# *An Introduction to Statistical Machine Learning - Support Vector Machines -*

**Ronan Collobert**

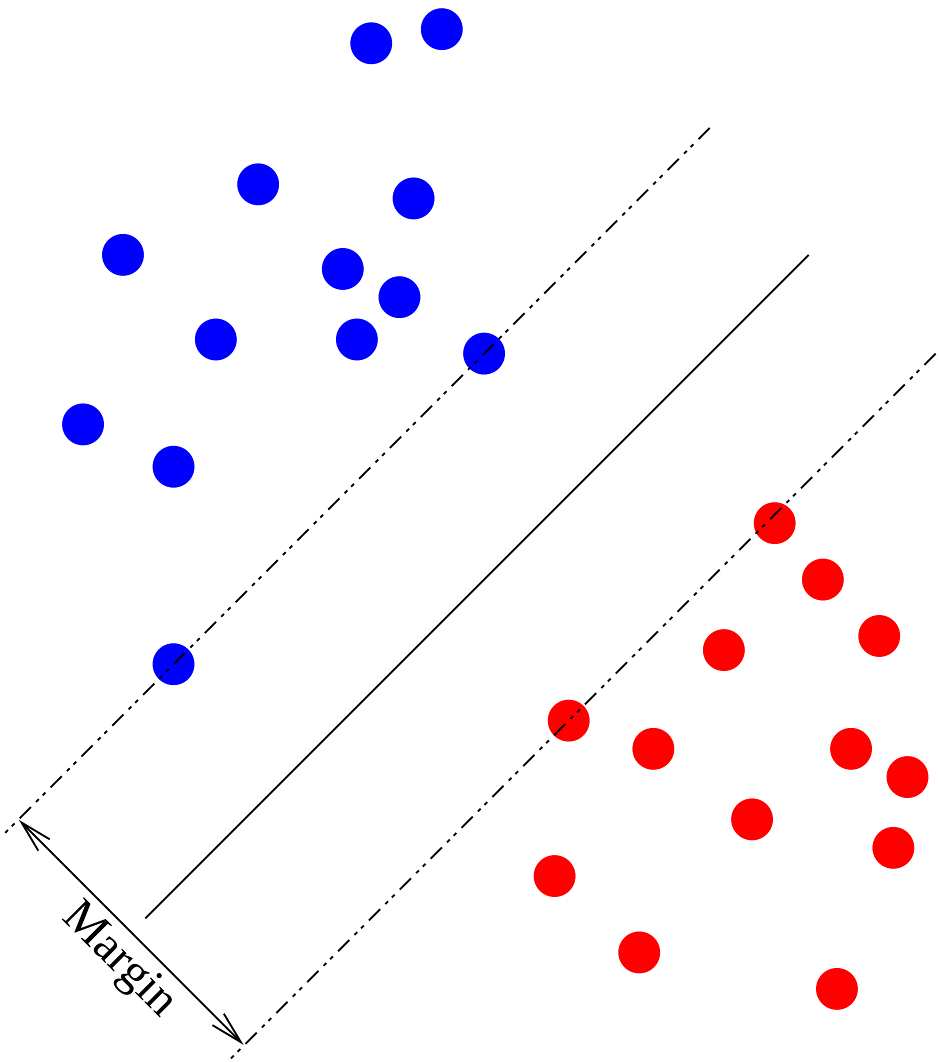
colober@idiap.ch

Dalle Molle Institute for Perceptual Artificial Intelligence (IDIAP)
CP 592, rue du Simplon 4
1920 Martigny, Switzerland
http://www.idiap.ch/~collober

# Support Vector Machines

1. The aim of SVMs
2. Linear SVMs and soft margin
3. Solving the SVMs problem using a Lagrangian method
4. Kernel trick
5. Support Vector Regression

# SVMs in Two Slides (1/2)



# SVMs in Two Slides (2/2)

**ENTERING MATHEMATICAL AREA**

# Few Notations (1/2)

- Training set:

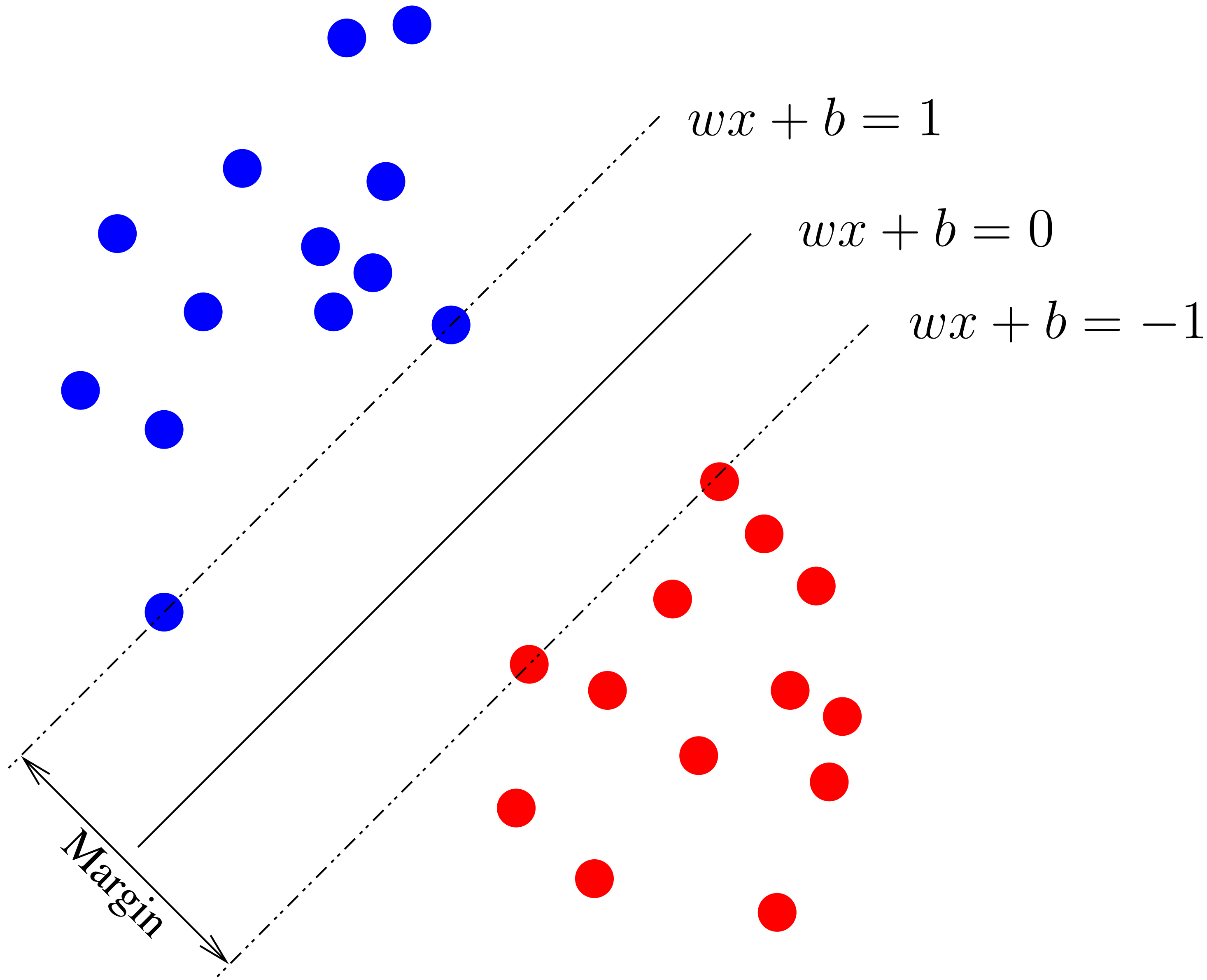
$$(x_t, y_t)_{t=1\ldots T} \in \mathbb{R}^d \times \{-1, 1\}$$

- We would like to find *one* hyperplane

$$wx + b = 0 \quad (w \in \mathbb{R}^d,\ b \in \mathbb{R})$$

which separates the two classes and maximizes the margin.

# Few Notations (2/2)



# My First Equation

- Margin to *maximize*:

$$\text{dist}(wx + b = 1,\ wx + b = -1) = \frac{2}{\|w\|}$$

- We would like to minimize:

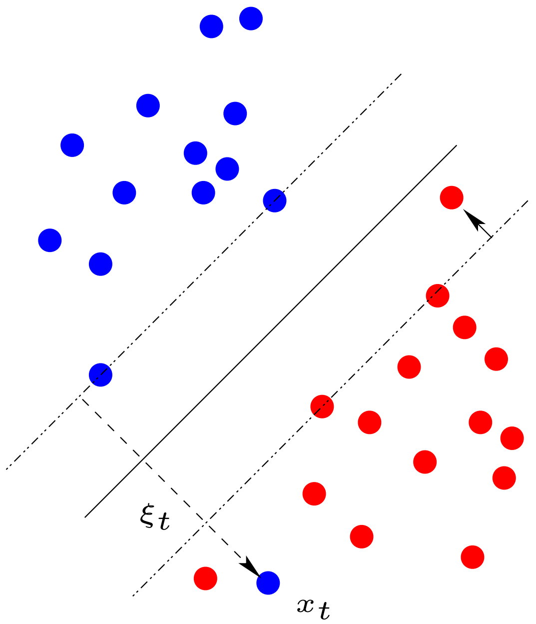
$$J(w,\ b) = \frac{\|w\|^2}{2}$$

Under the constraints:

$$y_t(wx_t + b) \geq 1 \quad \forall t$$

# A Bug

This minimization problem does not have any solution if the two classes are not separable.



# Fixing The Bug: "Soft" Margin

- Relax the constraints: use a soft margin instead of a hard margin.
- We would like to minimize:

$$J(w, b, \xi) = \frac{\|w\|^2}{2} + C \sum_{t=1}^{T} \xi_t$$

Under the constraints:

$$y_t(wx_t + b) \geq 1 - \xi_t \quad \forall t$$

$$\xi_t \geq 0 \quad \forall t$$

# Two Slides on Lagrangian Method (1/2)

- We want to find $u$ such that:

$$J(u) = \inf_{v \in U} J(v)$$

$$u \in U = \{v \in \mathbb{R}^n \ : \ \varphi_i(v) \leq 0 \quad \forall i\}$$

- Introduce the Lagrangian:

$$L(v,\, \mu) = J(v) + \sum_i \mu_i \, \varphi_i(v) \qquad (\mu_i \geq 0)$$

# Two Slides on Lagrangian Method (2/2)

- Theorem: If $(u, \lambda)$ is a saddle point of the Lagrangian $L$, then $(u, \lambda)$ is a solution of the constrained minimization problem.

- $(u, \lambda)$ is a saddle point of the function $L$ if $u$ is a minimum for the function $v \mapsto L(v, \lambda)$ and $\lambda$ is a maximum for the function $\mu \mapsto L(u, \mu)$.

# Back to SVMs

- Our Lagrangian:

$$L(w,\, b,\, \xi,\, \alpha,\, \mu) = J(w,\, b,\, \xi) + \sum_t \alpha_t [1 - \xi_t - y_t(wx_t + b)] - \sum_t \mu_t \xi_t$$

$$= \frac{\|w\|^2}{2} + C \sum_{t=1}^{T} \xi_t + \sum_t \alpha_t [1 - \xi_t - y_t(wx_t + b)] - \sum_t \mu_t \xi_t$$

$$(\alpha_t \geq 0 \quad \text{and} \quad \mu_t \geq 0)$$

- Look for $(w,\, b,\, \xi)$ minimum of $L$:

$$\frac{\partial L}{\partial w} = 0 \quad \Leftrightarrow \quad w = \sum_t \alpha_t y_t \, x_t$$

$$\frac{\partial L}{\partial b} = 0 \quad \Leftrightarrow \quad \sum_t \alpha_t y_t = 0$$

$$\frac{\partial L}{\partial \xi} = 0 \quad \Leftrightarrow \quad C - \alpha_t - \mu_t = 0$$

# The Nightmare Continues...

- Insert in the Lagrangian:

$$L = \sum_t \alpha_t - \frac{1}{2} \sum_{s,t} \alpha_s \alpha_t y_s y_t \, x_s x_t$$

$$0 \leq \alpha_t \leq C$$

$$\sum_t \alpha_t y_t = 0$$

$$w = \sum_t \alpha_t y_t \, x_t$$

- Look for $(\alpha, \mu)$ maximum of $L$:

$$\begin{aligned} \alpha_t [1 - \xi_t - y_t (w x_t + b)] &= 0 \\ \mu_t \xi_t &= 0 \end{aligned}$$

# Yes!

- Finaly, we "just" have to minimize

$$\alpha \mapsto \frac{1}{2}\alpha^{\mathbf{T}} Q \alpha - \alpha^{\mathbf{T}} 1$$

where

$$Q_{ij} = y_i y_j \, x_i x_j$$

Under the constraints

$$0 \leq \alpha_t \leq C \quad \text{and} \quad \sum_t \alpha_t y_t = 0$$

- Then we obtain $w$ and $b$ with

$$w = \sum_t \alpha_t y_t \, x_t$$

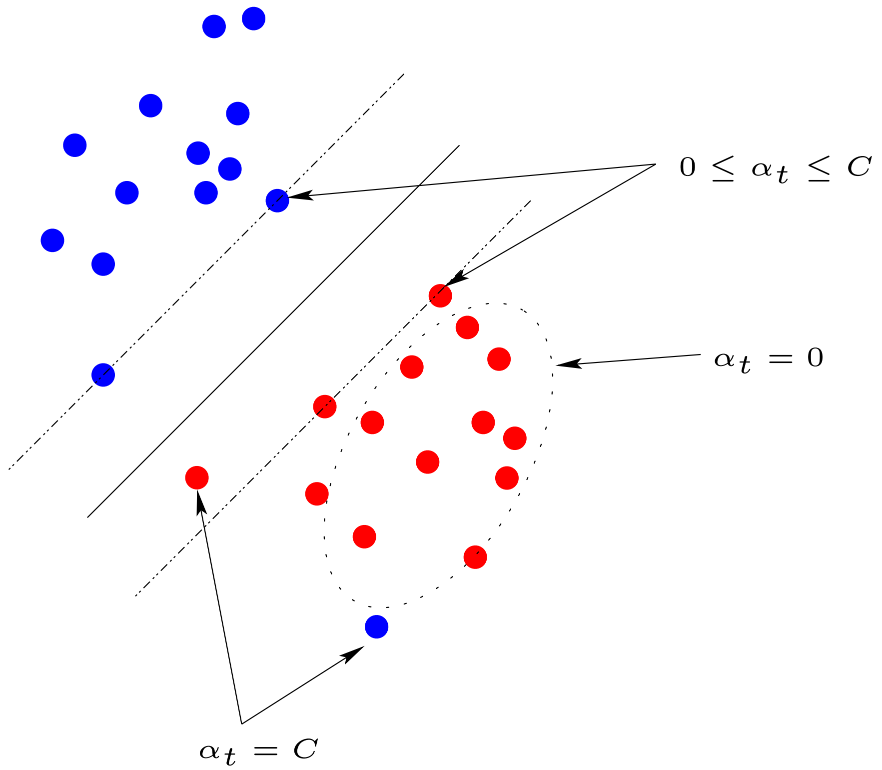
$$\alpha_t [1 - \xi_t - y_t(w x_t + b)] = 0$$

# Support Vector Etymology

- Note that the decision function could be rewritten as:

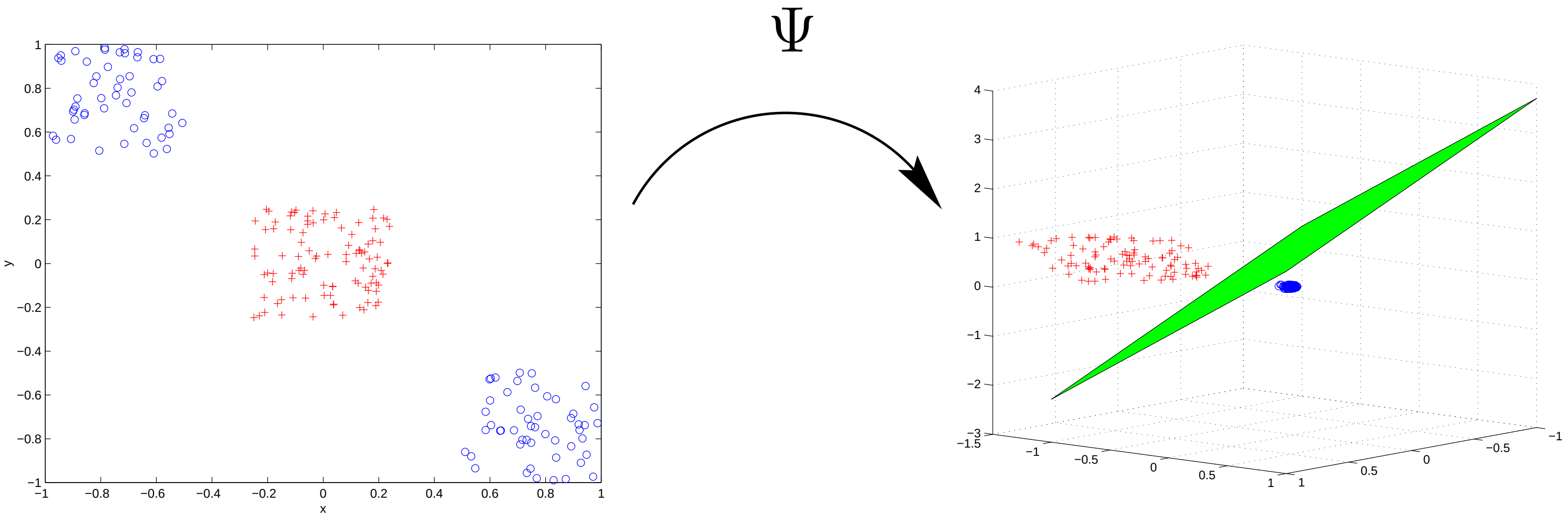
$$x \mapsto \sum_t \alpha_t y_t \, x_t x + b$$

- Training examples $x_t$ with $\alpha_t \neq 0$ are support vectors.



# Non Linear SVMs

- Project the data into a higher dimensional space: it should be easier to separate the two classes.
- Given a function $\Psi : \mathbb{R}^d \rightarrow F$, work with $\Psi(x_t)$ instead of working with $x_t$.



# The Kernel Trick

- Note that we have only dot products $\Psi(x_s)\Psi(x_t)$ to compute.
- Unfortunately, it could be expensive in a high dimensional space.
- Use instead a kernel: a function $(x, z) \mapsto k(x, z)$ which represents a dot product in a "hidden" feature space.

$$k(x, z) = \Psi(x)\Psi(z)$$

- Example: instead of

$$\Psi(x) = \begin{pmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{pmatrix}$$

use

$$k(x, z) = (xz)^2$$

# Common Kernels

- Polynomial:

$$k(x,\ z) = (u\,xz + v)^p \quad (u \in \mathbb{R},\ v \in \mathbb{R},\ p \in \mathbb{N}_+^*)$$

- Gaussian:

$$k(x,\ z) = \exp\left(-\frac{\|x-z\|^2}{2\sigma^2}\right) \quad (\sigma \in \mathbb{R}_+^*)$$

- ⚠ The function

$$k(x,\ z) = \tanh(u\,xz + v)$$

is not a kernel!

# Final Abstract

- Choose a kernel $k()$.
- Minimize

$$\alpha \;\mapsto\; \frac{1}{2}\alpha^{\mathbf{T}} Q \alpha - \alpha^{\mathbf{T}} 1$$

where

$$Q_{ij} = y_i y_j \, k(x_i, x_j)$$

Under the constraints

$$0 \;\leq\; \alpha_t \;\leq\; C \quad \text{and} \quad \sum_t \alpha_t y_t = 0$$

- For $0 < \alpha_t < C$, compute $b$ using

$$1 - y_t \left[ \sum_s \alpha_s y_s \, k(x_s, x_t) + b \right] \;=\; 0$$

# Final Abstract

- The decision function will be

$$x \;\mapsto\; \text{sign}\left(\sum_t \alpha_t y_t \, k(x_t, x) + b\right)$$

# Facts to Remember

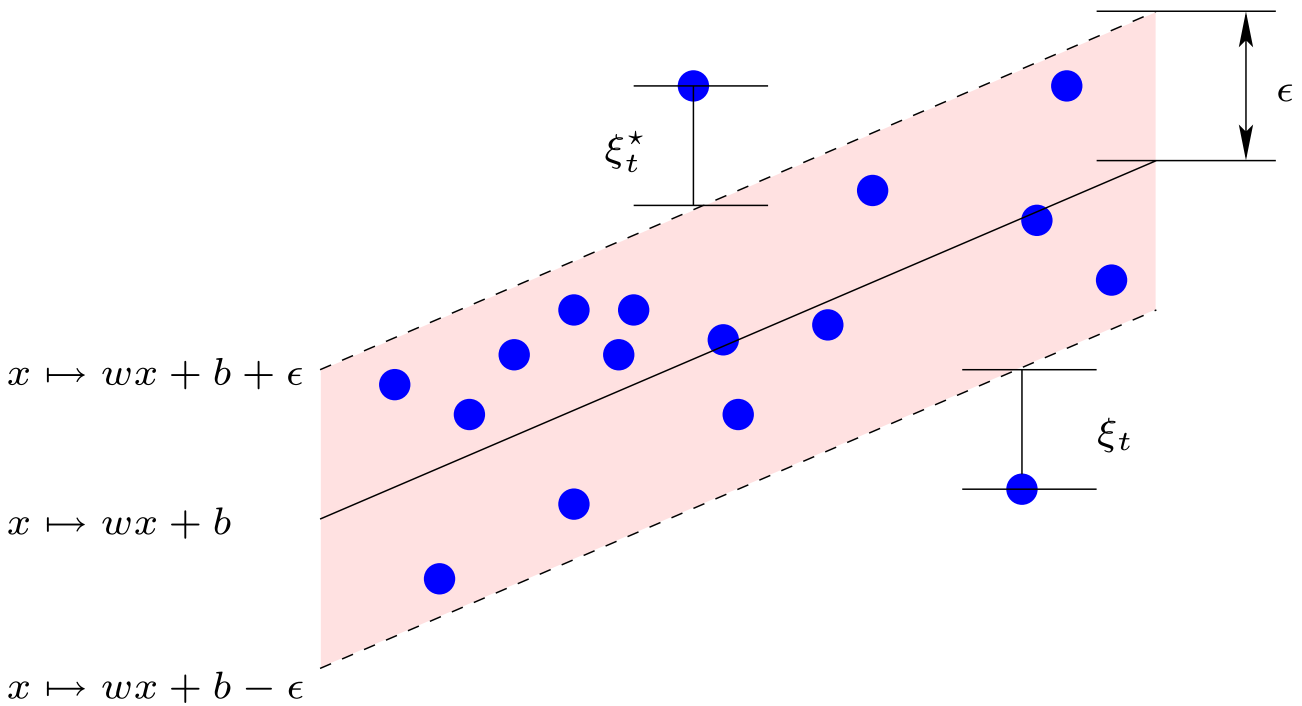
- SVMs maximize the margin (*in the feature space*)
- Use the soft margin trick
- Project the data into a higher dimensional space for non-linear relations
- Kernels simplify the computation
- A Lagrangian method leads to a "nice" quadratic minimization problem under constraints.

# SVMs in Practice

- In order to tune the capacity, the kernel is the most important parameter to choose.
  - Polynomial kernel: increasing the degree will increase the capacity.
  - Gaussian kernel: increasing $\sigma$ will decrease the capacity.
- Tune $C$, the trade-off between the margin and the errors.
  - For non-noisy data sets, $C$ usually has not much influence.
  - Carefully choose $C$ for noisy data sets: small values usually give better results.

# Two Bonus Slides: SVMs in Regression (1/2)

- We are looking for an hyperplane $x \mapsto wx + b$ such that...



# Two Bonus Slides: SVMs in Regression (2/2)

- We would like to minimize

$$\frac{1}{2}\|w\|^2 + C\sum_t |wx_t + b - y_t|_\epsilon$$

where

$$|u|_\epsilon = \max\left(0, |u| - \epsilon\right)$$

- "Epsilon insensitive loss": we "ignore" errors lower than $\epsilon$.
- Equivalent to minimize

$$\frac{1}{2}\|w\|^2 + C\sum_t (\xi_t + \xi_t^\star)$$

under the constraints

$$(wx_t + b) - y_t \leq \epsilon + \xi_t$$

$$y_t - (wx_t + b) \leq \epsilon + \xi_t^\star$$

$$\xi_t,\ \xi_t^\star \geq 0$$

# Other kernel methods

- Multi-class SVMs
- Kernel PCA
- Gaussian Processes
- `http://www.kernel-machines.org`