Optimal Linear Classifiers: Support Vector Machines (SVM)

Nicolás Pérez de la Blanca
DECSAI-UGR

Background

- Linears Models:
  - Powerful when NLT are used
  - VC dimension increase very fast with NLT
  - Difficult of tune with high order transformations

- Neural networks (FeedForward)
  - Very high expressive power
  - Prone to overfitting
  - High computation time (training)

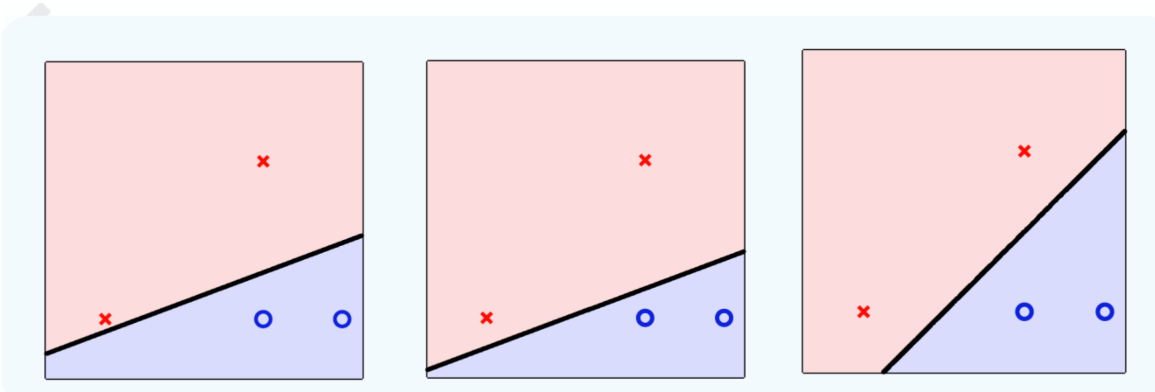
- Can we get the expressive power without paying the price ?

- Yes !! The Support Vector Machine does it
- How ? Implementing the SRM criteria as:

Keep the empirical error constant (small) y minimize the VC dimension

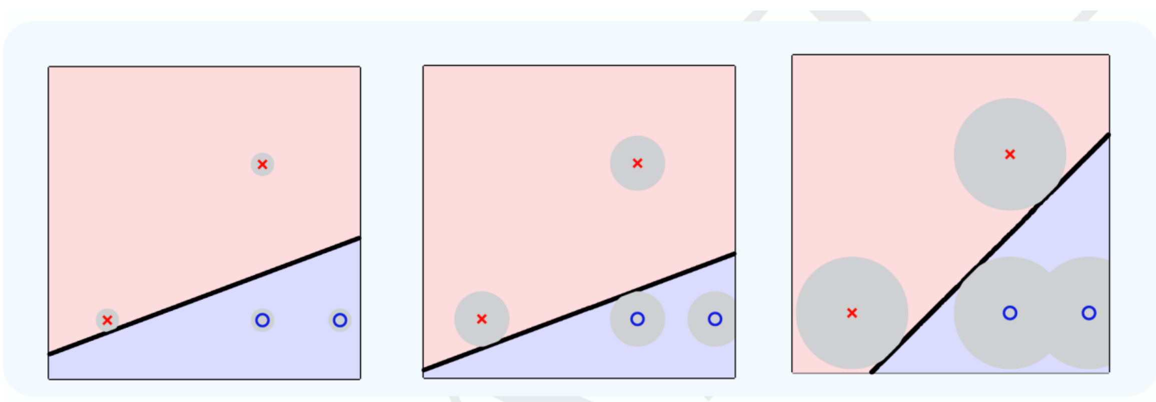
The Optimal Hyperplane

- Let us revisit the perceptron model



Three possible separating hyperplanes, all three with E_{in}=0 and the same VC- dimension (vector dimensionality)

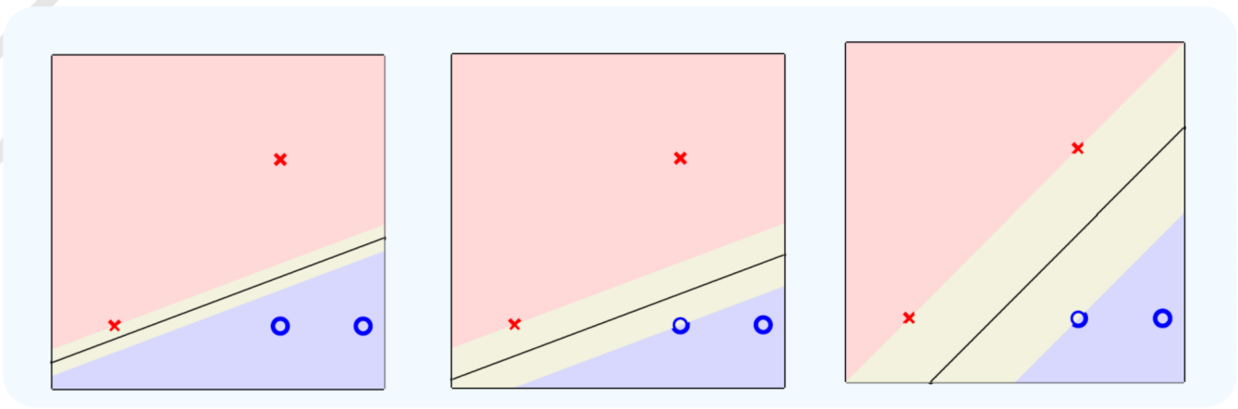
Accoding to the VC-theory all have the same generalization bound



Question: Could we order our preferences?

Let's use noise tolerance !

Fat separator



The robustness to the noise can be quantified by a fat separator.

The thickness reflect the amount of noise the separator can tolerate.

Margin: the maximun thickness possible for a separator

Three important questions:

1. Can we efficiently find the fattest separator? (Algorithm)
2. Why is a fat separator better than a thin one ? (E_{out})
3. What should we do if the data is not separable? (E_{in})

Finding the Fattest Separating Hyperplane

- Notation: Now go back to denote an hyperplane in R^d as: $\mathbf{w}^T\mathbf{x} + b = 0$
  - **w** and b are estimated in different way.
  - $h(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b, \ \mathbf{w} \in R^d, \mathbf{x} \in R^d, b \in R$

- The hyperplane (**w,**b) separates the data if y only if for i=1,…,N

$$y_n(\mathbf{w}^T\mathbf{x}_n + b) > 0$$

  - The magnitud depends on a free scale that must be normalized

$$\rho = \min_{n=1,\ldots,N} y_n(\mathbf{w}^T\mathbf{x}_n + b)$$

$$\min_{n=1,\ldots,N} y_n\left(\frac{w^T}{\rho}x_n + \frac{b}{\rho}\right) = \frac{1}{\rho}\min_{n=1,\ldots,N} y_n(\mathbf{w}^T\mathbf{x}_n + b) = \frac{\rho}{\rho} = 1$$

- Thus, for any separating hyperplane , it is always possible to choose weights so that all the signals $y_n(\mathbf{w}^T\mathbf{x}_n + b)$ are of magnitud greater or equal to 1, with equality satisfied by at least one $(\mathbf{x}_n, y_n)$

Finding the Fattest Separating Hyperplane

- **Definition** (Separating Hyperplane): The hyperplane *h* separates the data if y only if it can be represented by weights (**w**,*b*) that satisfy $\min_{n=1,\ldots,N} y_n(\mathbf{w}^T\mathbf{x}_n + b) = 1$

- Margin of a hyperplane:
  - Consider the distance from a point to a hyperplane (**w**,*b*):

$$d(\mathbf{x}_n, h) = \frac{|\mathbf{w}^T\mathbf{x}_n + b|}{||\mathbf{w}||} = \frac{y_n(\mathbf{w}^T\mathbf{x}_n + b)}{||\mathbf{w}||}$$

  - Then, we can compute the distance from the separating hyperplane to the nearest point

$$\min_{n=1,\ldots,N} \mathrm{dist}(\mathbf{x}_n, h) = \frac{1}{||\mathbf{w}||} \cdot \min_{n=1,\ldots,N} y_n(\mathbf{w}^T\mathbf{x}_n + b) = \frac{1}{||\mathbf{w}||}$$

- This simple expression for the distance of the the nearest point to the hyperplane is a consequence of above normalization

- Clearly, to maximize $\frac{1}{||\mathbf{w}||}$ is equivalent to minimize $\mathbf{w}^{\mathrm{T}}\mathbf{w}$

Finding the Optimal Hyperplane

To estimate the maximum-margin separating hyperplane we need to solve the following optimzation problem:

$$\underset{b,\mathbf{w}}{\text{minimize}} \;\; \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

Subject to: $\min_{n} y_n\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_{\mathrm{n}} + b\right) = 1, \quad (\mathrm{n} = 1, \dots, N)$

To make the optimization problem easier to solve, we approach

PRIMAL

$$\underset{b,\mathbf{w}}{\text{minimize}} \;\; \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

Subject to: $y_n\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_{\mathrm{n}} + b\right) \geq 1, \quad (\mathrm{n} = 1, \dots, N)$

It can be proved that the solution verify the constraints in equality

This problem can be written as:

$$\underset{\mathrm{u}\in\mathbb{R}^{L}}{\text{minimize}} \;\; \frac{1}{2}\mathbf{u}^{\mathrm{T}}\mathrm{Q}\mathbf{u} + \mathbf{p}^{\mathbf{T}}\mathbf{u}$$

Subject to: $\mathbf{Au} \geq \mathbf{c}$

$$\mathrm{Q} = \begin{bmatrix} 0 & \mathbf{0}_d^T \\ \mathbf{0}_d & \mathrm{I}_d \end{bmatrix}, \qquad \mathbf{p} = \mathbf{0}_{d+1}$$

$$[y_n \;\; y_n\mathbf{x}_n^T]\mathbf{u} \geq \mathbf{1}$$

Linear Hard-Margin SVM with QP

Linear Hard-Margin SVM with QP

1: Let $\mathbf{p} = \mathbf{0}_{d+1}$ ($(d+1)$-dimensional zero vector) and $\mathbf{c} = \mathbf{1}_N$ (N-dimensional vector of ones). Construct matrices Q and A, where

$$\mathrm{Q} = \begin{bmatrix} 0 & \mathbf{0}_d^{\mathrm{T}} \\ \mathbf{0}_d & \mathrm{I}_d \end{bmatrix}, \qquad \mathrm{A} = \underbrace{\begin{bmatrix} y_1 & -y_1\mathbf{x}_1^{\mathrm{T}}- \\ \vdots & \vdots \\ y_N & -y_N\mathbf{x}_N^{\mathrm{T}}- \end{bmatrix}}_{\text{signed data matrix}}.$$

2: Calculate $\begin{bmatrix} b^* \\ \mathbf{w}^* \end{bmatrix} = \mathbf{u}^* \leftarrow \mathrm{QP}(\mathrm{Q}, \mathbf{p}, \mathrm{A}, \mathbf{c})$.

3: Return the hypothesis $g(\mathbf{x}) = \mathrm{sign}({\mathbf{w}^*}^{\mathrm{T}}\mathbf{x} + b^*)$.

- The solution depends only on a few data points, called “support vectors”.
- These points are those that verify the restriction in equality

Is a Fat Separator Better ?

- The SVM (optimal hyperplane) is a linear model, therefore $d_{\text{VC}} = \text{d} + 1$.

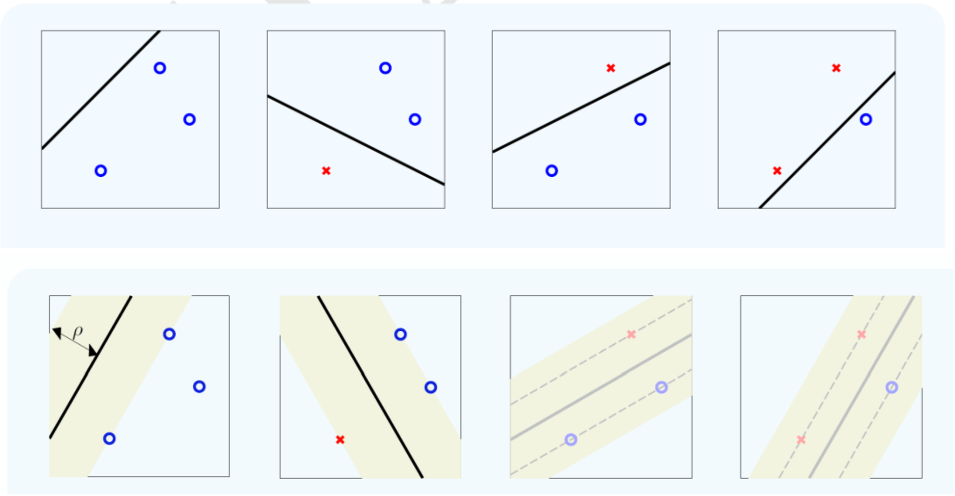
- The question is : does the support vector machine gain any more generalization ability by maximizing the margin ?

- Compare regularization with the optimal hyperplane problem

	optimal hyperplane	regularization
minimize:	$\mathbf{w}^{\mathrm{T}}\mathbf{w}$	E_{in}
subject to:	$E_{\text{in}} = 0$	$\mathbf{w}^{\mathrm{T}}\mathbf{w} \leq C$

  - In SVM the error is fixed (to zero) and minimize the complexity of the model.
  - There exist possibilities of improving in terms of VC dimension

Fat Hyperplanes Shatter Fewer Points



Thin hyperplanes can implement all 8 dichotomies

Only 4 of the 8 dichotomies can be separated by hyperplanes with thickness ρ

What matters is the thickness of the hyperplane relative to the spacing of the data

MAIN RESULT: (VC dimension of ρ-fat hyperplanes)

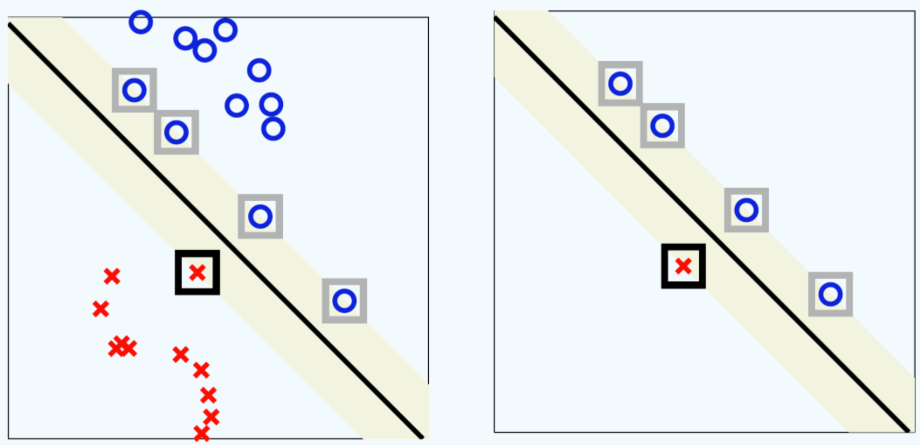
Suppose the input space is a ball of radius R in $\mathbb{R}^d$, so $\|\mathbf{x}\| \leq R$. Then

$$d_{\mathrm{vc}}(\rho) \leq \left\lceil R^2/\rho^2 \right\rceil + 1$$

where $\left\lceil R^2/\rho^2 \right\rceil$ is the smallest integer greater than or equal to R^2/ρ^2 . Combining results

$$d_{\mathrm{vc}}(\rho) \leq \min\left(\left\lceil R^2/\rho^2 \right\rceil, d \right) + 1$$

Bounding the Cross Validation Error



(a) All data (b) Only support vectors

$$E_{\text{cv}} = \frac{1}{N}\sum_{n=1}^{N} \mathrm{e}_n \quad \text{(LOO)}$$

But $\mathrm{e_n} = 0$ for any point not support vector

For the support vectors $\mathrm{e_n} \leq 1$ (binary classification)

$$E_{\text{out}}(\text{SVM}) \leq E_{\text{cv}}(\text{SVM}) = \frac{1}{N}\sum_{n=1}^{N} \mathrm{e_n} \leq \frac{\#\text{ support vectors}}{\mathrm{N}}$$

Clearly removing points not support vectors do not influence the solution

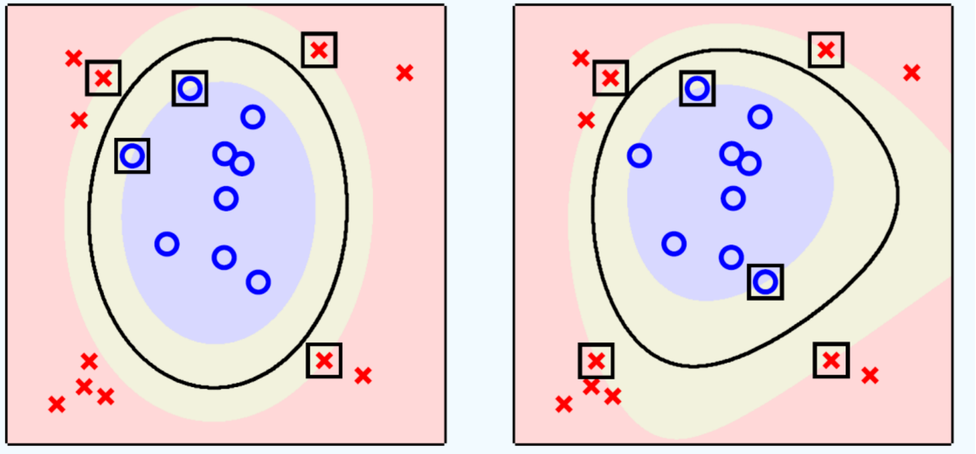
Non-Separable data

- Let's assume the data are separable after some non-linear data transformation

$$\mathbf{z}_n = \Phi(\mathbf{x}_\mathrm{n})$$

- We solve the hard-margin SVM in the $\mathcal{Z}$-space:

$$\Phi_2(\mathbf{x}) = (x_1, x_2, x_1^2, x_1 x_2, x_2^2)$$


(a) SVM with Φ_2 (b) SVM with Φ_3

$$\underset{\tilde{b},\tilde{\mathbf{w}}}{\text{minimize}} \quad \frac{1}{2}\tilde{\mathbf{w}}^\mathrm{T}\tilde{\mathbf{w}}$$

Subject to: $y_n\left(\tilde{\mathbf{w}}^\mathrm{T}\mathbf{z}_\mathrm{n} + \tilde{b}\right) \geq 1,$
$(\mathrm{n} = 1, \ldots, N)$

$$g(\mathbf{x}) = \mathrm{sign}(\tilde{\mathbf{w}}^{*T}\Phi(\mathbf{x}) + \tilde{b}^*)$$

- NLT is used to lower E_{in}. However, you pay a price to get this sophisticated boundary in terms of a larger d_{vc} and a tendency to overfit.

- Now only a light increasing of the boundary complexity is observed from Φ_2 to Φ_3

Lagrange **Dual** for a QP-Problem

$$\underset{\mathbf{u}\in\mathbb{R}^L}{\text{minimize}} \ \frac{1}{2}\mathbf{u}^{\mathrm{T}}\mathrm{Q}\mathbf{u} + \mathbf{p}^{\mathbf{T}}\mathbf{u}$$

Subject to: $\boldsymbol{a}^{\boldsymbol{T}}\mathbf{u} \geq \mathbf{c}$

$$\underset{\mathbf{u}\in\mathbb{R}^L}{\text{minimize}} \ \frac{1}{2}\mathbf{u}^{\mathrm{T}}\mathrm{Q}\mathbf{u} + \mathbf{p}^{\mathbf{T}}\mathbf{u} + \max_{\alpha\geq 0} \alpha\left(c - \boldsymbol{a}^{\boldsymbol{T}}\mathbf{u}\right)$$

$$\mathcal{L}(\boldsymbol{u},\alpha) = \frac{1}{2}\mathbf{u}^{\mathrm{T}}\mathrm{Q}\mathbf{u} + \mathbf{p}^{\mathbf{T}}\mathbf{u} + \alpha\left(c - \boldsymbol{a}^{\boldsymbol{T}}\mathbf{u}\right)$$

$$\min_{\boldsymbol{u}} \max_{\alpha\geq 0} \mathcal{L}(\boldsymbol{u},\alpha)$$

- For convex quadratic programming when $\mathcal{L}(\boldsymbol{u},\alpha)$ has the same form that here and there exits $\boldsymbol{u}$ such that $c - \boldsymbol{a}^{\boldsymbol{T}}\mathbf{u} \leq 0$, then can be proved

$$\min_{\mathbf{u}} \max_{\alpha\geq 0} \mathcal{L}(\mathbf{u},\alpha) = \max_{\alpha\geq 0} \min_{\mathbf{u}} \mathcal{L}(\mathbf{u},\alpha)$$

- The optimization problem on the RHS is easier to solve since the minimum in ***u*** can be calculated analitically

Optimal solution conditions(KKT)

- The Karush-Khun-Tucker (KKT) conditions characterize the optimal solution $(\boldsymbol{u}^*, \boldsymbol{\alpha}^*)$ of the primal and dual formulations of a Lagragian QP-problem,

$$\min_{\mathbf{u}} \max_{\boldsymbol{\alpha}\geq 0} \mathcal{L}(\mathbf{u}, \boldsymbol{\alpha}) = \max_{\boldsymbol{\alpha}\geq 0} \min_{\mathbf{u}} \mathcal{L}(\mathbf{u}, \boldsymbol{\alpha})$$

1. Primal and dual constraints : $\mathrm{a}_m^T \boldsymbol{u}^* \geq \boldsymbol{c}_m,\ \alpha_m \geq 0,\ m = 1,2,..,M$
2. Complementary slackness: $\alpha_m^* (y_m (\mathbf{x}_m^T \mathbf{w}^* + b^*) - 1) = 0\ \ (m = 1, \dots, M)$
3. Stationarity respect to $\boldsymbol{u}$: $\boldsymbol{\nabla}_u \mathcal{L}(\mathbf{u}, \boldsymbol{\alpha})|_{\boldsymbol{u}=\boldsymbol{u}^*, \boldsymbol{\alpha}=\boldsymbol{\alpha}^*} = \mathbf{0}$

- The consequence is
  1. The primal problem can be used to discover relations between the true unknows and its Lagrange multipliers.
  2. These relations allow reformulate the primal problem as an equivalent optimization problem on the Lagrange multipliers (dual)
  3. The solution of the dual problem give us the solution of the primal one.

Dual of the Hard-Margin SVM

$$\underset{b,\mathbf{w}}{\text{minimize}} \;\; \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}\,, \qquad \text{Subject to:} \;\; y_n\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_{\mathrm{n}} + b\right) \geq 1, \;\; (\mathrm{n} = 1, \dots, N)$$

$$\boldsymbol{\mathcal{L}}(b, \mathbf{w}, \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + \sum_{n=1}^{N} \alpha_n \left(1 - y_n\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_{\mathrm{n}} + b\right)\right)$$

$$= \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} - \sum_{n=1}^{N} \alpha_n y_n \, \mathbf{w}^{\mathrm{T}}\mathbf{x}_{\mathrm{n}} - b\sum_{n=1}^{N} \alpha_n y_n + \sum_{n=1}^{N} \alpha_n$$

Very important to realize that we must first minimize $\mathcal{L}$ respecto to (b, **w**) and then maximize with respect to $\alpha \geq 0$ **(this encourage to satisfy the constraints)**

We need derivatives of $\mathcal{L}$:

$$\frac{\partial \boldsymbol{\mathcal{L}}}{\partial b} = -\sum_{n=1}^{N} \alpha_n y_n \quad \text{and} \quad \frac{\partial \boldsymbol{\mathcal{L}}}{\partial \mathbf{w}} = \mathbf{w} - \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n$$

Setting these derivatives to zero,

$$\boxed{\sum_{n=1}^{N} \alpha_n y_n = 0} \qquad \boxed{\mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n}$$

Dual of the Hard-Margin SVM

- Plugging the above stationary conditions back into the Lagragian we get

$$\boldsymbol{\mathcal{L}}(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{M} y_n y_m \alpha_n \alpha_m x_n^T x_m + \sum_{n=1}^{N} \alpha_n$$

We must maximize $\boldsymbol{\mathcal{L}}(\boldsymbol{\alpha})$ subject to $\boldsymbol{\alpha} \geq \mathbf{0}$, and the condition $\sum_{n=1}^{N} \alpha_n y_n = 0$

We formulate this as a equivalent minimization

$$\underset{\alpha \in \mathbb{R}^N}{\text{minimize}} \ \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{M} y_n y_m \alpha_n \alpha_m x_n^T x_m - \sum_{n=1}^{N} \alpha_n$$

Subject to: $\sum_{n=1}^{N} \alpha_n y_n = 0$
$\alpha_n \geq 0 \quad (n = 1, \dots, N)$

This is a new QP-problem

Recovering the SVM from the Dual Solution

What remains is to compute the optimal hyperplane

$$\mathbf{w}^* = \sum_{n=1}^{\boldsymbol{N}} \alpha_n^* y_n \mathbf{x}_n$$

The optimal $(\mathbf{u}^*, \boldsymbol{\alpha}^*)$ of the problem $\max_{\alpha \geq 0} \min_{\mathbf{u}} \mathcal{L}(\mathbf{u}, \boldsymbol{\alpha})$ satisfy the KKT conditions

$$\boxed{\alpha_m^*(y_m(\mathbf{x}_m^T \mathbf{w}^* + b^*) - 1) = 0 \quad (m = 1, \dots, M)}$$

Complementary slackness

This means any data point with $\alpha_m^* > 0$ verifies $y_m(\mathbf{x}_m^T \mathbf{w}^* + b^*) = 1$

Let be $\alpha_s^* > 0$, then $b^* = y_s - w^{*T} x_s = y_s - \sum_{n=1}^{\boldsymbol{N}} \alpha_n^* y_n \mathbf{x}_n^T \mathbf{x}_s$

$$g(x) = \text{sign}(\mathbf{w}^{*T}\mathbf{x} + b^*) = \text{sign}\left(\sum_{n=1}^{N} y_n \alpha_n^* \, \mathbf{x}_n^T \mathbf{x} + b^*\right)$$

Remember that only a few support vectors have $\alpha_n^* > 0$

ALGORITHM

Hard-Margin SVM with Dual QP

1: Construct $\mathrm{Q_D}$ and $\mathrm{A_D}$ from the QP-Dual formulation

$$\mathrm{Q_D} = \begin{bmatrix} y_1 y_1 \mathbf{x}_1^{\mathrm{T}} \mathbf{x}_1 & \dots & y_1 y_N \mathbf{x}_1^{\mathrm{T}} \mathbf{x}_N \\ y_2 y_1 \mathbf{x}_2^{\mathrm{T}} \mathbf{x}_1 & \dots & y_2 y_N \mathbf{x}_2^{\mathrm{T}} \mathbf{x}_N \\ \vdots & \vdots & \vdots \\ y_N y_1 \mathbf{x}_N^{\mathrm{T}} \mathbf{x}_1 & \dots & y_N y_N \mathbf{x}_N^{\mathrm{T}} \mathbf{x}_N \end{bmatrix} \text{ and } \mathrm{A_D} = \begin{bmatrix} \mathbf{y}^{\mathrm{T}} \\ -\mathbf{y}^{\mathrm{T}} \\ \mathrm{I}_{N \times N} \end{bmatrix}.$$

2: Use a QP-solver to optimize the dual problem:

$$\boldsymbol{\alpha}^* \leftarrow \mathrm{QP}(\mathrm{Q_D}, -\mathbf{1}_N, \mathrm{A_D}, \mathbf{0}_{N+2}).$$

3: Let s be a support vector for which $\alpha_s^* > 0$. Compute b^*,

$$b^* = y_s - \sum_{\alpha_n^* > 0} y_n \alpha_n^* \mathbf{x}_n^{\mathrm{T}} \mathbf{x}_s.$$

4: Return the final hypothesis

$$g(\mathbf{x}) = \mathrm{sign}\left(\sum_{\alpha_n^* > 0} y_n \alpha_n^* \mathbf{x}_n^{\mathrm{T}} \mathbf{x} + b^* \right).$$

Kernel Trick via Dual SVM

- Let's consider again the problem of solving nonlinear SVM after a nonlinear transform $\Phi: \mathcal{X} \rightarrow \mathcal{Z}$

- We can observe that the only and important change in the dual formulation respect *to* the linear one is the needed of computing the $\mathcal{Z}$- space inner product

$$\Phi(\mathbf{x})^T \Phi(\mathbf{x}')$$

This inner product is needed to create $\mathbf{Q}_\mathrm{D}$ and to compute $g(\mathbf{x})$
When the dimension d' of $\Phi(\mathbf{x})$ is very large (or infinity) the inner product computation is inefficient or non-viable.

The kernel trick defines a mechanism to compute the inner product without explicitly compute the function $\Phi(\mathbf{x})$.

Let's define a function that combines both the transform and inner product

$$K_\Phi(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x})^T \Phi(\mathbf{x}') \qquad \text{Kernel Function}$$

Hard-Margin SVM with Kernel

Hard-Margin SVM with Kernel

1: Construct Q_D from the kernel K, and A_D:

$$Q_D = \begin{bmatrix} y_1 y_1 K_{11} & \dots & y_1 y_N K_{1N} \\ y_2 y_1 K_{21} & \dots & y_2 y_N K_{2N} \\ \vdots & \vdots & \vdots \\ y_N y_1 K_{N1} & \dots & y_N y_N K_{NN} \end{bmatrix} \quad \text{and} \quad A_D = \begin{bmatrix} \mathbf{y}^T \\ -\mathbf{y}^T \\ I_{N \times N} \end{bmatrix},$$

where $K_{mn} = K(\mathbf{x}_m, \mathbf{x}_n)$. (K is called the *Gram* matrix.)

2: Use a QP-solver to optimize the dual problem:

$$\boldsymbol{\alpha}^* \leftarrow \mathsf{QP}(Q_D, -\mathbf{1}_N, A_D, \mathbf{0}_{N+2}).$$

3: Let s be any support vector for which $\alpha_s^* > 0$. Compute

$$b^* = y_s - \sum_{\alpha_n^* > 0} y_n \alpha_n^* K(\mathbf{x}_n, \mathbf{x}_s).$$

4: Return the final hypothesis

$$g(\mathbf{x}) = \text{sign}\left(\sum_{\alpha_n^* > 0} y_n \alpha_n^* K(\mathbf{x}_n, \mathbf{x}) + b^* \right).$$

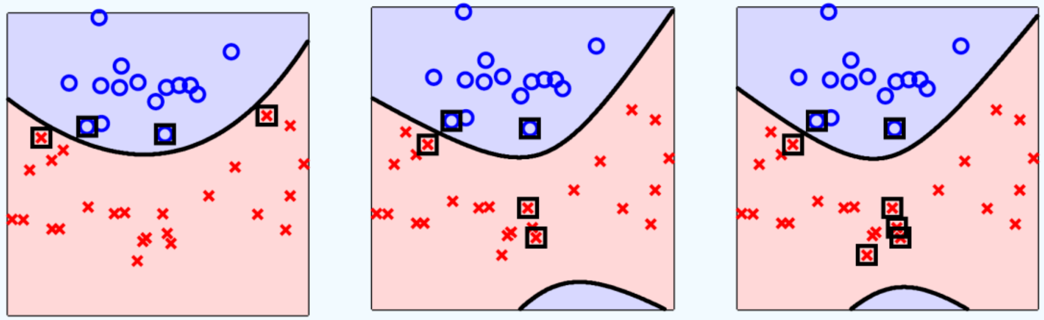
The algorithm is equivalent to the lineal one. (HM SVM with Dual QP), but needing the kernel values

First of all we needs to characterize the class of function and its properties

Example: $K(x, x') = (\zeta + \gamma x^T x')^Q$

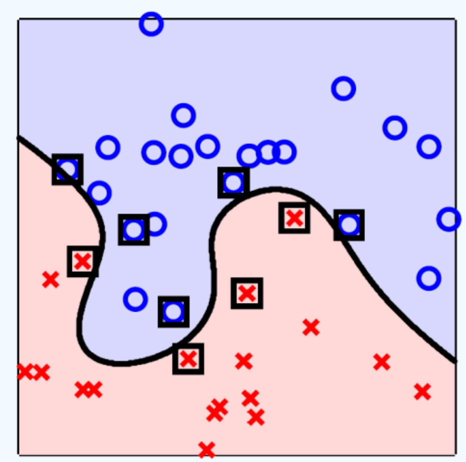
- This kernel is called the *polynomial kernel*

$K(\mathbf{x}, \mathbf{x}') = (\zeta + \gamma \mathbf{x}^T \mathbf{x}')^2$, $\zeta > 0$ $\gamma > 0$ $\mathbf{x} \in \mathbb{R}^d$



$(1 + 0.001\mathbf{x}^{\mathrm{T}}\mathbf{x}')^2$ $1 + (\mathbf{x}^{\mathrm{T}}\mathbf{x}') + (\mathbf{x}^{\mathrm{T}}\mathbf{x}')^2$ $(1 + 1000\mathbf{x}^{\mathrm{T}}\mathbf{x}')^2$

$K(x, x') = (\zeta + \gamma x^T x')^{10}$



Now $K_\Phi(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x})^T \Phi(\mathbf{x}')$,
with $\Phi(\mathbf{x}) = \mathbf{c}^{\mathrm{T}} \Phi_2^d(\mathbf{x})$ where

$\mathbf{c}^{\mathrm{T}} = \left(1, \sqrt{2\gamma\zeta}, \sqrt{2\gamma\zeta}, \cdots, \sqrt{2\gamma\zeta}, \gamma, \gamma, \cdots, \gamma\right)$ coeficients

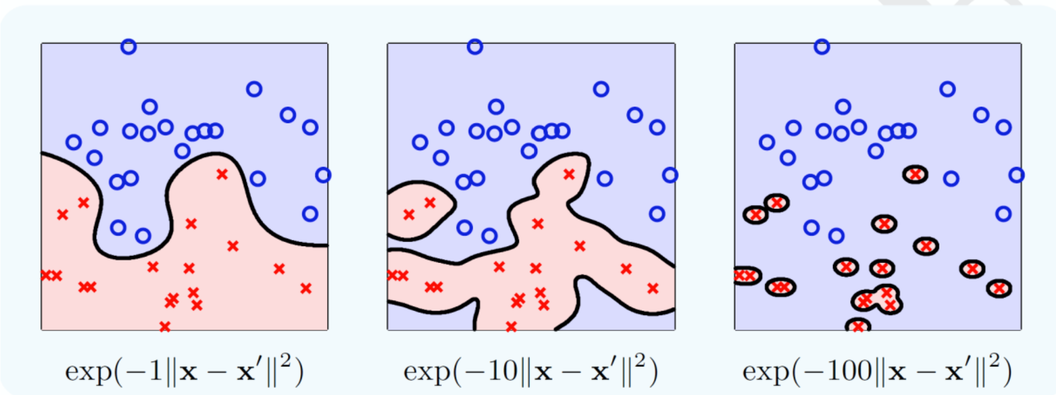
$\Phi_2^d(\mathbf{x}) = (1,, x_1, x_2, \cdots, x_d, x_1 x_1, x_1 x_2, \ldots, x_d x_d)$ NLT

We can efficiently use high-dimensional kernels.

The model complexity is controlled by maximizing the margin

Example: $K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2)$

- This kernel is called the *Gaussian-RBF Kernel*, where $\gamma > 0, \;\; \mathbf{x} \in \mathbb{R}^d$



- To understand the implicit NLT let consider the case $\gamma = 1$, $\mathrm{x} \in \mathbb{R}$

$$
\begin{aligned}
K(\mathrm{x}, \mathrm{x}') &= \exp(-\mathrm{x}^2) \cdot \exp(2\mathrm{x}\mathrm{x}') \cdot \exp(-\mathrm{x}'^2) \\
&= \exp(-\mathrm{x}^2) \cdot \exp\left(\sum_{k=0}^{\infty} \frac{2^k \mathrm{x}^k \mathrm{x}'^k}{k!}\right) \cdot \exp(-\mathrm{x}'^2)
\end{aligned}
$$

$$
\boxed{\Phi(\mathrm{x}) = \exp(-\mathrm{x}^2) \cdot \left(1, \sqrt{\frac{2^1}{1!}}\mathrm{x},, \sqrt{\frac{2^1}{1!}}\mathrm{x}^2, \sqrt{\frac{2^1}{1!}}\mathrm{x}^3, \ldots\right)}
$$

Note that in this case $\Phi(\mathrm{x})$ is an infinity NLT

Choice of Kernels

- Three kernels are popular in practice:
  - Linear: Polynomial con Q=1, ζ=0 , γ=1
  - Polynomial: Q < 10 and appropiated ζ and γ (not easy of selecting)
  - Gaussian-RBF: only one parameter $\gamma \in [0,1]$.

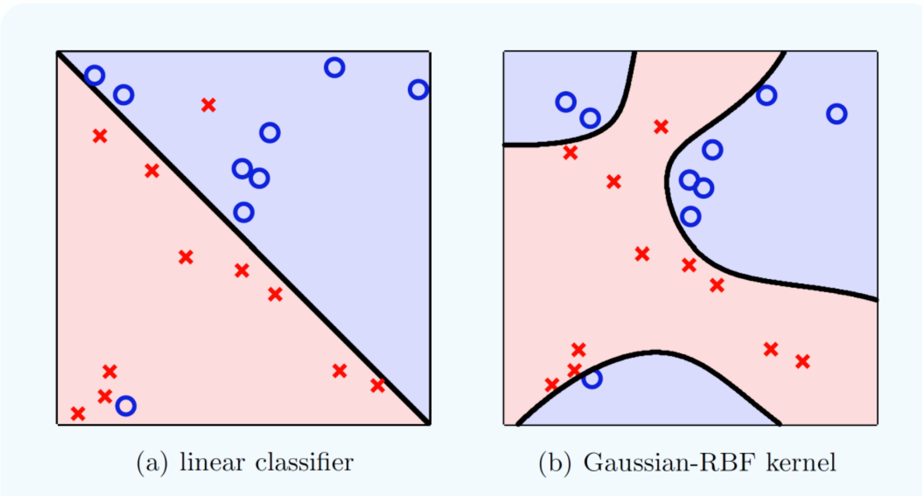
- Many other kernels can be proposed as combinations of simpler kernels.

- New kernels can be proposed but it is not an eay task since the function K(x,x') must acomplish the Mercer's condition, that is its Gram Matrix K is positive semidefinite for any given sample $\{\mathbf{x_1}, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$

$$\mathrm{K} = \begin{pmatrix} \mathrm{K}(\mathbf{x_1}, \mathbf{x}_1) & \cdots & \mathrm{K}(\mathbf{x_1}, \mathbf{x}_\mathrm{N} \\ \vdots & \ddots & \vdots \\ \mathrm{K}(\mathbf{x}_\mathrm{N}, \mathbf{x}_1) & \cdots & \mathrm{K}(\mathbf{x}_\mathrm{N}, \mathbf{x}_\mathrm{N}) \end{pmatrix}$$

Soft-margin SVM

- The hard-margin SVM assumes that the data are separable in the $\mathcal{Z}$ space



But according to the type of noise overfitting can appear after the transformation.

A best strategy is to assume some noise in the labels using a "soft" fomulation

Soft-Margin SVM: Let introduce an amount of margin violation $\xi_n \geq 0$ for each point $(\mathbf{x}_n, y_n)$ $\Longrightarrow$ $y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1 - \xi_n$

$$\underset{\mathrm{b},\mathbf{w},\boldsymbol{\xi}}{\text{minimize}} \quad \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + \mathrm{C}\sum_{n=1}^{N}\xi_n$$

Subject to: $y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1 - \xi_n$, for n $= 1,2,\ldots,N$

$\xi_n \geq 0$ for n $= 1,2,\ldots,N$

This problem can be solved by quadratic programming using the dual problem

Soft-margin SVM & Regularization

- Let's take a closer look at the parameter C
  - **If C is large**, it means that we do want all errors ξ_n to be as small as possible with the trade-off of higher complexity hypothesis
  - **If C is small**, we will tolerate some amounts of errors with less complicated hypothesis

Such trade-off was found when we studied regularization

Let's define

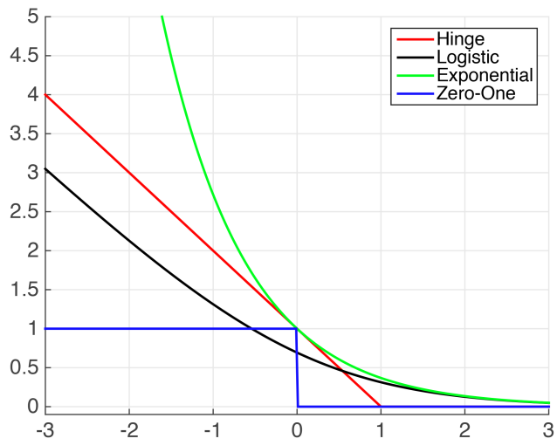
$$E_{SVM}(b, \mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} \max(1 - y_n(\mathbf{w}^T \mathbf{x}_n + b), 0)$$

Then the soft-margin SVM can be written as the following optimization problem:

$$\min_{b,\mathbf{w}} \lambda \mathbf{w}^T \mathbf{w} + E_{SVM}(b, \mathbf{w}), \qquad \lambda = 1/2CN$$

Soft-margin SVM can be viewed as a special case of regularized classification with $E_{SVM}(b, \mathbf{w})$ as a surrogate for the in-sample error and $\mathbf{w}^T\mathbf{w}$ as the regularizer

Soft-SVM and SGD



$$\min_{b,\mathbf{w}} \lambda \mathbf{w}^{\boldsymbol{T}}\mathbf{w} + E_{SVM}(b,\mathbf{w}), \qquad \lambda = 1/2CN$$

$$\Downarrow$$

$$\min_{b,\mathbf{w}} \lambda R(\mathbf{w}) + L(\mathbf{x}, y, b, \mathbf{w}),$$

SVM : Hinge Loss
Perceptron: Zero-One
Logistic Regression: Logistic
Boosting: Exponential

A strong connection between differents binary classifiers appears as consequence that all of them solve the same problem minimizing a different surrogate for the zero-one loss function, and using a different algorithm to reach its solution.

For separable or almost separable data set the soft-SVM provides in general the best option but for overlapping classes LR can provide a better performance.

SVM is not the only classifier that take advantage of the kernel trick, LR and many others can be formulated to be used with kernels.

Soft-Margin SVM: Summary

- Deliver a large-margin hyperplane, and in so doing it can control the effective model complexity
- Deal with high- or infinite-dimensional transforms using the kernel trick.
- Express the final hypothesis *g*(x) using only a few support vectors, their corresponding Lagrange multipliers, and the kernel.
- Control the sensivity to outliers and regularize the solution through setting C appropriately
- When C y the kernel are chosen properly, the soft-margin SVM enjoy a low E_{out} and define one of the most useful classification models.