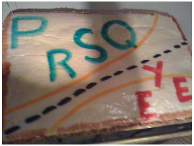
SVM and Kernel machine
linear and non-linear classification

Stéphane Canu
asi.insa-rouen.fr/enseignants/~scanu
scanu@insa-rouen.fr

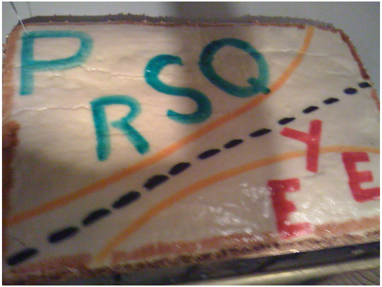


Workshop on Machine Learning

February 15, 2019

Road map

1. Supervised classification and prediction
2. Linear SVM
   - Separating hyperplanes
   - Linear SVM: the problem
   - Optimization in 5 slides
   - Dual formulation of the linear SVM
   - The non separable case
3. Kernels
4. Kernelized support vector machine

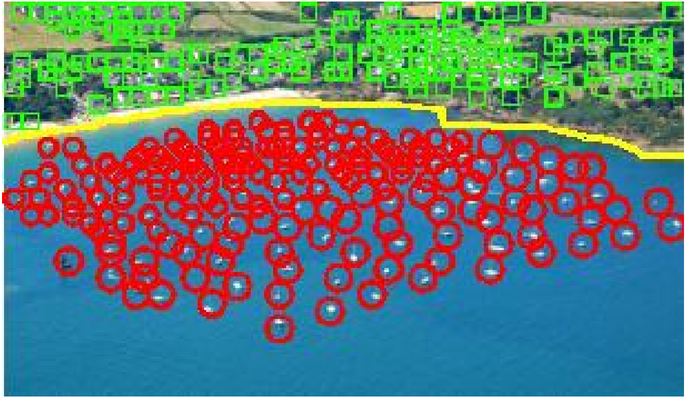


Supervised classification as Learning from examples

The task, use longitude and latitude to predict: is it a boat or a house?

credit: A Gentle Introduction to Support Vector Machines in Biomedicine A. Statnikov, D. Hardin, I. Guyon and C. F. Aliferis
www.nyuinformatics.org/downloads/supplements/SVM_Tutorial_2010/Final_WB.pdf

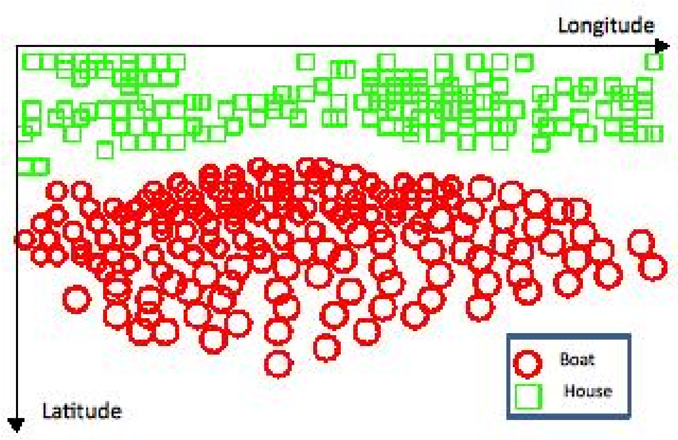
Supervised classification as Learning from examples



Using (red and green) labelled examples learn a (yellow) decision rule

credit: **A Gentle Introduction to Support Vector Machines in Biomedicine A. Statnikov, D. Hardin, I. Guyon and C. F. Aliferis**
`www.nyuinformatics.org/downloads/supplements/SVM_Tutorial_2010/Final_WB.pdf`

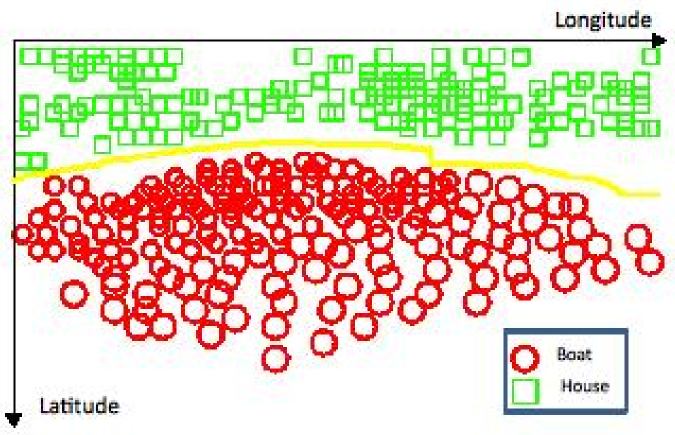
Supervised classification as Learning from examples



Using (red and green) labelled examples...

credit: A Gentle Introduction to Support Vector Machines in Biomedicine A. Statnikov, D. Hardin, I. Guyon and C. F. Aliferis
www.nyuinformatics.org/downloads/supplements/SVM_Tutorial_2010/Final_WB.pdf

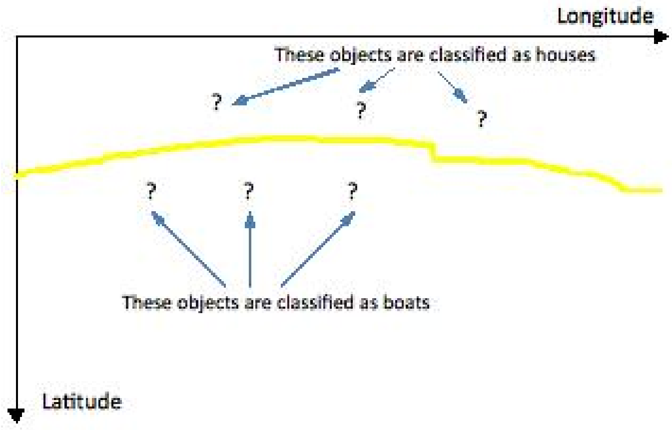
Supervised classification as Learning from examples



Using (red and green) labelled examples... learn a (yellow) decision rule

credit: A Gentle Introduction to Support Vector Machines in Biomedicine A. Statnikov, D. Hardin, I. Guyon and C. F. Aliferis
www.nyuinformatics.org/downloads/supplements/SVM_Tutorial_2010/Final_WB.pdf

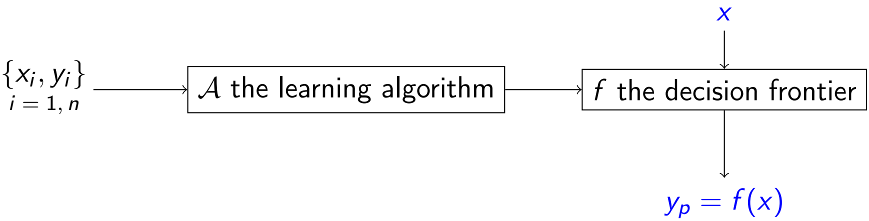
Supervised classification as Learning from examples



Use the decision border to predict unseen objects label

credit: A Gentle Introduction to Support Vector Machines in Biomedicine A. Statnikov, D. Hardin, I. Guyon and C. F. Aliferis
www.nyuinformatics.org/downloads/supplements/SVM_Tutorial_2010/Final_WB.pdf

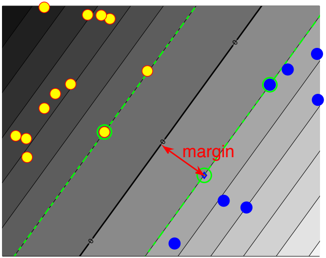
Suppervised classification: the 2 steps



1. the border $\leftarrow$ $Learn(xi, yi, n$ training data$)$ % $\mathcal{A}$ `is SVM_learn`
2. $y_p \leftarrow Predict($unseen x, the border$)$ % f `is SVM_val`

Road map

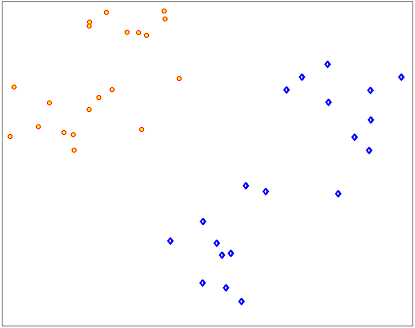
1. Supervised classification and prediction
2. Linear SVM
   - Separating hyperplanes
   - Linear SVM: the problem
   - Optimization in 5 slides
   - Dual formulation of the linear SVM
   - The non separable case
3. Kernels
4. Kernelized support vector machine



"The algorithms for constructing the separating hyperplane considered above will be utilized for developing a battery of programs for pattern recognition." in Learning with kernels, 2002 - from V .Vapnik, 1982

Separating hyperplanes

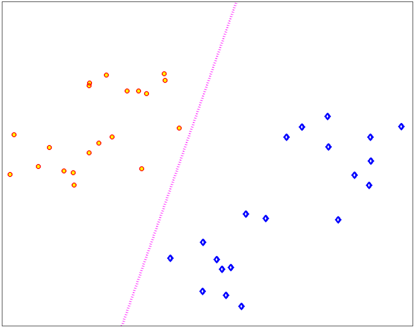
Find a line to separate (classify) blue from red



$$D(x) = \text{sign}\left(\mathbf{v}^\top \mathbf{x} + a\right)$$

Separating hyperplanes

Find a line to separate (classify) blue from red



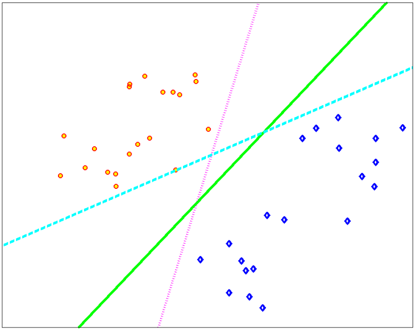
$$D(x) = \text{sign}\big(\mathbf{v}^\top \mathbf{x} + a\big)$$

the decision border:

$$\mathbf{v}^\top \mathbf{x} + a = 0$$

Separating hyperplanes

Find a line to separate (classify) blue from red



$$D(x) = \text{sign}\left(\mathbf{v}^\top \mathbf{x} + a\right)$$

the decision border:

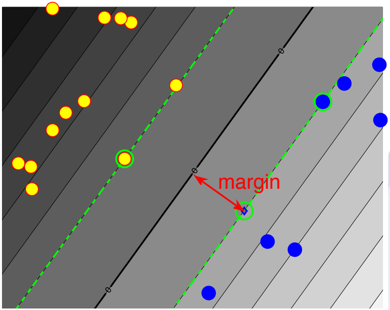
$$\mathbf{v}^\top \mathbf{x} + a = 0$$

there are many solutions...
The problem is ill posed

How to choose a solution?

Maximize our *confidence* = maximize the margin

the decision border: $\Delta(\mathbf{v}, a) = \{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{v}^\top \mathbf{x} + a = 0\}$



maximize the margin

$$\underbrace{\max_{\mathbf{v},a} \min_{i \in [1,n]} \text{dist}(\mathbf{x}_i, \Delta(\mathbf{v}, a))}_{\text{margin: } m}$$

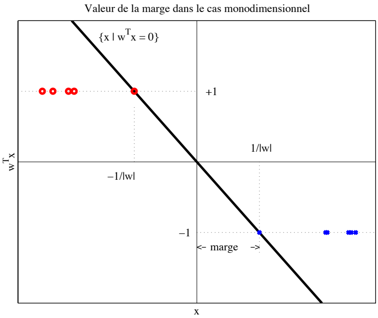
Maximize the confidence

$$\begin{cases} \displaystyle\max_{\mathbf{v},a} & m \\ \text{with} & \displaystyle\min_{i=1,n} \frac{|\mathbf{v}^\top \mathbf{x}_i + a|}{\|\mathbf{v}\|} \geq m \end{cases}$$

the problem is still ill posed

if $(\mathbf{v}, a)$ is a solution, $\forall\ 0 < k\ \ (k\mathbf{v}, ka)$ is also a solution...

From the geometrical to the numerical margin



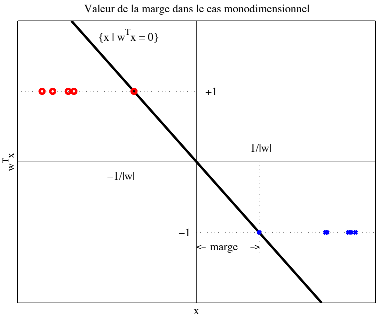
Maximize the (geometrical) margin

$$\begin{cases} \max\limits_{\mathbf{v},a} & m \\ \text{with} & \min\limits_{i=1,n} \dfrac{|\mathbf{v}^\top \mathbf{x}_i + a|}{\|\mathbf{v}\|} \geq m \end{cases}$$

if the min is greater, everybody is greater ($y_i \in \{-1, 1\}$)

$$\begin{cases} \max\limits_{\mathbf{v},a} & m \\ \text{with} & \dfrac{y_i(\mathbf{v}^\top \mathbf{x}_i + a)}{\|\mathbf{v}\|} \geq m, \quad i = 1, n \end{cases}$$

From the geometrical to the numerical margin



Maximize the (geometrical) margin

$$\begin{cases} \max\limits_{\mathbf{v},a} & m \\ \text{with} & \min\limits_{i=1,n} \dfrac{|\mathbf{v}^\top \mathbf{x}_i + a|}{\|\mathbf{v}\|} \geq m \end{cases}$$

if the min is greater, everybody is greater ($y_i \in \{-1, 1\}$)

$$\begin{cases} \max\limits_{\mathbf{v},a} & m \\ \text{with} & \dfrac{y_i(\mathbf{v}^\top \mathbf{x}_i + a)}{\|\mathbf{v}\|} \geq m, \quad i = 1, n \end{cases}$$

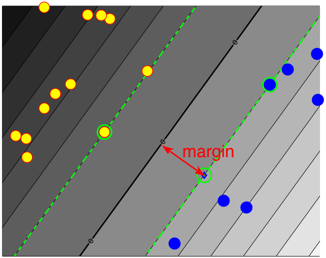
change variable: $\mathbf{w} = \frac{\mathbf{v}}{m\|\mathbf{v}\|}$ and $b = \frac{a}{m\|\mathbf{v}\|} \Longrightarrow \|\mathbf{w}\| = \frac{1}{m}$

$$\begin{cases} \max\limits_{\mathbf{w},b} & m \\ \text{with} & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \quad ; \; i = 1, n \\ \text{and} & m = \frac{1}{\|\mathbf{w}\|} \end{cases}$$

$$\begin{cases} \min\limits_{\mathbf{w},b} & \|\mathbf{w}\|^2 \\ \text{with} & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \\ & i = 1, n \end{cases}$$

Road map

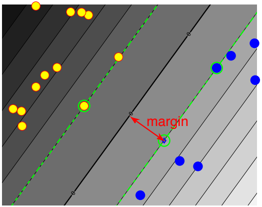
1. Supervised classification and prediction
2. Linear SVM
   - Separating hyperplanes
   - Linear SVM: the problem
   - Optimization in 5 slides
   - Dual formulation of the linear SVM
   - The non separable case
3. Kernels
4. Kernelized support vector machine



"The algorithms for constructing the separating hyperplane considered above will be utilized for developing a battery of programs for pattern recognition." in Learning with kernels, 2002 - from V .Vapnik, 1982

Linear SVM: the problem

The maximal margin (=minimal norm)
canonical hyperplane



Linear SVMs are the solution of the following problem (called primal)

Let $\{(\mathbf{x}_i, y_i);\ i = 1 : n\}$ be a set of labelled data with $\mathbf{x} \in \mathbb{R}^d, y_i \in \{1, -1\}$

A support vector machine (SVM) is a linear classifier associated with the following decision function: $D(x) = \text{sign}\big(\mathbf{w}^\top \mathbf{x} + b\big)$ where $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ a given thought the solution of the following problem:

$$\begin{cases} \min\limits_{\mathbf{w}\in\mathbb{R}^d,\ b\in\mathbb{R}} & \frac{1}{2}\,\|\mathbf{w}\|^2 \\ \text{with} & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1\ , \qquad i = 1, n \end{cases}$$

This is a quadratic program (QP): $\begin{cases} \min\limits_{\mathbf{z}} & \frac{1}{2}\,\mathbf{z}^\top A\mathbf{z} - \mathbf{d}^\top \mathbf{z} \\ \text{with} & B\mathbf{z} \leq \mathbf{e} \end{cases}$

Support vector machines as a QP

The Standart QP formulation

$$\begin{cases} \min\limits_{\mathbf{w},b} & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{with} & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, i = 1, n \end{cases} \Leftrightarrow \begin{cases} \min\limits_{\mathbf{z}\in\mathbb{R}^{d+1}} & \frac{1}{2}\mathbf{z}^\top A\mathbf{z} - \mathbf{d}^\top \mathbf{z} \\ \text{with} & B\mathbf{z} \leq \mathbf{e} \end{cases}$$

$\mathbf{z} = (\mathbf{w}, b)^\top$, $\mathbf{d} = (0, \ldots, 0)^\top$, $A = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$, $B = -[\text{diag}(\mathbf{y})X, \mathbf{y}]$ and $\mathbf{e} = -(1, \ldots, 1)^\top$

Solve it using a standard QP solver such as (for instance)

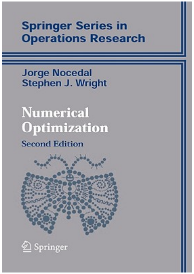
```
% QUADPROG Quadratic programming.
%    X = QUADPROG(H,f,A,b) attempts to solve the quadratic programming problem:
%
%             min 0.5*x'*H*x + f'*x   subject to:  A*x <= b
%              x
%  so that the solution is in the range LB <= X <= UB
```

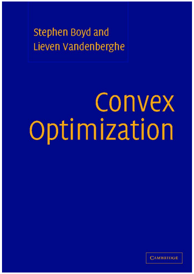
For more solvers (just to name a few) have a look at:

- plato.asu.edu/sub/nlores.html#QP-problem
- www.numerical.rl.ac.uk/people/nimg/qp/qp.html

Road map

1. Supervised classification and prediction
2. Linear SVM
   - Separating hyperplanes
   - Linear SVM: the problem
   - Optimization in 5 slides
   - Dual formulation of the linear SVM
   - The non separable case
3. Kernels
4. Kernelized support vector machine





First order optimality condition (1)

$$\text{problem } \mathcal{P} = \begin{cases} \min\limits_{\mathbf{x}\in\mathbb{R}^n} & J(\mathbf{x}) \\ \text{with} & h_j(x) = 0 \quad j = 1, \dots, p \\ \text{and} & g_i(x) \le 0 \; i = 1, \dots, q \end{cases}$$

Definition: Karush, Kuhn and Tucker (KKT) conditions

stationarity $\nabla J(x^\star) + \sum_{j=1}^{p} \lambda_j \nabla h_j(x^\star) + \sum_{i=1}^{q} \mu_i \nabla g_i(x^\star) = 0$

primal admissibility $h_j(x^\star) = 0 \qquad j = 1, \dots, p$

$g_i(x^\star) \le 0 \qquad i = 1, \dots, q$

dual admissibility $\mu_i \ge 0 \qquad i = 1, \dots, q$

complementarity $\mu_i g_i(x^\star) = 0 \qquad i = 1, \dots, q$

λ_j and μ_i are called the Lagrange multipliers of problem $\mathcal{P}$

First order optimality condition (2)

Theorem (12.1 Nocedal & Wright pp 321)

If a vector $x^\star$ is a stationary point of problem $\mathcal{P}$
Then there exists[a] Lagrange multipliers such that $\left(x^\star, \{\lambda_j\}_{j=1:p}, \{\mu_i\}_{i=1:q}\right)$ fulfill KKT conditions

[a] under some conditions *e.g.* linear independence constraint qualification

If the problem is convex, then a stationary point is the solution of the problem

A quadratic program (QP) is convex when...

$$(QP) \quad \begin{cases} \displaystyle\min_{\mathbf{z}} & \frac{1}{2}\mathbf{z}^\top A\mathbf{z} - \mathbf{d}^\top \mathbf{z} \\ \text{with} & B\mathbf{z} \leq \mathbf{e} \end{cases}$$

...when matrix A is positive definite

KKT condition - Lagrangian (3)

$$\text{problem } \mathcal{P} = \begin{cases} \min\limits_{\mathbf{x}\in\mathbb{R}^n} & J(\mathbf{x}) \\ \text{with} & h_j(x) = 0 \quad j = 1, \dots, p \\ \text{and} & g_i(x) \leq 0 \; i = 1, \dots, q \end{cases}$$

Definition: Lagrangian

The lagrangian of problem $\mathcal{P}$ is the following function:

$$\mathcal{L}(\mathbf{x}, \lambda, \mu) = J(x) + \sum_{j=1}^{p} \lambda_j h_j(x) + \sum_{i=1}^{q} \mu_i g_i(x)$$

The importance of being a lagrangian

- the stationarity condition can be written: $\nabla \mathcal{L}(\mathbf{x}^\star, \lambda, \mu) = 0$
- the lagrangian saddle point $\max\limits_{\lambda,\mu} \min\limits_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \lambda, \mu)$

Primal variables: x and dual variables λ, μ (the Lagrange multipliers)

Duality – definitions (1)

Primal and (Lagrange) dual problems

$$\mathcal{P} = \begin{cases} \min\limits_{\mathbf{x}\in\mathbb{R}^n} & J(\mathbf{x}) \\ \text{with} & h_j(x) = 0 \quad j = 1, p \\ \text{and} & g_i(x) \leq 0 \quad i = 1, q \end{cases} \qquad \mathcal{D} = \begin{cases} \max\limits_{\lambda\in\mathbb{R}^p, \mu\in\mathbb{R}^q} & Q(\lambda, \mu) \\ \text{with} & \mu_j \geq 0 \quad j = 1, q \end{cases}$$

Dual objective function:

$$\begin{aligned} Q(\lambda, \mu) &= \inf_x \quad \mathcal{L}(\mathbf{x}, \lambda, \mu) \\ &= \inf_x \quad J(x) + \sum_{j=1}^{p} \lambda_j h_j(x) + \sum_{i=1}^{q} \mu_i g_i(x) \end{aligned}$$

Wolf dual problem

$$\mathcal{W} = \begin{cases} \max\limits_{\mathbf{x}, \lambda\in\mathbb{R}^p, \mu\in\mathbb{R}^q} & \mathcal{L}(\mathbf{x}, \lambda, \mu) \\ \text{with} & \mu_j \geq 0 \quad j = 1, q \\ \text{and} & \nabla J(x^\star) + \sum_{j=1}^{p} \lambda_j \nabla h_j(x^\star) + \sum_{i=1}^{q} \mu_i \nabla g_i(x^\star) = 0 \end{cases}$$

Duality – theorems (2)

Theorem (12.12, 12.13 and 12.14 Nocedal & Wright pp 346)

If f, g and h are convex and continuously differentiable[a], then the solution of the dual problem is the same as the solution of the primal

[a] under some conditions *e.g.* linear independence constraint qualification

$$\begin{aligned} (\lambda^\star, \mu^\star) &= \text{solution of problem } \mathcal{D} \\ \mathbf{x}^\star &= \arg\min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \lambda^\star, \mu^\star) \end{aligned}$$

$$\begin{aligned} Q(\lambda^\star, \mu^\star) = \arg\min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \lambda^\star, \mu^\star) &= \mathcal{L}(\mathbf{x}^\star, \lambda^\star, \mu^\star) \\ &= J(\mathbf{x}^\star) + \lambda^\star H(\mathbf{x}^\star) + \mu^\star G(\mathbf{x}^\star) = J(\mathbf{x}^\star) \end{aligned}$$

and for any feasible point $\mathbf{x}$

$$Q(\lambda, \mu) \leq J(\mathbf{x}) \qquad \rightarrow \qquad 0 \leq J(\mathbf{x}) - Q(\lambda, \mu)$$

The duality gap is the difference between the primal and dual cost functions

Road map

1. Supervised classification and prediction
2. Linear SVM
   - Separating hyperplanes
   - Linear SVM: the problem
   - Optimization in 5 slides
   - Dual formulation of the linear SVM
   - The non separable case
3. Kernels
4. Kernelized support vector machine

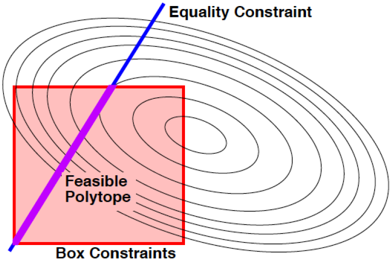


Figure from L. Bottou & C.J. Lin, Support vector machine solvers, in Large scale kernel machines, 2007.

Linear SVM dual formulation - The lagrangian

$$\begin{cases} \min_{\mathbf{w},b} & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{with} & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \qquad i = 1, n \end{cases}$$

Looking for the lagrangian saddle point $\max_{\alpha} \min_{\mathbf{w},b} \mathcal{L}(\mathbf{w}, b, \alpha)$ with so called lagrange multipliers $\alpha_i \geq 0$

$$\mathcal{L}(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \big(y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1\big)$$

α_i represents the influence of constraint thus the influence of the training example (x_i, y_i)

Stationarity conditions

$$\mathcal{L}(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \big(y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1\big)$$

Computing the gradients: $\begin{cases} \nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w}, b, \alpha) & = \mathbf{w} - \displaystyle\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \\ \dfrac{\partial \mathcal{L}(\mathbf{w}, b, \alpha)}{\partial b} & = \sum_{i=1}^{n} \alpha_i \; y_i \end{cases}$

we have the following optimality conditions

$$\begin{cases} \nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w}, b, \alpha) & = 0 \quad \Rightarrow \quad \mathbf{w} = \displaystyle\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \\ \dfrac{\partial \mathcal{L}(\mathbf{w}, b, \alpha)}{\partial b} & = 0 \quad \Rightarrow \quad \displaystyle\sum_{i=1}^{n} \alpha_i \; y_i = 0 \end{cases}$$

KKT conditions for SVM

stationarity	$\mathbf{w} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0$ and $\sum_{i=1}^{n} \alpha_i \; y_i = 0$	
primal admissibility	$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1$	$i = 1, \dots, n$
dual admissibility	$\alpha_i \geq 0$	$i = 1, \dots, n$
complementarity	$\alpha_i \Big(y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1 \Big) = 0$	$i = 1, \dots, n$

The complementary condition split the data into two sets

- $\mathcal{A}$ be the set of active constraints: usefull points

$$\mathcal{A} = \{ i \in [1, n] \;\big|\; y_i(\mathbf{w}^{*\top} \mathbf{x}_i + b^*) = 1 \}$$

- its complementary $\bar{\mathcal{A}}$ useless points

$$\text{if } i \notin \mathcal{A}, \alpha_i = 0$$

The KKT conditions for SVM

The same KKT but using matrix notations and the active set $\mathcal{A}$

stationarity	$\mathbf{w} - X^\top D_y \alpha = 0$
	$\alpha^\top y = 0$
primal admissibility	$D_y(Xw + b\mathbb{I}) \geq \mathbb{I}$
dual admissibility	$\alpha \geq 0$
complementarity	$D_y(X_{\mathcal{A}}\mathbf{w} + b\mathbb{I}_{\mathcal{A}}) = \mathbb{I}_{\mathcal{A}}$
	$\alpha_{\bar{\mathcal{A}}} = 0$

Knowing $\mathcal{A}$, the solution verifies the following linear system:

$$\begin{cases} \mathbf{w} & -X_{\mathcal{A}}^\top D_y \alpha_{\mathcal{A}} & & = 0 \\ -D_y X_{\mathcal{A}}\mathbf{w} & & -b\mathbf{y}_{\mathcal{A}} & = -\mathbf{e}_{\mathcal{A}} \\ & -\mathbf{y}_{\mathcal{A}}^\top \alpha_{\mathcal{A}} & & = 0 \end{cases}$$

with $D_y = \text{diag}(\mathbf{y}_{\mathcal{A}})$, $\alpha_{\mathcal{A}} = \alpha(\mathcal{A})$, $\mathbf{y}_{\mathcal{A}} = \mathbf{y}(\mathcal{A})$ et $X_{\mathcal{A}} = X(X_{\mathcal{A}};:)$.

The KKT conditions as a linear system

$$\begin{cases} \mathbf{w} & -X_{\mathcal{A}}^\top D_y \alpha_{\mathcal{A}} & & = 0 \\ -D_y X_{\mathcal{A}} \mathbf{w} & & -b\mathbf{y}_{\mathcal{A}} & = -\mathbf{e}_{\mathcal{A}} \\ & -\mathbf{y}_{\mathcal{A}}^\top \alpha_{\mathcal{A}} & & = 0 \end{cases}$$

with $D_y = \text{diag}(\mathbf{y}_{\mathcal{A}})$, $\alpha_{\mathcal{A}} = \alpha(\mathcal{A})$, $\mathbf{y}_{\mathcal{A}} = \mathbf{y}(\mathcal{A})$ et $X_{\mathcal{A}} = X(X_{\mathcal{A}};:)$.

$$\begin{bmatrix} I & -X_{\mathcal{A}}^\top D_y & 0 \\ -D_y X_{\mathcal{A}} & 0 & -\mathbf{y}_{\mathcal{A}} \\ 0 & -\mathbf{y}_{\mathcal{A}}^\top & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \alpha_{\mathcal{A}} \\ b \end{bmatrix} = \begin{bmatrix} 0 \\ -\mathbf{e}_{\mathcal{A}} \\ 0 \end{bmatrix}$$

we can work on it to separate $\mathbf{w}$ from $(\alpha_{\mathcal{A}}, b)$

The SVM dual formulation

The SVM Wolfe dual

$$\left\{\begin{array}{lll} \displaystyle\max_{\mathbf{w},b,\alpha} & \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \big(y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1\big) & \\ \text{with} & \alpha_i \geq 0 & i = 1, \dots, n \\ \text{and} & \mathbf{w} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0 \text{ and } \sum_{i=1}^{n} \alpha_i \; y_i = 0 & \end{array}\right.$$

using the fact: $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$

The SVM Wolfe dual without $\mathbf{w}$ and b

$$\left\{\begin{array}{lll} \displaystyle\max_{\alpha} & -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_j \alpha_i y_i y_j \mathbf{x}_j^\top \mathbf{x}_i + \sum_{i=1}^{n} \alpha_i & \\ \text{with} & \alpha_i \geq 0 & i = 1, \dots, n \\ \text{and} & \sum_{i=1}^{n} \alpha_i \; y_i = 0 & \end{array}\right.$$

Linear SVM dual formulation

$$\mathcal{L}(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \big(y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1\big)$$

Optimality: $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \qquad \sum_{i=1}^{n} \alpha_i \; y_i = 0$

$$\begin{aligned}
\mathcal{L}(\alpha) &= \tfrac{1}{2} \underbrace{\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_j \alpha_i y_i y_j \mathbf{x}_j^\top \mathbf{x}_i}_{\mathbf{w}^\top \mathbf{w}} - \textstyle\sum_{i=1}^{n} \alpha_i y_i \underbrace{\sum_{j=1}^{n} \alpha_j y_j \mathbf{x}_j^\top}_{\mathbf{w}^\top} \mathbf{x}_i - b \underbrace{\sum_{i=1}^{n} \alpha_i y_i}_{=0} + \textstyle\sum_{i=1}^{n} \alpha_i \\
&= -\frac{1}{2} \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_j \alpha_i y_i y_j \mathbf{x}_j^\top \mathbf{x}_i + \sum_{i=1}^{n} \alpha_i
\end{aligned}$$

Dual linear SVM is also a quadratic program

$$\text{problem } \mathcal{D} \quad \begin{cases} \min\limits_{\alpha \in \mathbb{R}^n} & \frac{1}{2}\alpha^\top G \alpha - \mathbf{e}^\top \alpha \\ \text{with} & \mathbf{y}^\top \alpha = 0 \\ \text{and} & 0 \le \alpha_i \qquad\qquad i = 1, n \end{cases}$$

with G a symmetric matrix $n \times n$ such that $G_{ij} = y_i y_j \mathbf{x}_j^\top \mathbf{x}_i$

SVM primal vs. dual

Primal

$$\begin{cases} \min_{\mathbf{w}\in\mathbb{R}^d, b\in\mathbb{R}} & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{with} & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \\ & i = 1, n \end{cases}$$

- $d + 1$ unknown
- n constraints
- classical QP
- perfect when $d << n$

Dual

$$\begin{cases} \min_{\alpha\in\mathbb{R}^n} & \frac{1}{2}\alpha^\top G\alpha - \mathbf{e}^\top\alpha \\ \text{with} & \mathbf{y}^\top\alpha = 0 \\ \text{and} & 0 \leq \alpha_i \qquad i = 1, n \end{cases}$$

- n unknown
- G Gram matrix (pairwise influence matrix)
- n box constraints
- easy to solve
- to be used when $d > n$

SVM primal vs. dual

Primal

$$\begin{cases} \min\limits_{\mathbf{w}\in\mathbb{R}^d, b\in\mathbb{R}} & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{with} & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \\ & i = 1, n \end{cases}$$

- $d + 1$ unknown
- n constraints
- classical QP
- perfect when $d << n$

Dual

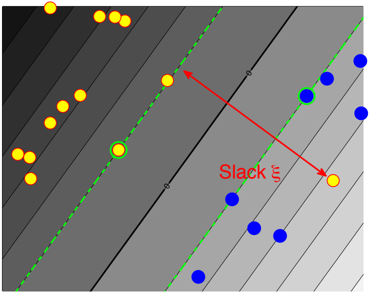
$$\begin{cases} \min\limits_{\alpha\in\mathbb{R}^n} & \frac{1}{2}\alpha^\top G\alpha - \mathbf{e}^\top\alpha \\ \text{with} & \mathbf{y}^\top\alpha = 0 \\ \text{and} & 0 \leq \alpha_i \qquad i = 1, n \end{cases}$$

- n unknown
- G Gram matrix (pairwise influence matrix)
- n box constraints
- easy to solve
- to be used when $d > n$

$$f(\mathbf{x}) = \sum_{j=1}^{d} w_j x_j + b = \sum_{i=1}^{n} \alpha_i\, y_i(\mathbf{x}^\top \mathbf{x}_i) + b$$

Road map

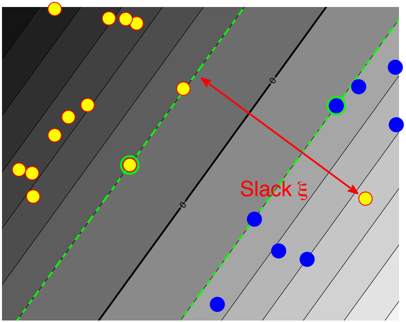
1. Supervised classification and prediction
2. Linear SVM
   - Separating hyperplanes
   - Linear SVM: the problem
   - Optimization in 5 slides
   - Dual formulation of the linear SVM
   - The non separable case
3. Kernels
4. Kernelized support vector machine



The non separable case: a bi criteria optimization problem

Modeling potential errors: introducing slack variables ξ_i

$$(x_i, y_i) \qquad \begin{cases} \text{no error:} & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \Rightarrow \quad \xi_i = 0 \\ \text{error:} & \qquad\qquad\qquad\qquad\quad \xi_i = 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b) > 0 \end{cases}$$



$$\begin{cases} \min\limits_{\mathbf{w},b,\xi} & \frac{1}{2}\|\mathbf{w}\|^2 \\ \min\limits_{\mathbf{w},b,\xi} & \frac{C}{p}\sum_{i=1}^{n} \xi_i^p \\ \text{with} & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \quad i = 1, n \end{cases}$$

Our hope: almost all $\xi_i = 0$

The non separable case

Modeling potential errors: introducing slack variables ξ_i

$$(x_i, y_i) \qquad \begin{cases} \text{no error:} & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \Rightarrow & \xi_i = 0 \\ \text{error:} & & \xi_i = 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b) > 0 \end{cases}$$

Minimizing also the slack (the error), for a given $C > 0$

$$\begin{cases} \min\limits_{\mathbf{w},b,\xi} & \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{p}\sum_{i=1}^n \xi_i^p \\ \text{with} & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i \quad & i = 1, n \\ & \xi_i \geq 0 & i = 1, n \end{cases}$$

Looking for the saddle point of the lagrangian with the Lagrange multipliers $\alpha_i \geq 0$ and $\beta_i \geq 0$

$$\mathcal{L}(\mathbf{w}, b, \alpha, \beta) = \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{p}\sum_{i=1}^n \xi_i^p - \sum_{i=1}^n \alpha_i\big(y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1 + \xi_i\big) - \sum_{i=1}^n \beta_i \xi_i$$

The KKT

$$\mathcal{L}(\mathbf{w}, b, \alpha, \beta) = \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{p}\sum_{i=1}^{n}\xi_i^p - \sum_{i=1}^{n}\alpha_i\big(y_i(\mathbf{w}^\top\mathbf{x}_i + b) - 1 + \xi_i\big) - \sum_{i=1}^{n}\beta_i\xi_i$$

stationarity	$\mathbf{w} - \sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i = 0$ and $\sum_{i=1}^{n}\alpha_i\ y_i = 0$	
	$C - \alpha_i - \beta_i = 0$	$i = 1, \dots, n$
primal admissibility	$y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1$	$i = 1, \dots, n$
	$\xi_i \geq 0$	$i = 1, \dots, n$
dual admissibility	$\alpha_i \geq 0$	$i = 1, \dots, n$
	$\beta_i \geq 0$	$i = 1, \dots, n$
complementarity	$\alpha_i\Big(y_i(\mathbf{w}^\top\mathbf{x}_i + b) - 1 + \xi_i\Big) = 0$	$i = 1, \dots, n$
	$\beta_i\xi_i = 0$	$i = 1, \dots, n$

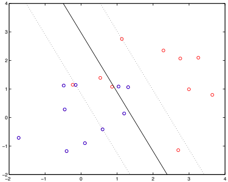
Let's eliminate β!

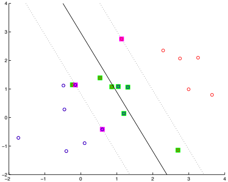
KKT

stationarity	$\mathbf{w} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0$	and	$\sum_{i=1}^{n} \alpha_i \; y_i = 0$
primal admissibility	$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1$		$i = 1, \dots, n$
	$\xi_i \geq 0$		$i = 1, \dots, n;$
dual admissibility	$\alpha_i \geq 0$		$i = 1, \dots, n$
	$C - \alpha_i \geq 0$		$i = 1, \dots, n;$
complementarity	$\alpha_i \Big(y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1 + \xi_i \Big) = 0$		$i = 1, \dots, n$
	$(C - \alpha_i)\; \xi_i = 0$		$i = 1, \dots, n$

sets	I_0	$I_{\mathcal{A}}$	I_C
α_i	0	$0 < \alpha < C$	C
β_i	C	$C - \alpha$	0
ξ_i	0	0	$1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)$
	$y_i(\mathbf{w}^\top \mathbf{x}_i + b) > 1$	$y_i(\mathbf{w}^\top \mathbf{x}_i + b) = 1$	$y_i(\mathbf{w}^\top \mathbf{x}_i + b) < 1$
	useless	usefull (support vec)	suspicious

The importance of being support





data point	α	constraint value	set
$\mathbf{x}_i$ *useless*	$\alpha_i = 0$	$y_i\big(\mathbf{w}^\top \mathbf{x}_i + b\big) > 1$	I_0
$\mathbf{x}_i$ *support*	$0 < \alpha_i < C$	$y_i\big(\mathbf{w}^\top \mathbf{x}_i + b\big) = 1$	I_α
$\mathbf{x}_i$ *suspicious*	$\alpha_i = C$	$y_i\big(\mathbf{w}^\top \mathbf{x}_i + b\big) < 1$	I_C

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

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

sparsity: $\alpha_i = 0$

Optimality conditions ($p = 1$)

$$\mathcal{L}(\mathbf{w}, b, \alpha, \beta) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i\big(y_i(\mathbf{w}^\top\mathbf{x}_i + b) - 1 + \xi_i\big) - \sum_{i=1}^{n}\beta_i\xi_i$$

Computing the gradients:
$$\begin{cases} \nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}, b, \alpha) & = \mathbf{w} - \sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i \\ \dfrac{\partial \mathcal{L}(\mathbf{w}, b, \alpha)}{\partial b} & = \sum_{i=1}^{n}\alpha_i \, y_i \\ \nabla_{\xi_i}\mathcal{L}(\mathbf{w}, b, \alpha) & = C - \alpha_i - \beta_i \end{cases}$$

- no change for $\mathbf{w}$ and b
- $\beta_i \geq 0$ and $C - \alpha_i - \beta_i = 0 \quad \Rightarrow \quad \alpha_i \leq C$

The dual formulation:

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

SVM primal vs. dual

Primal

$$\begin{cases} \min\limits_{\mathbf{w},b,\xi\in\mathbb{R}^n} & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum\limits_{i=1}^{n}\xi_i \\ \text{with} & y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \quad i = 1, n \end{cases}$$

- $d + n + 1$ unknown
- $2n$ constraints
- classical QP
- to be used when n is too large to build G

Dual

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

- n unknown
- G Gram matrix (pairwise influence matrix)
- $2n$ box constraints
- easy to solve
- to be used when n is not too large

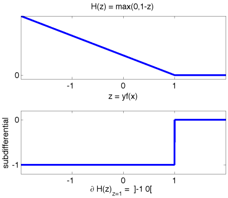
Eliminating the slack but not the possible mistakes

$$\begin{cases} \min\limits_{\mathbf{w},b,\xi\in\mathbb{R}^n} & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \\ \text{with} & y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \quad i = 1, n \end{cases}$$

Introducing the hinge loss

$$\xi_i = \max\big(1 - y_i(\mathbf{w}^\top\mathbf{x}_i + b), 0\big)$$

$$\min_{\mathbf{w},b} \frac{1}{2} \;\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\max\big(0, 1 - y_i(\mathbf{w}^\top\mathbf{x}_i + b)\big)$$



Back to $d+1$ variables, but this is no longer an explicit QP

The hinge and other loss

Square hinge: (huber/hinge) and Lasso SVM

$$\min_{\mathbf{w},b} \quad \|\mathbf{w}\|_1 + \; C \sum_{i=1}^{n} \max\big(1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b), 0\big)^p$$

Penalized Logistic regression (Maxent)

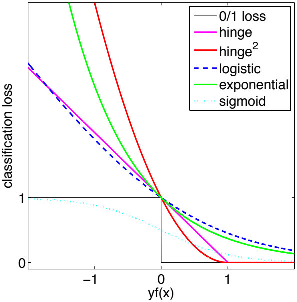
$$\min_{\mathbf{w},b} \quad \|\mathbf{w}\|_2^2 - \; C \sum_{i=1}^{n} \log\big(1 + \exp^{-2y_i(\mathbf{w}^\top \mathbf{x}_i + b)}\big)$$

The exponential loss (commonly used in boosting)

$$\min_{\mathbf{w},b} \quad \|\mathbf{w}\|_2^2 + \; C \sum_{i=1}^{n} \exp^{-y_i(\mathbf{w}^\top \mathbf{x}_i + b)}$$

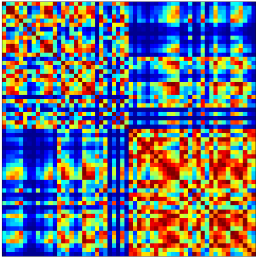
The sigmoid loss

$$\min_{\mathbf{w},b} \quad \|\mathbf{w}\|_2^2 - \; C \sum_{i=1}^{n} \tanh\big(y_i(\mathbf{w}^\top \mathbf{x}_i + b)\big)$$



Roadmap

1. Supervised classification and prediction
2. Linear SVM
   - Separating hyperplanes
   - Linear SVM: the problem
   - Optimization in 5 slides
   - Dual formulation of the linear SVM
   - The non separable case
3. Kernels
4. Kernelized support vector machine



Introducing non linearities through the feature map

SVM Val

$$f(\mathbf{x}) \quad = \quad \sum_{j=1}^{d} x_j w_j + b \quad = \quad \sum_{i=1}^{n} \alpha_i (\mathbf{x}_i^\top \mathbf{x}) + b$$

$\begin{pmatrix} t_1 \\ t_2 \end{pmatrix} \in \mathbb{R}^2$

	x_1
	x_2
	x_3
	x_4
	x_5

linear in $\mathbf{x} \in \mathbb{R}^5$

Introducing non linearities through the feature map

SVM Val

$$f(\mathbf{x}) \quad = \quad \sum_{j=1}^{d} x_j w_j + b \quad = \quad \sum_{i=1}^{n} \alpha_i (\mathbf{x}_i^\top \mathbf{x}) + b$$

$\begin{pmatrix} t_1 \\ t_2 \end{pmatrix} \in \mathbb{R}^2$

$$\phi(t) = \begin{array}{|c|c|} \hline t_1 & x_1 \\ t_1^2 & x_2 \\ t_2 & x_3 \\ t_2^2 & x_4 \\ t_1 t_2 & x_5 \\ \hline \end{array}$$

linear in $\mathbf{x} \in \mathbb{R}^5$
quadratic in $\mathrm{t} \in \mathbb{R}^2$

The feature map

$$\begin{array}{rccc} \phi: & \mathbb{R}^2 & \longrightarrow & \mathbb{R}^5 \\ & \mathrm{t} & \longmapsto & \phi(\mathrm{t}) = \mathbf{x} \end{array}$$

$$\mathbf{x}_i^\top \mathbf{x} = \phi(\mathrm{t}_i)^\top \phi(\mathrm{t})$$

Introducing non linearities through the feature map

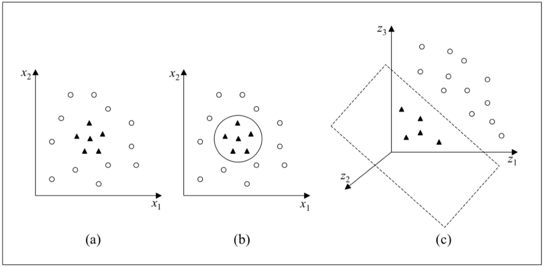


Figura 8. (a) Conjunto de dados não linear; (b) Fronteira não linear no espaço de entradas; (c) Fronteira linear no espaço de características [28]

A. Lorena & A. de Carvalho, Uma Introducão às Support Vector Machines, 2007

Non linear case: dictionary *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...

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

so that

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

Non linear case: dictionary *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...

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

so that

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

$p \geq n$ 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 $\mathbb{R}^d$:

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

in this case

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

closed form kernel: the quadratic kernel

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

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

in this case

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

The quadratic kenrel: $\begin{array}{rl} \mathbf{s}, \mathrm{t} \in \mathbb{R}^d, \quad k(\mathbf{s}, \mathrm{t}) & = \big(\mathbf{s}^\top \mathrm{t} + 1\big)^2 \\ & = 1 + 2\mathbf{s}^\top \mathrm{t} + \big(\mathbf{s}^\top \mathrm{t}\big)^2 \end{array}$ computes the dot product of the reweighted dictionary:

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

closed form kernel: the quadratic kernel

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

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

in this case

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

The quadratic kenrel: $\mathbf{s}, \mathrm{t} \in \mathbb{R}^d, \quad k(\mathbf{s}, \mathrm{t}) = \big(\mathbf{s}^\top \mathrm{t} + 1\big)^2 = 1 + 2\mathbf{s}^\top \mathrm{t} + \big(\mathbf{s}^\top \mathrm{t}\big)^2$ computes the dot product of the reweighted dictionary:

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

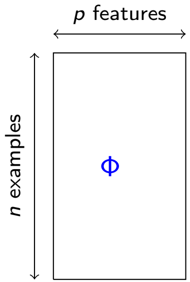
$p = 1 + d + \frac{d(d+1)}{2}$ multiplications *vs.* $d + 1$

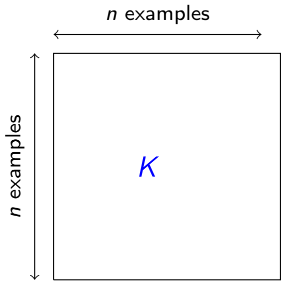
use kernel to save computration

kernel: features through pairwise comparisons

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

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





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

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

Kenrel machine

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)

Let Ω be a non empty set (the input space).

A *kernel* is a function k from $\Omega \times \Omega$ onto $\mathbb{R}$.

$$k: \begin{array}{ccc} \Omega \times \Omega & \longmapsto & \mathbb{R} \\ \mathbf{s}, \mathrm{t} & \longrightarrow & k(\mathbf{s}, \mathrm{t}) \end{array}$$

Kenrel machine

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)

Let Ω be a non empty set (the input space).

A *kernel* is a function k from $\Omega \times \Omega$ onto $\mathbb{R}$.

$$k: \begin{array}{ccc} \Omega \times \Omega & \longmapsto & \mathbb{R} \\ \mathbf{s}, \mathrm{t} & \longrightarrow & k(\mathbf{s}, \mathrm{t}) \end{array}$$

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

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

In the beginning was the kernel...

Definition (Kernel)

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

Definition (Positive kernel)

A kernel $k(s,t)$ on Ω is said to be positive

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

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

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

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

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: $k(\mathbf{s}, \mathrm{t}) = g(\mathbf{s}) g(\mathrm{t})$ 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) $\Leftrightarrow$ $k(\mathbf{s}, \mathrm{t}) = \langle \phi_{\mathbf{s}}, \phi_{\mathrm{t}} \rangle$

Positive definite Kernel (PDK) algebra (closure)

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

- DPK are a convex cone: $\forall a_1 \in \mathbb{R}^+ \quad a_1 k_1(\mathbf{s}, \mathrm{t}) + k_2(\mathbf{s}, \mathrm{t})$
- product kernel $k_1(\mathbf{s}, \mathrm{t}) k_2(\mathbf{s}, \mathrm{t})$

proofs

- by linearity:

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

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

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

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 $\mathbb{R}$

$$\phi\big(k(\mathbf{s},\mathrm{t})\big)$$

- if $\Psi$is a function from $\mathbb{R}^d$ to $\mathbb{R}^d$

$$k\big(\Psi(\mathbf{s}),\Psi(\mathrm{t})\big)$$

- if φ from $\mathbb{R}^d$ to $\mathbb{R}^+$, is minimum in 0

$$k(\mathbf{s},\mathrm{t}) = \varphi(\mathbf{s}+\mathrm{t}) - \varphi(\mathbf{s}-\mathrm{t})$$

- convolution of two positive kernels is a positive kernel

$$K_1 \star K_2$$

Example : the Gaussian kernel is a PDK

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

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

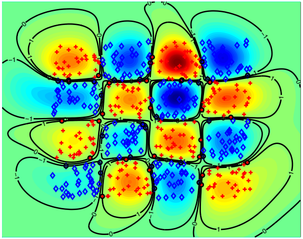
some examples of PD kernels...

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

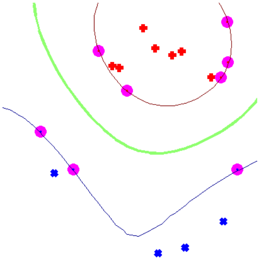
Roadmap

1. Supervised classification and prediction
2. Linear SVM
   - Separating hyperplanes
   - Linear SVM: the problem
   - Optimization in 5 slides
   - Dual formulation of the linear SVM
   - The non separable case
3. Kernels
4. Kernelized support vector machine

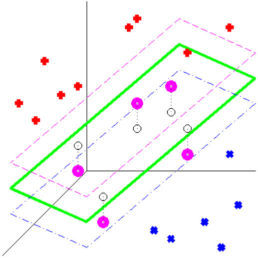


using relevant features...

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



input space representation: x



feature space: k(x,.)

Representer theorem for SVM

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

Lagrangian

$$L(f,b,\alpha) = \frac{1}{2}\|f\|^2_{\mathcal{H}} - \sum_{i=1}^{n} \alpha_i \big(y_i(f(\mathbf{x}_i)+b) - 1\big) \qquad \alpha \geq 0$$

optimility condition: $\nabla_f L(f,b,\alpha) = 0 \Leftrightarrow f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x})$

Eliminate f from L: $\begin{cases} \|f\|^2_{\mathcal{H}} = \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j k(\mathbf{x}_i,\mathbf{x}_j) \\ \sum_{i=1}^{n} \alpha_i y_i f(\mathbf{x}_i) = \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j k(\mathbf{x}_i,\mathbf{x}_j) \end{cases}$

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

Dual formulation for SVM

the intermediate function

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

$$\max_{\alpha}\min_{b} \; Q(b,\alpha)$$

b can be seen as the Lagrange multiplier of the following (balanced) constaint $\sum_{i=1}^{n}\alpha_i y_i = 0$ which is also the optimality KKT condition on b

Dual formulation

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

SVM dual formulation

Dual formulation

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

The dual formulation gives a quadratic program (QP)

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

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

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

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

$p = 1$, L_1 SVM

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

$p = 2$, L_2 SVM

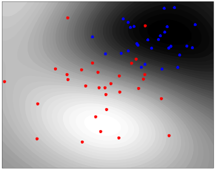
$$\begin{cases} \max\limits_{\alpha \in \mathbb{R}^n} & -\frac{1}{2}\alpha^\top \left(G + \frac{1}{C}I\right)\alpha + \alpha^\top \mathbb{1} \\ \text{such that} & \alpha^\top \mathbf{y} = 0 \text{ and } 0 \leq \alpha_i \quad i = 1, n \end{cases}$$

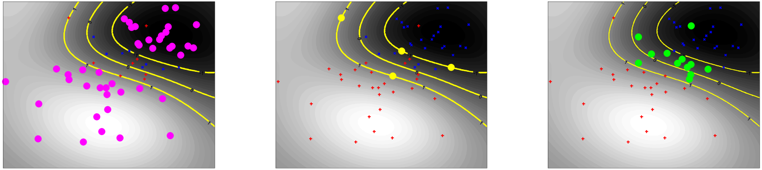
the regularization path: is the set of solutions $\alpha(C)$ when C varies

Data groups: illustration

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

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





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

the regularization path: is the set of solutions $\alpha(C)$ 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	set
$\mathbf{x}_i$ *useless*	$\alpha_i = 0$	$y_i\big(f(\mathbf{x}_i) + b\big) > 1$	I_0
$\mathbf{x}_i$ *support*	$0 < \alpha_i < C$	$y_i\big(f(\mathbf{x}_i) + b\big) = 1$	I_α
$\mathbf{x}_i$ *suspicious*	$\alpha_i = C$	$y_i\big(f(\mathbf{x}_i) + b\big) < 1$	I_C

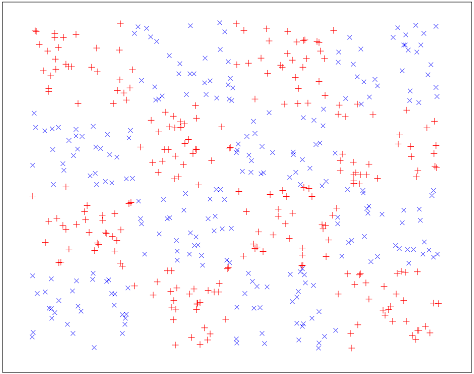
Table : When a data point is « support » it lies exactly on the margin.

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

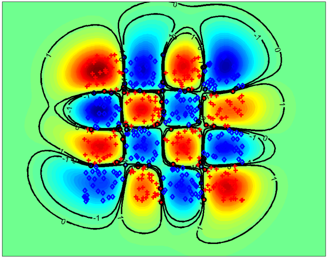
sparsity: $\alpha_i = 0$

checker board

- 2 classes
- 500 examples
- separable

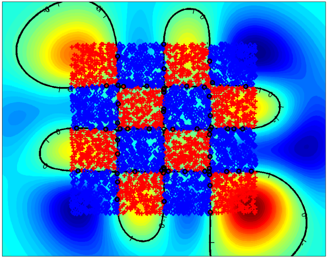


a separable case

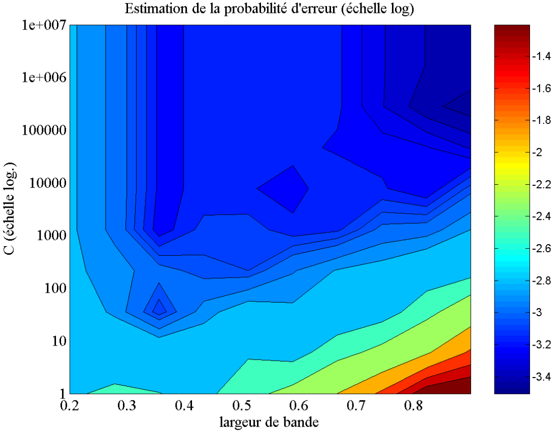


$n = 500$ data points

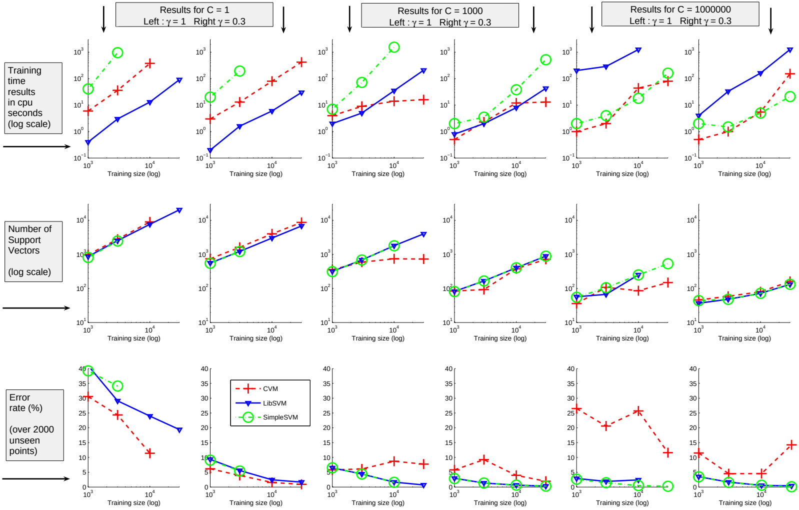
$n = 5000$ data points



Tuning C and γ (the kernel width) : *grid search*



Empirical complexity



G. Loosli *et al* JMLR, 2007

Conclusion

- Learning as an optimization problem
  - use CVX to prototype
  - MonQP
  - specific parallel and distributed solvers
- Universal through Kernelization (dual trick)
- Scalability
  - Sparsity provides scalability
  - Kernel implies "locality"
  - Big data limitations: back to primal (an linear)