i \leq C$

# Optimal separating hyperplane

$$\alpha_i (y_i (\beta^T x_i + \beta_0) - (1 - \xi_i)) = 0 \text{ and } \beta = \sum_{i=1}^N \alpha_i y_i x_i$$

Again:

- ▶  $\alpha_i > 0$ , then  $y_i (\beta^T x_i + \beta_0) = 1 - \xi_i$ :  $x_i$  is a *support vector*.  
Among these:
  - ▶  $\xi_i = 0$ , then  $0 \leq \alpha_i \leq C$
  - ▶  $\xi_i > 0$ , then  $\alpha_i = C$  (because  $\mu_i = 0$ , because  $\mu_i \xi_i = 0$ )
- ▶  $\alpha_i = 0$ , then  $x_i$  does not participate in  $\beta$ .

# Optimal separating hyperplane



images/non\_lin\_sep1.pdf

Overall:

$$\beta = \sum_{i=1}^N \alpha_i y_i x_i$$

With  $\alpha_i > 0$  only for  $x_i$  support vectors.

Prediction:  $f(x) = \text{sign} (\beta^T x + \beta_0) = \text{sign} \left( \sum_{i=1}^N \alpha_i y_i x_i^T x + \beta_0 \right)$

# Non-linear SVMs?

## Key remark

$h : \begin{cases} X & \rightarrow \mathcal{H} \\ x & \mapsto h(x) \end{cases}$  is a mapping to a p-dimensional Euclidean space.  
( $p \gg n$ , possibly infinite)

SVM classifier in  $\mathcal{H}$ :  $f(x') = sign \left( \sum_{i=1}^N \alpha_i y_i \langle x'_i, x' \rangle + \beta_0 \right).$

Suppose  $K(x, x') = \langle h(x), h(x') \rangle$ ,

Then:

$$f(x) = sign \left( \sum_{i=1}^N \alpha_i y_i K(x_i, x) + \beta_0 \right).$$

# Kernels

## Kernel

$K(x, y) = \langle h(x), h(y) \rangle$  is called a kernel function.

# Kernels

## Kernel

$K(x, y) = \langle h(x), h(y) \rangle$  is called a kernel function.

Example:

$$X = \mathbb{R}^2, \quad \mathcal{H} = \mathbb{R}^3, \quad h(x) = \begin{pmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{pmatrix}$$

$$K(x, y) = h(x)^T h(y)$$

# Kernels

## Kernel

$K(x, y) = \langle h(x), h(y) \rangle$  is called a kernel function.

What if we knew that  $K(\cdot, \cdot)$  is a kernel, without explicitly building  $h$ ?

The SVM would be a *linear* classifier in  $\mathcal{H}$  but we would never have to compute  $h(x)$  for training or prediction!

This is called the *kernel trick*.

# Kernels

## Kernel

$K(x, y) = \langle h(x), h(y) \rangle$  is called a kernel function.

Under what conditions is  $K(\cdot, \cdot)$  an acceptable kernel?

Answer: if it is an inner product on a (separable) Hilbert space.

In more general words, we are interested in *positive, definite kernel* on a Hilbert space:

## Positive Definite Kernels

$K(\cdot, \cdot)$  is a positive definite kernel on  $X$  if

$$\forall n \in \mathbb{N}, x \in X^n \text{ and } c \in \mathbb{R}^n, \sum_{i,j=1}^n c_i c_j K(x_i, x_j) \geq 0$$

# Kernels

## Kernel

$K(x, y) = \langle h(x), h(y) \rangle$  is called a kernel function.

## Mercer's condition

Given  $K(x, y)$ , if:

$$\forall g(x) / \int g(x)^2 dx < \infty, \iint K(x, y)g(x)g(y)dxdy \geq 0$$

Then, there exists a mapping  $h(\cdot)$  such that:

$$K(x, y) = \langle h(x), h(y) \rangle$$

# Kernels

## Kernel

$K(x, y) = \langle h(x), h(y) \rangle$  is called a kernel function.

Examples of kernels:

- ▶ polynomial  $K(x, y) = (1 + \langle x, y \rangle)^d$
- ▶ radial basis  $K(x, y) = e^{-\gamma \|x-y\|^2}$  (very often used in  $\mathbb{R}^n$ )
- ▶ sigmoid  $K(x, y) = \tanh(\kappa_1 \langle x, y \rangle + \kappa_2)$

# Kernels

## Kernel

$K(x, y) = \langle h(x), h(y) \rangle$  is called a kernel function.

What do you think:

Is it good or bad to send all data points in a feature space with  $p \gg n$ ?

# SVM and kernels for classification

$$\min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^N \xi_i$$

such that  $\forall i = 1..N, \begin{cases} y_i (\beta^T h(x_i) + \beta_0) \geq 1 - \xi_i, \\ \xi_i \geq 0 \end{cases}$

# SVM and kernels for classification

$$\min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^N \xi_i$$

such that  $\forall i = 1..N, \begin{cases} y_i (\beta^T h(x_i) + \beta_0) \geq 1 - \xi_i, \\ \xi_i \geq 0 \end{cases}$

Dual problem:  $\max_{\alpha \in \mathbb{R}^{+N}} L_D(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \langle h(x_i), h(x_j) \rangle$

such that  $\sum_{i=1}^N \alpha_i y_i = 0$

and  $0 \leq \alpha_i \leq C$

# SVM and kernels for classification

$$\min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^N \xi_i$$

such that  $\forall i = 1..N, \begin{cases} y_i (\beta^T h(x_i) + \beta_0) \geq 1 - \xi_i, \\ \xi_i \geq 0 \end{cases}$

Dual problem:  $\max_{\alpha \in \mathbb{R}^{+N}} L_D(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(x_i, x_j)$

such that  $\sum_{i=1}^N \alpha_i y_i = 0$

and  $0 \leq \alpha_i \leq C$

# SVM and kernels for classification



images/non\_lin1.png

Overall:

$$\beta = \sum_{i=1}^N \alpha_i y_i x_i$$

With  $\alpha_i > 0$  only for  $x_i$  support vectors.

Prediction:  $f(x) = sign (\beta^T x + \beta_0) = sign \left( \sum_{i=1}^N \alpha_i y_i K(x_i, x) + \beta_0 \right)$

# Why would you use SVM?

- ▶ With kernels, sends the data into higher (sometimes infinite) dimension feature space, where the data is separable / linearly interpolable.
- ▶ Produces a sparse predictor (many coefficients are zero).
- ▶ Automatically maximizes margin (thus generalization error?).
- ▶ Performs very well on complex, non-linearly separable / fittable data.

# SVM for regression

Now we don't want to separate, but to fit.

Contradictory goals?

- ▶ Fit the data: minimize  $\sum_{i=1}^N V(y_i - f(x_i))$   
 $V$  is a loss function.
- ▶ Keep large margins: minimize  $\|\beta\|$

# SVM for regression

Now we don't want to separate, but to fit.

Contradictory goals?

- ▶ Fit the data: minimize  $\sum_{i=1}^N V(y_i - f(x_i))$   
 $V$  is a loss function.
- ▶ Keep large margins: minimize  $\|\beta\|$

## Support Vector Regression

$$\min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^N V(y_i - \beta^T x_i + \beta_0))$$

# Loss functions

$\epsilon$ -insensitive	$V(z) = \begin{cases} 0 & \text{if }  z  \leq \epsilon \\  z  - \epsilon & \text{otherwise} \end{cases}$
Laplacian	$V(z) =  z $
Gaussian	$V(z) = \frac{1}{2}z^2$
Huber's robust loss	$V(z) = \begin{cases} \frac{1}{2\sigma}z^2 & \text{if }  z  \leq \sigma \\  z  - \frac{\sigma}{2} & \text{otherwise} \end{cases}$

images/epsilon\_insensitive2.png

# $\epsilon$ -SVR

$$\min_{\beta, \beta_0} \frac{\lambda}{2} \|\beta\|^2 + C \sum_{i=1}^N (\xi_i + \xi_i^*)$$

subject to 
$$\begin{cases} y_i - \langle \beta, x_i \rangle - \beta_0 & \leq \epsilon + \xi_i \\ \langle \beta, x_i \rangle + \beta_0 - y_i & \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* & \geq 0 \end{cases}$$

# $\epsilon$ -SVR

$$\min_{\beta, \beta_0} \frac{\lambda}{2} \|\beta\|^2 + C \sum_{i=1}^N (\xi_i + \xi_i^*)$$

subject to 
$$\begin{cases} y_i - \langle \beta, x_i \rangle - \beta_0 & \leq \epsilon + \xi_i \\ \langle \beta, x_i \rangle + \beta_0 - y_i & \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* & \geq 0 \end{cases}$$

As previously, this is a QP problem.

$$L_P = \frac{\lambda}{2} \|\beta\|^2 + C \sum_{i=1}^N (\xi_i + \xi_i^*) - \sum_{i=1}^N \alpha_i (\epsilon + \xi_i - y_i + \langle \beta, x_i \rangle + \beta_0) \\ - \sum_{i=1}^N \alpha_i^* (\epsilon + \xi_i^* + y_i - \langle \beta, x_i \rangle - \beta_0) - \sum_{i=1}^N (\eta_i \xi_i + \eta_i^* \xi_i^*)$$

## $\epsilon$ -SVR cont'd

$$L_D = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) \langle x_i, x_j \rangle$$
$$- \epsilon \sum_{i=1}^N (\alpha_i + \alpha_i^*) + \sum_{i=1}^N y_i (\alpha_i - \alpha_i^*)$$

Dual optimization problem:

$$\max_{\alpha} L_D$$

subject to 
$$\begin{cases} \sum_{i=1}^N (\alpha_i - \alpha_i^*) = 0 \\ \alpha_i, \alpha_i^* \in [0, C] \end{cases}$$

## $\epsilon$ -SVR, support vectors

KKT conditions:

$$\begin{cases} \alpha_i (\epsilon + \xi_i - y_i + \langle \beta, x_i \rangle + \beta_0) = 0 \\ \alpha_i^* (\epsilon + \xi_i^* - y_i + \langle \beta, x_i \rangle + \beta_0) = 0 \\ (C - \alpha_i) \xi_i = 0 \\ (C - \alpha_i^*) \xi_i^* = 0 \end{cases}$$

- ▶ if  $\alpha_i^{(*)} = 0$ , then  $\xi_i^{(*)} = 0$ : points inside the  $\epsilon$ -insensitivity “tube” don’t participate in  $\beta$
- ▶ if  $\alpha_i^{(*)} > 0$ , then
  - ▶ if  $\xi_i^{(*)} = 0$ , then  $x_i$  is exactly on the border of the “tube”,  $\alpha_i^{(*)} \in [0, C]$
  - ▶ if  $\xi_i^{(*)} > 0$ , then  $\alpha_i^{(*)} = C$ : outliers are support vectors.



# SVR prediction

$$f(x) = \sum_{i=1}^N (\alpha_i - \alpha_i^*) \langle x_i, x \rangle + \beta_0$$

# Kernels and SVR?

Just as you would expect it!  
Left to you as an exercice.

# Why would you use SVM?

- ▶ With kernels, sends the data into higher (sometimes infinite) dimension feature space, where the data is separable / linearly interpolable.
- ▶ Produces a sparse predictor (many coefficients are zero).
- ▶ Automatically maximizes margin (thus generalization error?).
- ▶ Performs very well on complex, non-linearly separable / fittable data.

## Further reading / tutorials

### **A tutorial on Support Vector Machines for Pattern Recognition.**

C. J. C. Burges, *Data Mining and Knowledge Discovery*, **2**, 131–167, (1998).

### **A tutorial on Support Vector Regression.**

A. J. Smola and B. Schölkopf, *Journal of Statistics and Computing*, **14**(3), 199-222, (2004).