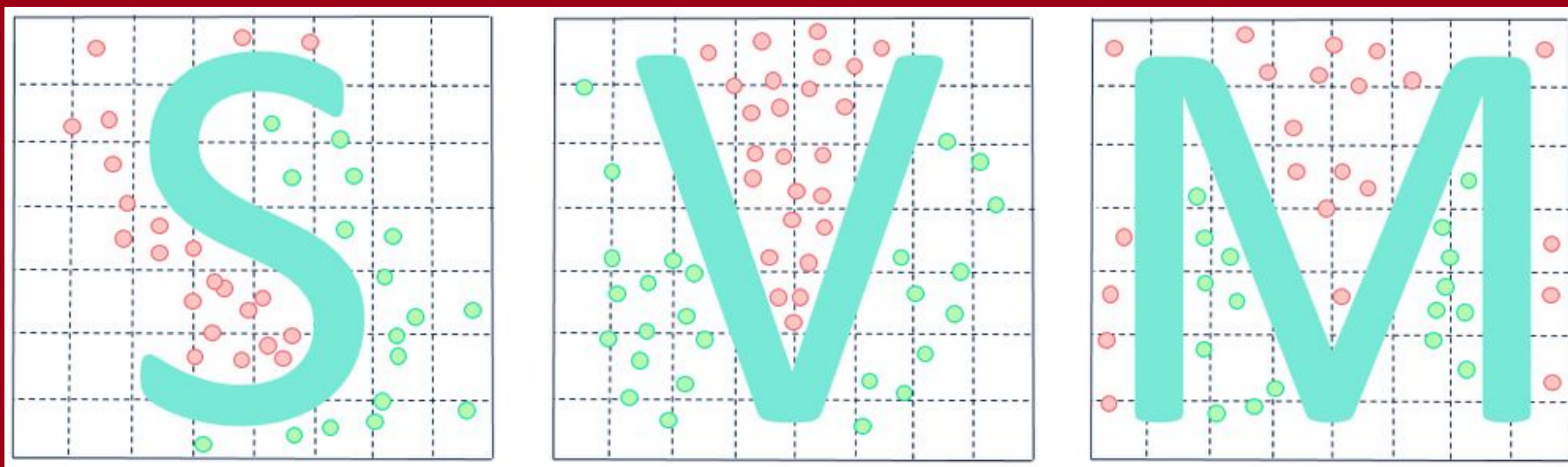
DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

UNIVERSITÀ
DEGLI STUDI
DI PADOVA



Support Vector Machines

Machine Learning 2022-23
UML book chapter 15
Slides: F. Chiariottim, P. Zanuttigh, F. Vandin

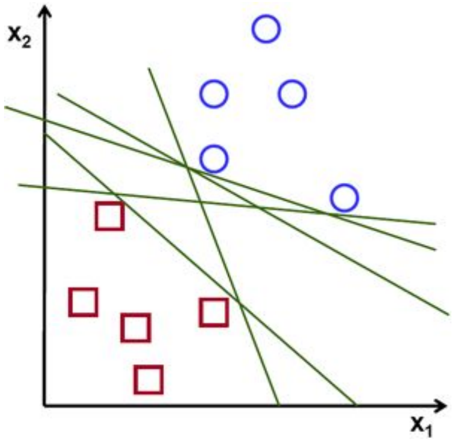
SVM: Topics

1. Classification margin
2. Hard-SVM (linearly separable data and linear model)
3. Soft-SVM (not linearly separable data, still a linear model)
4. Kernel Methods for SVM (non-linear classification)
5. Examples and exercises
6. LAB2 on SVM

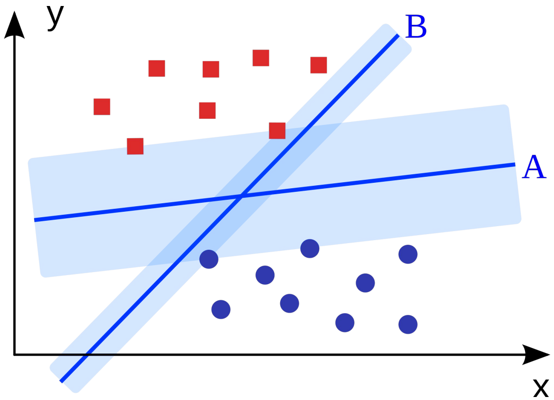
Classification Margin

Consider a classification problem with two classes:

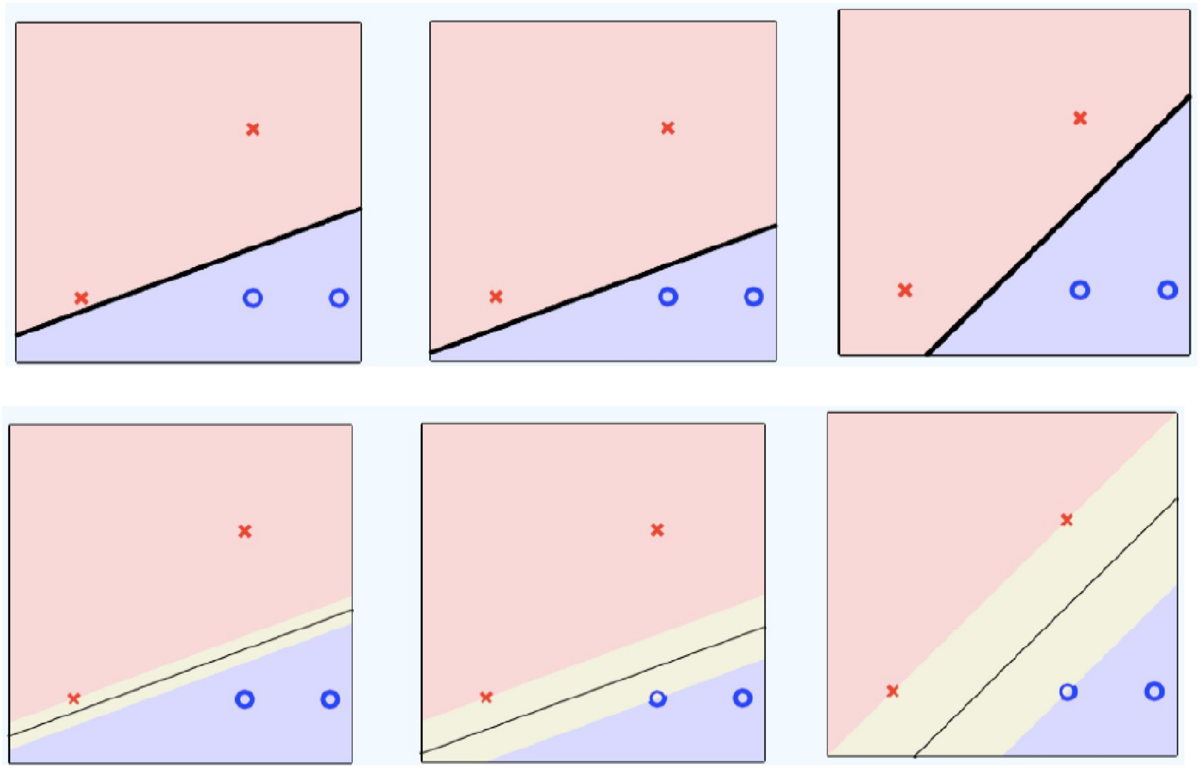
- Training data: $S = ((\boldsymbol{x_1}, y_1), \ldots, (\boldsymbol{x_m}, y_m))$
- $\boldsymbol{x_i} \in \mathbb{R}^d$ ($\mathbb{R}^2$ in the visual example for simplicity)
- Label set $\mathcal{Y} = \{-1,1\}$
- Hypothesis set $\mathcal{H}$ = halfspaces



- *Assumption*: the data is linearly separable
  → there exist a halfspace that perfectly classify the training set
- Find a solution: there are multiple separating hyperplanes that correctly classify the training set : *which one is the best ?*



Classification Margin: Example



- *Margin*: minimum distance from an example in the training set
- Idea: best separating hyperplane is the one with the largest margin
  - Can tolerate more "*noise*"

Linearly Separable Training Set

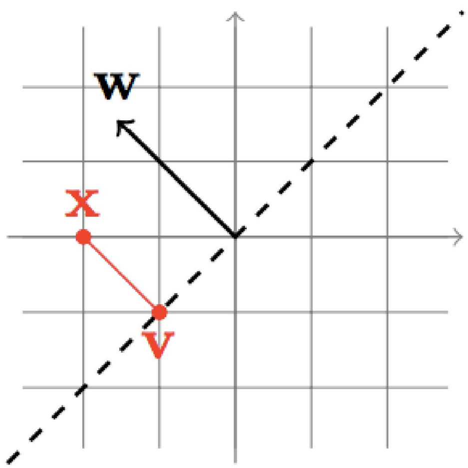
Linearly Separable Training Set

- A training set $S = ((\boldsymbol{x_1}, y_1), \ldots, (\boldsymbol{x_m}, y_m))$ is linearly separable if there exists a halfspace (**w**, b) such that $y_i = sign(< \boldsymbol{w}, \boldsymbol{x_i} > +b)$ $\forall i = 1, \ldots, m$
  - i.e., it perfectly separate all data in the training set
  - or, equivalently $\forall i : y_i(< \boldsymbol{w}, \boldsymbol{x_i} > +b) > 0$

Margin

- Given a separating hyperplane defined by $L = \{\boldsymbol{v}: < \boldsymbol{v}, \boldsymbol{w} > +b = 0\}$ and given a sample ***x***, the distance of ***x*** to L is

$$d(\boldsymbol{x}, L) = \min\{\|\boldsymbol{x} - \boldsymbol{v}\|: \boldsymbol{v} \in L\}$$



Theorem

If $\|\boldsymbol{w}\| = 1$ then $d(\boldsymbol{x}, L) = |< \boldsymbol{w}, \boldsymbol{x} > +b|$

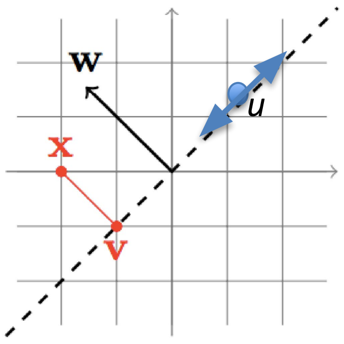
In this case the margin is $\min_i |< \boldsymbol{w}, \boldsymbol{x_i} > +b|, \ \boldsymbol{x_i} \in S$

- The closest examples are called *support vectors*

Demonstration

Theorem

If $\|w\| = 1$ then $d(x, L) = |\langle w, x\rangle + b|$



1. The distance between ***x*** and the hyperplane is:

$$\min\{\|x - z\|, z: \langle w, z\rangle + b = 0\}$$

2. Define point $v = x - (\langle w, x\rangle + b)w$ **(*)**
   a) It lies on the hyperplane:
   $\langle w, v\rangle + \mathrm{b} = \langle w, x\rangle - (\langle w, x\rangle + b)\|w\|^2 + b = 0 \rightarrow \langle w, v\rangle = -\mathrm{b}\ \ (**)$
   b) The distance is $d(x, v) = |< w, x > +b|$
   $\|x - v\| = \|x - x + (\langle w, x\rangle + b)\, w\| = |\langle w, x\rangle + b|\,\|w\| = |\langle w, x\rangle + b|$

3. Since v lies on the hyperplane the distance is at most the one of v, let's prove that no other point is closer, take a generic point ***u*** on hyperplane:

$$\|x - u\|^2 = \|(x - v) + (v - u)\|^2 =$$

$$= \|x - v\|^2 + \|v - u\|^2 + 2\langle x - v, v - u\rangle$$

$from\ (*)\ and\ norm \geq 0$

$$\geq \|x - v\|^2 + 2\langle x - x + (\langle w, x\rangle + b)w, v - u\rangle$$

$green = 0\ from (**) and < w, u > = -b$

$$= \|x - v\|^2 + 2(\langle w, x\rangle + b)\langle w, v - u\rangle = \|x - v\|^2$$

Support Vector Machines (Hard-SVM)

Hard-SVM: seek for the separating hyperplane with largest margin (*works only for linearly separable data*)

Computational problem:

Recall theorem: If $||\boldsymbol{w}|| = 1$ then the margin is $|\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b|$

$$\underset{(\boldsymbol{w},b):||\boldsymbol{w}||=1}{\operatorname{argmax}} \; \min_i |< \boldsymbol{w}, \boldsymbol{x_i} > +b|$$

Need to correctly classify all samples

$$\textit{subject to } \forall i: y_i(< \boldsymbol{w}, \boldsymbol{x_i} > +b) > 0$$

Equivalent formulation (in the case of separable data):

$$\underset{(\boldsymbol{w},b):||\boldsymbol{w}||=1}{\operatorname{argmax}} \; \min_i y_i(< \boldsymbol{w}, \boldsymbol{x_i} > +b)$$

Quadratic Programming Formulation

- Input: $S = ((\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_m, y_m))$
- Solve:

$$(\boldsymbol{w}_0, b_0) = argmin_{(\boldsymbol{w},b)} \|\boldsymbol{w}\|^2$$

subject to $\forall i: y_i(<\boldsymbol{w}, \boldsymbol{x}_i> + b) \geq 1$

- Output: $\hat{\boldsymbol{w}} = \frac{\boldsymbol{w}_0}{\|\boldsymbol{w}_0\|}, \hat{b} = \frac{b_0}{\|\boldsymbol{w}_0\|}$

The objective is a convex quadratic function, constraints are linear inequalities: can be solved with quadratic programming solvers

*It is equivalent to Hard-SVM: instead of maximizing margin, fix margin to 1 by scaling its unit of measure with **w**, search of max margin equals to search for minimum norm scaling factor **w***

Demonstration: Hard-SVM $\leftrightarrow$ Quadratic Programming

Hard-SVM: $\underset{(\mathbf{w},b):\|\mathbf{w}\|=1}{\operatorname{argmax}} \; \min_i y_i(<\mathbf{w},\boldsymbol{x_i}> + b)$

QP: $(\mathbf{w}_0, b_0) = argmin_{(\mathbf{w},b)} \|\mathbf{w}\|^2$ *subject to* $\forall i$: $y_i(<\mathbf{w},\boldsymbol{x_i}> + b) \geq 1$

Output: $\hat{\mathbf{w}} = \frac{\mathbf{w}_0}{\|\mathbf{w}_0\|}$, $\hat{b} = \frac{b_0}{\|\mathbf{w}_0\|}$

1. Let *($\mathbf{w}$*,b*) be a* solution of Hard-SVM
2. Define $\gamma^* = \min_{i\in[m]} y_i(<\mathbf{w}^*,\boldsymbol{x_i}> + b^*)$, i.e., margin of ($\mathbf{w}^*$,b^*)
3. $\forall i$: $y_i(<\mathbf{w}^*,\boldsymbol{x_i}> + b^*) \geq \gamma^* \rightarrow y_i\left(<\frac{\mathbf{w}^*}{\gamma^*},\boldsymbol{x_i}> + \frac{b^*}{\gamma^*}\right) \geq 1$
4. The pair $\left(\frac{\mathbf{w}^*}{\gamma^*},\frac{b^*}{\gamma^*}\right)$ satisfies QP constraint: it is a solution and $\mathbf{w}_0$ is the one of minimum norm $\rightarrow$ $\|\mathbf{w}_0\| \leq \left\|\frac{\mathbf{w}^*}{\gamma^*}\right\| = \frac{1}{\gamma^*}$ $(\|\mathbf{w}^*\| = 1)$
5. $\forall i$: $y_i(<\hat{\mathbf{w}},\boldsymbol{x_i}> + \hat{b}) = \frac{1}{\|\mathbf{w}_0\|} y_i(<\mathbf{w}_0,\boldsymbol{x_i}> + b_0) \geq \frac{1}{\|\mathbf{w}_0\|} \geq \gamma^*$ (apply definition of $\hat{\mathbf{w}}$, then first inequality from purple condition, second from 4)
6. Since $\|\hat{\mathbf{w}}\| = 1$ and $(\hat{\mathbf{w}}, \hat{b})$ has a margin $\geq \gamma^*$ $\rightarrow$ $(\hat{\mathbf{w}}, \hat{b})$ is an optimal solution of Hard-SVM

Homogeneous Representation

Formulation with homogeneous halfspaces:

- Assume first component of $\boldsymbol{x} \in \mathcal{X}$ is 1 (homog. representation), then

$$\boldsymbol{w_0} = \underset{\boldsymbol{w}}{\operatorname{argmin}} ||\boldsymbol{w}||^2 \quad \text{subject to } \forall i: \; y_i < \boldsymbol{w}, \boldsymbol{x_i} > \geq 1$$

- Notice that this constraint is similar but not exactly the same as the non-homogeneous one
  - the bias now also goes inside the regularization
- However in practice no big difference

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

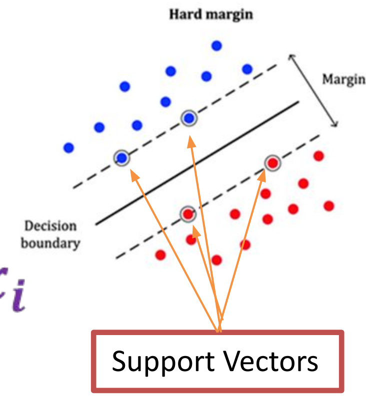
Theorem (Support Vectors)

The Support Vectors are the vectors at minimum distance from $\boldsymbol{w}_0$

They are the only training vectors that matter for defining $\boldsymbol{w}_0$!

Hypothesis:

- w_0 defined as before: $\boldsymbol{w_0} = \min_{\boldsymbol{w}} ||\boldsymbol{w}||^2$ subject to $\forall i: y_i \langle \boldsymbol{w}, \boldsymbol{x_i} \rangle \geq 1$
- $I = \{i: |\langle \boldsymbol{w_0}, \boldsymbol{x_i} \rangle| = 1\}$ (indexes of support vectors)



Thesis:

There exist coefficients $\alpha_1, \dots, \alpha_m$ such that $\boldsymbol{w_0} = \sum_{i \in I} \alpha_i \boldsymbol{x_i}$

- $\boldsymbol{x_i}$ for $i \in I$ are the "*Support Vectors*"
- Note: Solving Hard-SVM is equivalent to find α_i for the support vectors ($\alpha_i \neq 0$ only for support vectors)
- Demonstration not part of the course

Duality

The hard-SVM minimization

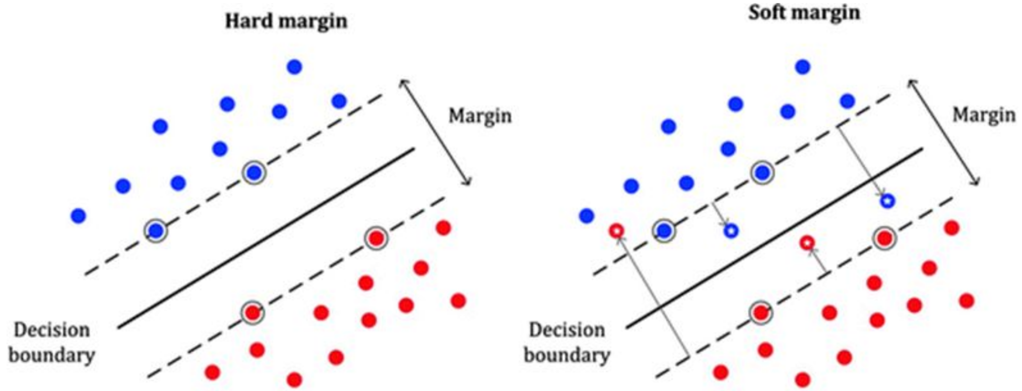
$$\boldsymbol{w}_0 = \underset{\boldsymbol{w}}{\operatorname{argmin}} \|\boldsymbol{w}\|^2 \text{ subject to } \forall i: y_i \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle \geq 1$$

Can be rewritten as a maximization problem:

$$\max_{\boldsymbol{\alpha} \in \mathbb{R}^m : \boldsymbol{\alpha} \geq 0} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$$

- It is called the "*dual*" problem
- Key property: only requires the inner product between instances but not the direct access to instances
  - Will be very useful for the "*kernel trick*"

Soft-SVM



Key issue: Hard-SVM needs the data to be linearly separable
 Almost never true in practical problems

We need an approach that can work also with non linearly separable data →*Soft-SVM*

Soft-SVM: Relax the constraints of Hard-SVM but take into account the violations of the separation into the objective function

Soft SVM: How it works

Relax the constraint :

- Introduce slack variables: $\boldsymbol{\xi} = (\xi_1, \dots, \xi_m),\ \xi_i \geq 0$
- for each *i = 1, . . . ,m*: $y_i(<\boldsymbol{w}, \boldsymbol{x_i}> +b) \geq 1 - \xi_i$
- ξ_i : how much the constraint is violated

Soft-SVM jointly minimizes

1. the norm of **w** (→ maximize margin)
2. the average of ξ_i (→ minimize constraint violations)

The tradeoff between the two objectives is controlled by a parameter $\lambda > 0$

Optimization Problem

- Input: $(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_m, y_m)$, parameter $\lambda > 0$
- Solve:

$$\min_{\boldsymbol{w}, b, \xi} \left(\lambda \|\boldsymbol{w}\|^2 + \frac{1}{m} \sum_{i=1}^{m} \xi_i \right)$$

subject to $\forall i: y_i(< \boldsymbol{w}, \boldsymbol{x_i} > \ +b) \geq 1 - \xi_i$ and $\xi_i \geq 0$

- Output $\boldsymbol{w}, b$

- Large λ : focus on margin ($\lambda \ \rightarrow \infty$: Hard-SVM)
- Small λ :focus on avoiding errors

Reformulate with Hinge Loss

Hinge Loss:

$$\ell^{hinge}((\boldsymbol{w}, b), (\boldsymbol{x}, y)) = \max\{0, 1 - y(< w, x > +b)\}$$

The problem can be reformulated with the Hinge loss:

$$\min_{\boldsymbol{w}, b} \left(\lambda \|\boldsymbol{w}\|^2 + \frac{1}{m} \sum_{i=1}^{m} \ell^{hinge}((\boldsymbol{w}, b), (\boldsymbol{x}_i, y_i)) \right)$$

$$L_S^{hinge}(\boldsymbol{w}, b)$$

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

The Two Formulations Are Equivalent

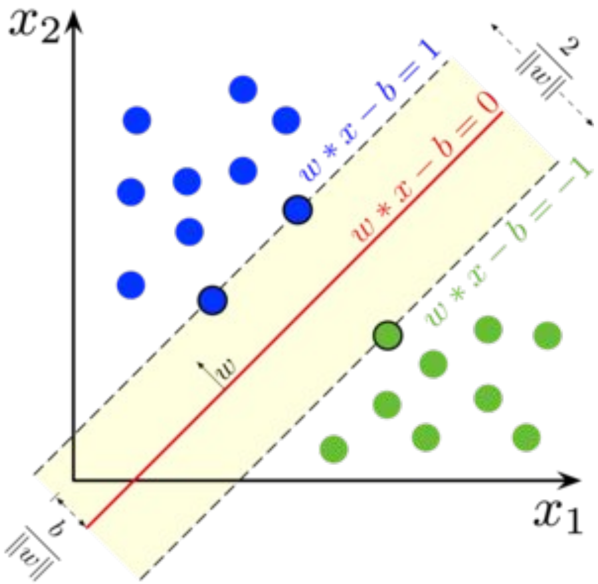
1. $\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \left(\lambda\|\boldsymbol{w}\|^2 + \frac{1}{m}\sum_{i=1}^{m}\xi_i\right)$ subject to $\forall i: y_i(<\boldsymbol{w},\boldsymbol{x_i}> \ +b) \geq 1-\xi_i$ and $\xi_i \geq 0$
2. $\min_{\boldsymbol{w},b} \left(\lambda\|\boldsymbol{w}\|^2 + \frac{1}{m}\sum_{i=1}^{m}\ell^{hinge}((\boldsymbol{w},b),(\boldsymbol{x}_i,y_i))\right)$

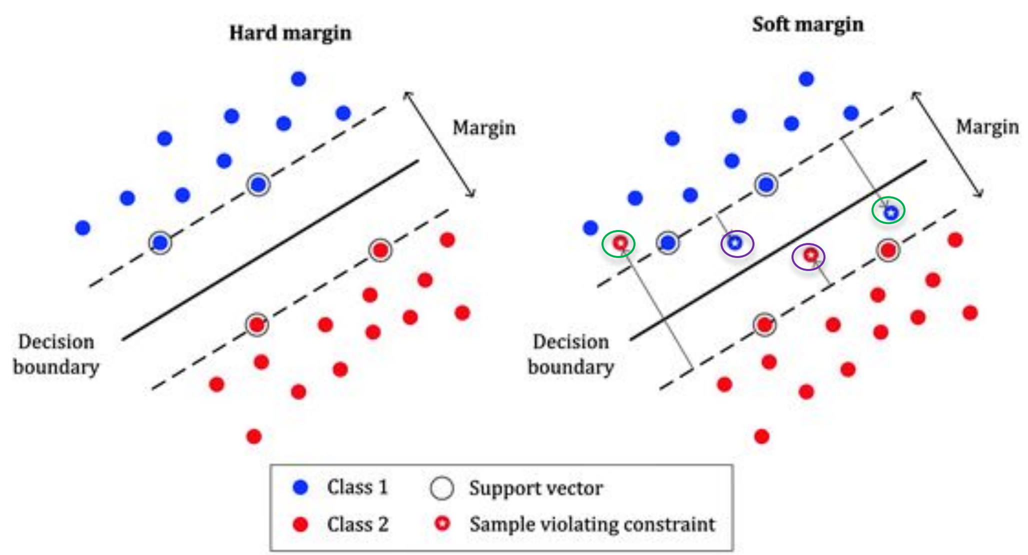
Demonstration:

1. Fix ***w***,*b* and consider minimization over ξ in (1)
2. $\xi_i \geq 0 \rightarrow$ the best assignment is 0 if $y_i(\langle w,x_i\rangle + b) \geq 1$ or $1 - y_i(\langle w,x_i\rangle + b)$ otherwise
3. This corresponds to $\xi_i = \ell^{hinge}((\boldsymbol{w},b),(\boldsymbol{x}_i,y_i)) \ \ \forall i$

$\longrightarrow$ Soft SVM falls into regularized loss minimization (RLM) paradigm

Examples



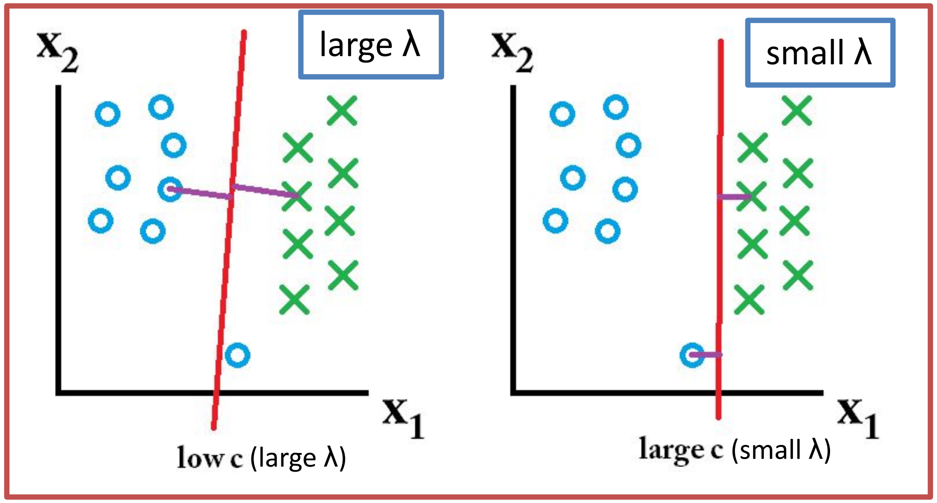


Two situations require $\xi_i > 0$

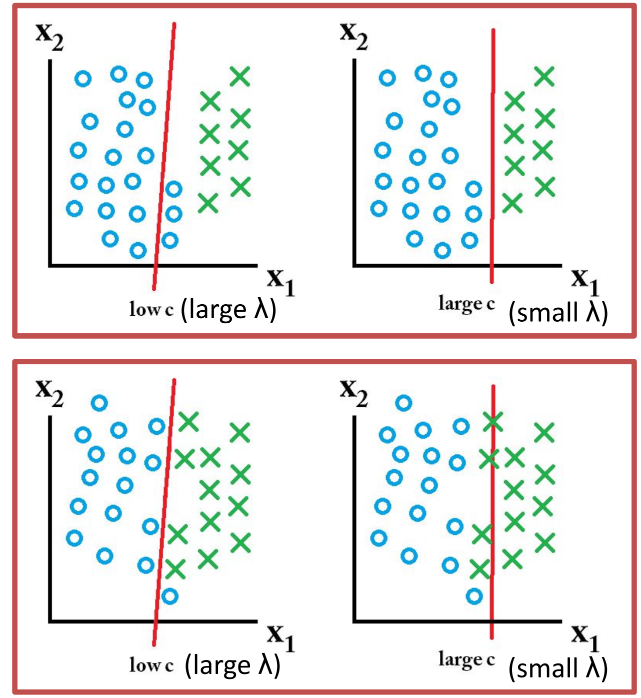
- Wrong classification ($\xi_i > 1$)
- Correct classification but violating margin ($0 < \xi_i \leq 1$)

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Practical SVM: The λ Parameter



Training Set



Examples on 2 different test sets

$$\min_{\mathbf{w}} \left(\lambda \|\mathbf{w}\|^2 + L_S^{\text{hinge}}(\mathbf{w}) \right)$$

The parameter λ controls the trade-off between a solution with a large margin that makes some errors or one with a lower margin but with less errors

(the parameter C in *sklearn*, *libsvm* and other ML tools has the same role but weights the loss term, i.e., works in the opposite direction)

images from stackexchange

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Homogeneous Version (Soft SVM)

- Rewrite with homogeneous coordinates

$$\min_{\boldsymbol{w}}\left(\lambda\|\boldsymbol{w}\|^2 + L_S^{hinge}(\boldsymbol{w})\right)$$

- The Hinge loss is given by

$$L_S^{hinge}(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^{m}\max\{0, 1 - y_i < \boldsymbol{w}, \boldsymbol{x}_i >\}$$

Approaches to solve the problem:

- Use standard solvers for optimization problems
- Use Stochastic Gradient Descent (wait for SGD lecture!)