Introduction to the use of SVM [IIA – Lect.______]

Alessio Micheli

micheli@di.unipi.it

Dipartimento di Informatica
Università di Pisa - Italy

Computational Machine

SVM:
- INTRO
- WHAT IS SVM?
- I NOSTRI OBIETTIVI SULLE SVM

1)
- MARGIN EXAMPLE
  - Toward margin example opt. problem & Canonical rep. of hyperplane and SV
  - Two useful fact
  - Hard margin SVM
    - Dual problem
    - Solution
    - role of support vector
    - role of inner product
  - Soft Margin

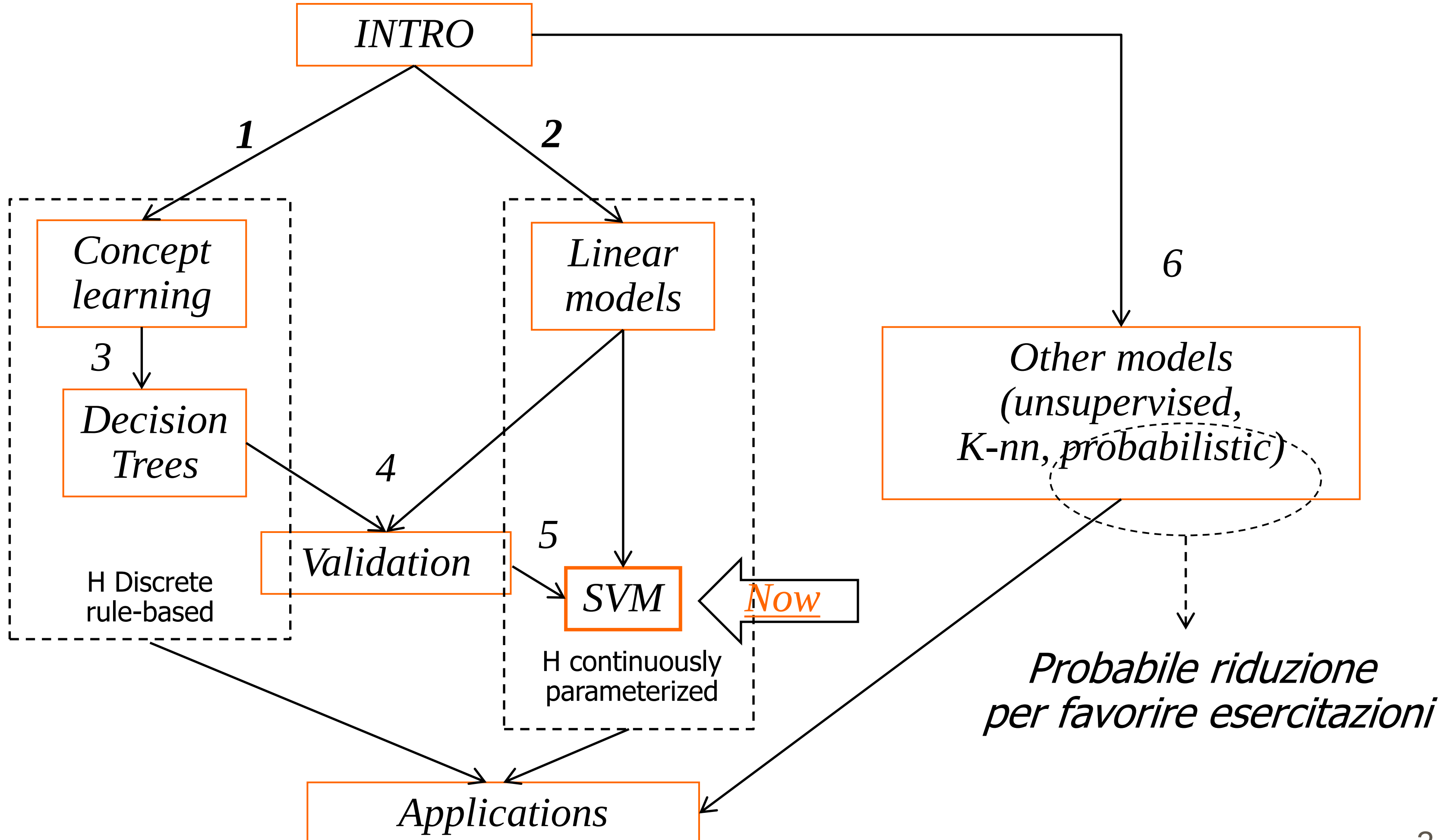
2) Kernel
- Mapping to High-Dimensional Spa

DRAFT, please do not circulate! *2023*

Course structure: preview

Dip. Informatica
University of Pisa

There is a structure in this intro to ML (bottom-up: starting from simple cases to view the concrete side of the general principles), collocating each lecture help you.



Support Vector Machine (SVM) intro

Dip. Informatica
University of Pisa

- A classifier derived from *statistical learning theory* by Vapnik, et al. (see past lectures)
- After years of theoretical developments, SVM became famous when, using images as input, it gave accuracy comparable to SotA neural-network (in the 90s) with hand-designed features in a handwriting recognition task
- Currently, SVM is widely used all the application fields of supervised learning
  - Also used for regression (will not cover today)
- Even if current Deep Neural Networks often outperforms them in many domain (e.g. image analysis)

WHAT IS SVM?

Dip. Informatica
University of Pisa

- This is a typical topic at the end of a full ML course (#ML)
  - Discussing SVM details requires a deeper view on ML
- On the other hand,

  you may be tempted to use it (e.g. popular sw tools)

So it anyway deserve to have a *first* (critical) *look at it*.

- After all, it is close to our view of *linear models* ...
- And an opportunity to see ta least 1 model at the state-of- the-art

Indeed, we restart from linear model for classification

And again, we are also interested in enrich it for non-linear problems

For a reduced version of the slides:
You can skip all the slides/parts with #tech label and dashed boxes
(or, better, read them later)

Our Aims for SVM intro

Dip. Informatica
University of Pisa

Objectives of this SVM intro in IIA are to show:

1) Control of the *model complexity* via optimization approach
   - to directly approximate the Structural risk minimization

   Max margin classifier

2) Use efficiently linear *basis expansion* via kernel
   - so we obtain another flexible approach for *non-linear* supervised learning

   Kernel

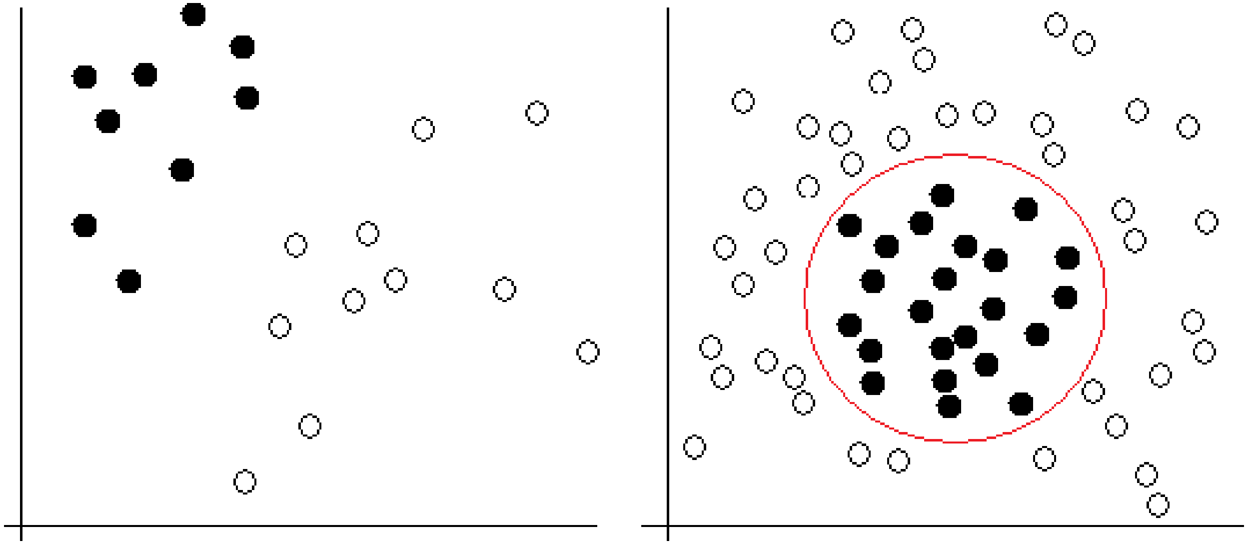
3) Avoid typical misinterpretations in the use of SVM
   - Such are overoptimistic beliefs on overfitting and course-of-dimensionatilty avoidance

   Practice

Repetita Linear vs Non-Linearly Separable

Dip. Informatica
University of Pisa

Exercise: which one is the non-linearly separable case (in the inputs)?

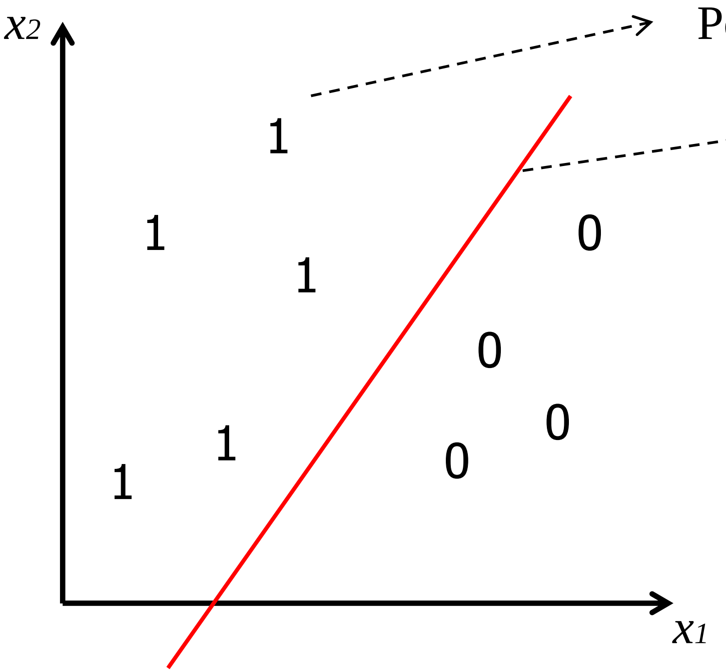


Repetita: Classification by linear decision boundary

The classification may be viewed as the allocation of the input space in decision regions (e.g. **0/1**)

Example: linear separator on

2-dim instance space $\mathbf{x}=(x_1,x_2)$ in R^2 , $f(\mathbf{x})=0/1$ (or $-1/+1$)



Point belonging to class 1

Separating (hyper)plane : $\mathbf{x}$ s.t.

$$\mathbf{w}^T\mathbf{x} + w_0 = w_1x_1 + w_2x_2 + w_0 = 0$$

Def

$$\boxed{h(\mathbf{x}) = \begin{cases} 1 & \text{\_}if \quad \mathbf{w}^T\mathbf{x} + w_0 \geq 0 \\ 0 & \text{\_\_\_\_\_\_\_}otherwise \end{cases}}$$

[0,1] outputs

or

Def

$$\boxed{h(\mathbf{x}) = sign(\mathbf{w}^T\mathbf{x} + w_0)}$$

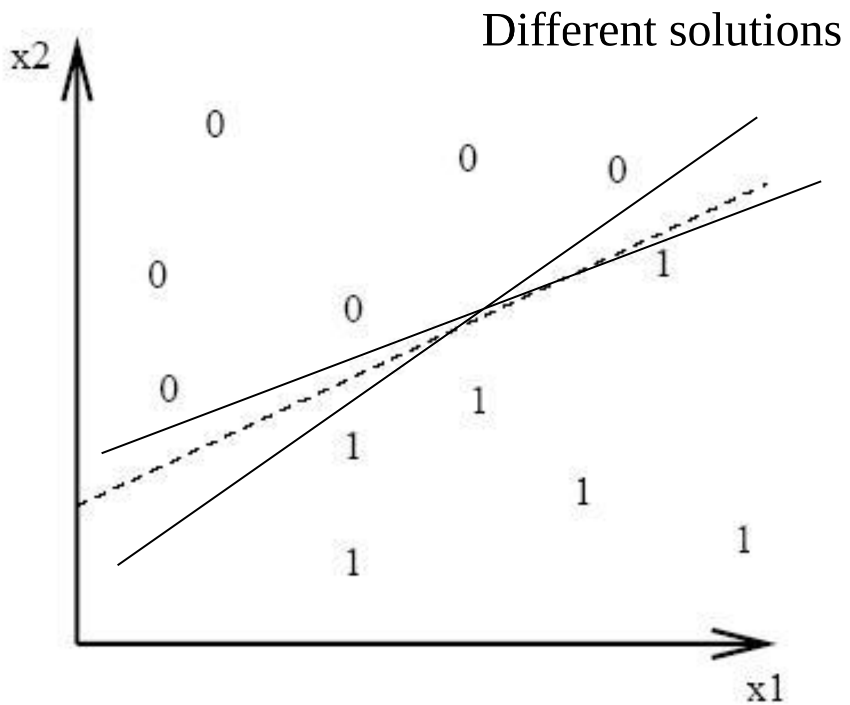
[-1,+1] outputs

Linear threshold unit (LTU)

How many? (H): set of dichotomies induced by hyperplanes

Multiple exact solutions

Dip. Informatica
University of Pisa



How many? (H): set of dichotomies induced by hyperplanes
For ML not all the solution are "equivalent"

Dip. Informatica
University of Pisa

AIM 1)

1) **Maximum margin classifier**

- Binary classification problem
- Let us start assuming a *linear separable* problem
- and assuming also *no noise* data

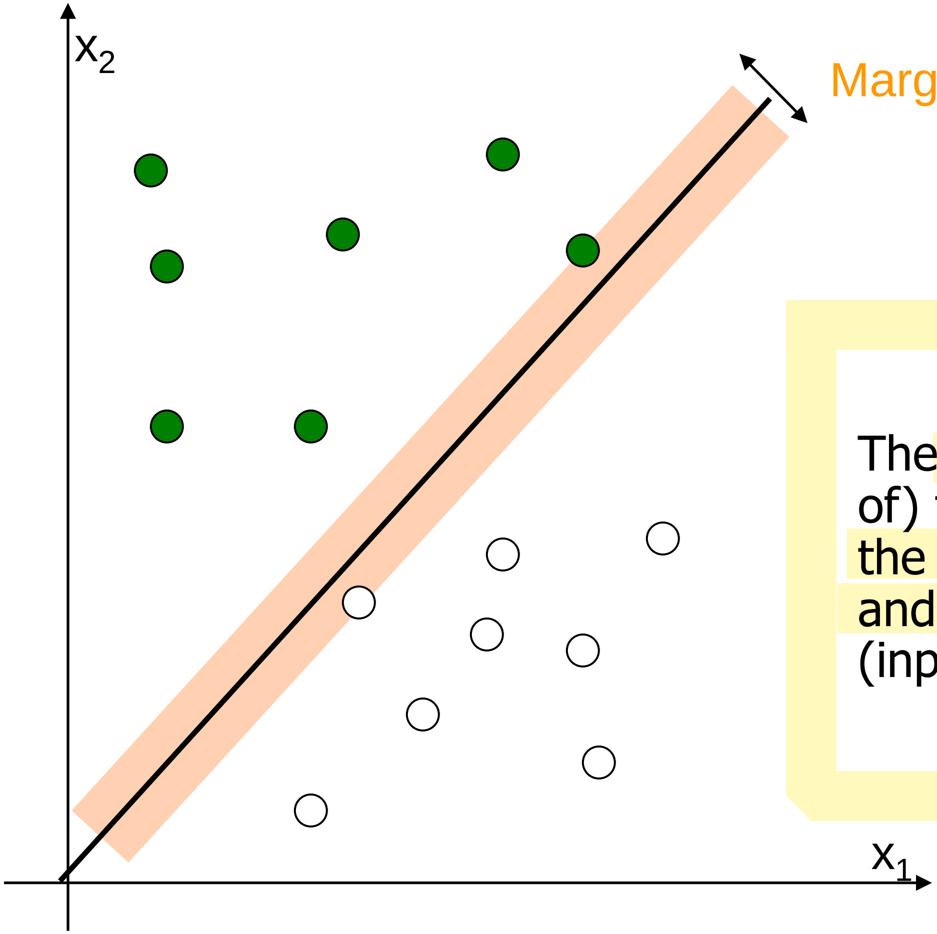
(hard margin SVM)

(we will release later these assumptions)

Margin examples

Dip. Informatica
University of Pisa

- Not all the hyperplanes solving the task are equals....
- Varying the separating hyperplane the margin varies as well.



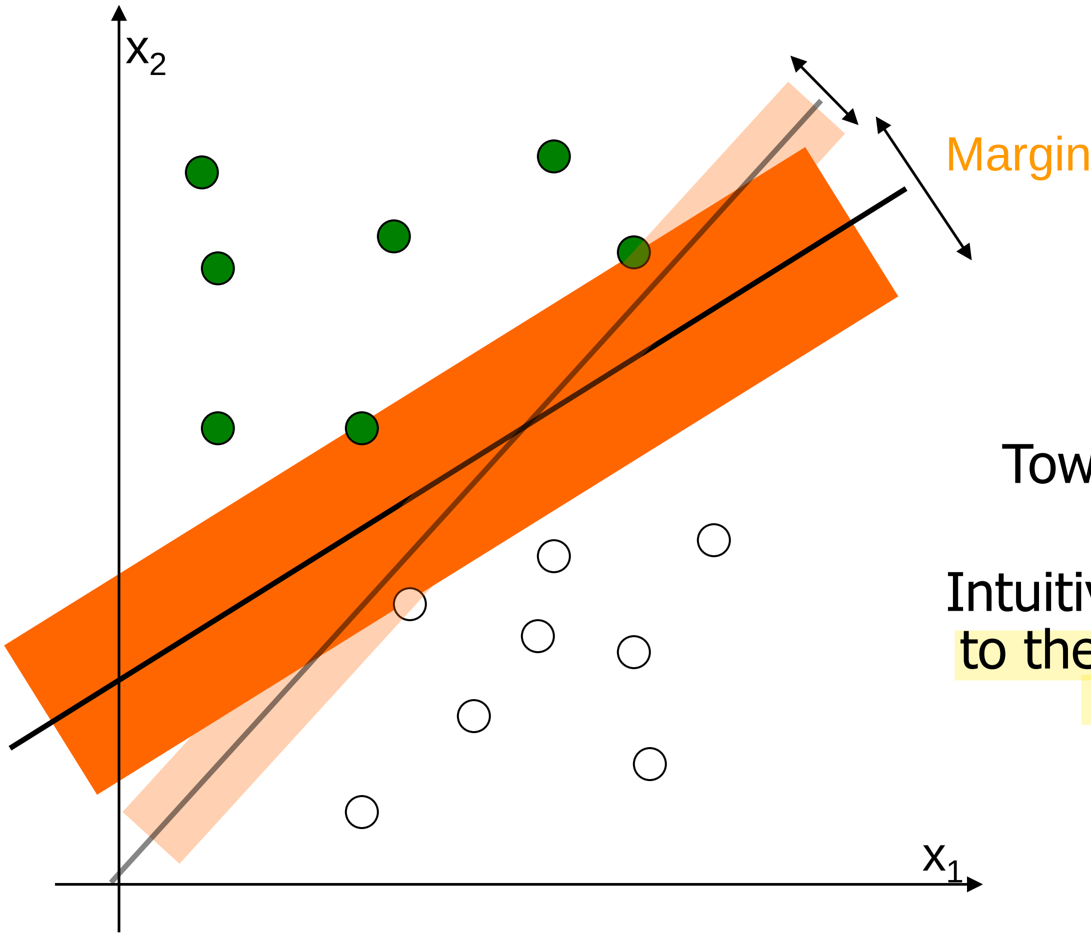
Def

The margin is (the double of) the distance between the separating hyperplane and the closest data points (input examples).

Margin examples

Dip. Informatica
University of Pisa

- Not all the hyperplanes solving the task are equals....
- Varying the separating hyperplane the margin varies as well.

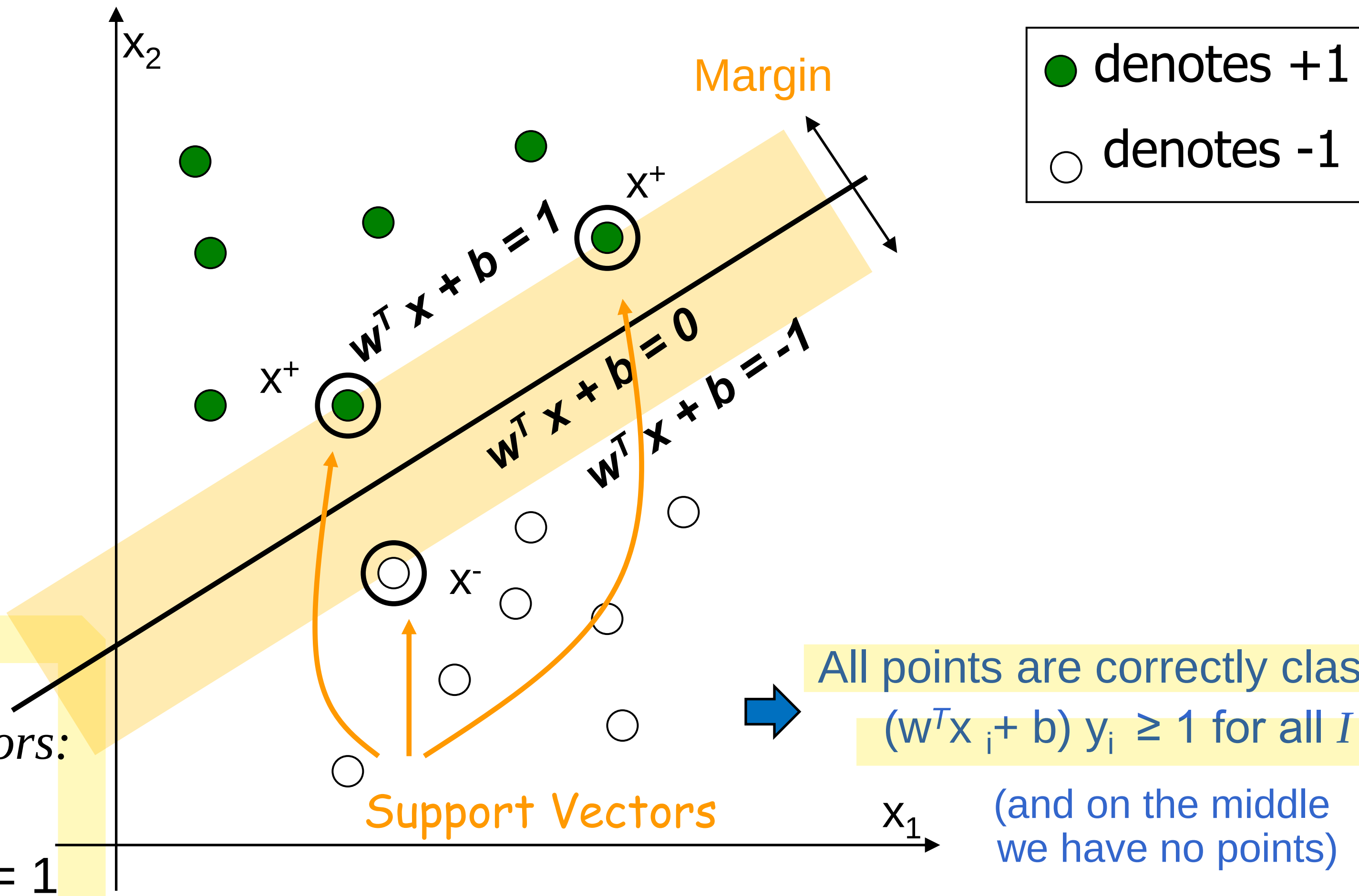


Toward max margin classifier …

Intuitively: max distance to the data points, max “safe zone”

Canonical representation of the hyperplane and SV

Dip. Informatica
University of Pisa



Def

Support Vectors:

$\boldsymbol{x}_p$ *s.t*

$|\boldsymbol{w}^T \boldsymbol{x}_p + b| = 1$

Note: $b = w_0$

All points are correctly classified if $(w^T x_i + b)\, y_i \geq 1$ for all l data

(and on the middle we have no points)

Toward max margin optimization problem

Dip. Informatica
University of Pisa

Consider the problem to learn a *linear model* for binary classification,

i.e. a function $h: \mathbb{R}^n \rightarrow \{-1,1\}$ $\quad h(\mathbf{x}) = sign(\mathbf{w}^T\mathbf{x} + b)$

based on examples $(\mathbf{x}_p, y_p)$ in TR set $\quad b \downarrow w_0$

Def | Training problem: find $(\mathbf{w}, b)$ such that all points are classified correctly and the *margin is maximized* ← ***new***

Canonical representation of the hyperplane **[see picture in the previous slide]:**

#tech

$(\mathbf{x}_p, y_p)$ is classified correctly for all p

$\leftrightarrow \quad \mathbf{w}^T\mathbf{x}_p + b \geq 0$ if $y_p = 1$ and $\mathbf{w}^T\mathbf{x}_p + b < 0$ if $y_p = -1$ for all p

w.l.o.g. (it is possible to rescale **w**, b (scaling freedom prop.) so that the closest points to the separating hyperplane satisfy $|\mathbf{w}^T\mathbf{x}_p + b| = 1$, i.e the *support vectors*)

$\mathbf{w}^T\mathbf{x}_p + b \geq 1$ if $y_p = 1$ and $\mathbf{w}^T\mathbf{x}_p + b \leq -1$ if $y_p = -1$ for all p

$\leftrightarrow \quad (\mathbf{w}^T\mathbf{x}_p + b)\, y_p \geq 1$ for all p ← constraints: all points are correctly classified

Note also that, differently from the LMS solution, for linear separable cases here we have 0 errors

Two Useful Facts

Dip. Informatica
University of Pisa

Def

- Margin $\propto 2/|\, \boldsymbol{w} \,|$ $[|\, \boldsymbol{w} \,|^2 = (\boldsymbol{w}^T\boldsymbol{w})$, the norm$]$ (#ML)

  maximize margin $\leftrightarrow$ minimize $|\, \boldsymbol{w} \,|$ $\leftrightarrow$ minimize $|\, \boldsymbol{w} \,|^2/2$

Def

- VC-dim of the SVM is inverse to the margin, i.e. it decreases with high margin (result by the theory, more at #ML)
  → control of model complexity by the margin (not strictly only on the input dim. as in standard linear models)
  → connect it to the SLT lecture

The optimal hyperplane is the hyperplane which *maximizes* the margin (and solves the training problem)

Hard margin SVM: Quadratic optimization problem

Dip. Informatica
University of Pisa

Def

Training problem: find $(\mathbf{w},b)$ such that all training points are classified correctly and the margin is maximized

Def

Training problem (primal form) :

minimize $|\mathbf{w}|^2/2$ (i.e. $\mathbf{w}^T\mathbf{w}$) — *Objective function*

such that $(\mathbf{w}^T\mathbf{x}_p + b)\, y_p \geq 1$ for all $p = 1..l$ — *Constraints*

quadratic optimization problem (sw packages)

- Note: Direct minimization of model complexity (optimization function),
- holding solution (0 errors) in the *constraints*
- Note: Objective function is convex in w
- *Exercise*: relate this primal form to the concepts in SLT VC-bounds
- *Exercise*: what is the meaning of the objective function and of the constraints?

Dual Problem

Dip. Informatica
University of Pisa

Dual formulation of training:

Maximize$_\alpha$ $\Sigma_i \alpha_i - \Sigma_{ij} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^t \mathbf{x}_j / 2$ $\quad i, j = 1..l$

with $\alpha_i \geq 0$, $\Sigma_i \alpha_i y_i = 0$

#tech

- Find optimal α_p $p=1..l$ (Lagrangian multipliers) by quadratic programming

#tech

- Convex → unique solution
- Computational cost scales with l (num. of data) instead of n (input dimension)

#tech

- Dual formulation (computing alpha values) allows us to show: support vectors and a special form of the solution → next slides

Solution: the classifier *h(x)*

Dip. Informatica
University of Pisa

- With the alfa (computed in the dual form) we can compute $(\boldsymbol{w},b)$

$$\boldsymbol{w} = \sum_p \alpha_p y_p \boldsymbol{x}_p \qquad p=1..l \qquad b = y_k - \boldsymbol{w}^T\boldsymbol{x}_k \quad \text{for any } \alpha_k > 0$$

Def

$$h(\boldsymbol{x}) = sign(\boldsymbol{w}^T\boldsymbol{x} + b) = sign(\sum_{p=1}^{l} \alpha_p y_p \boldsymbol{x}_p^T\boldsymbol{x} + b) = \boxed{sign(\sum_{p \in SV} \alpha_p y_p \boldsymbol{x}_p^T\boldsymbol{x} + b)}$$

Def

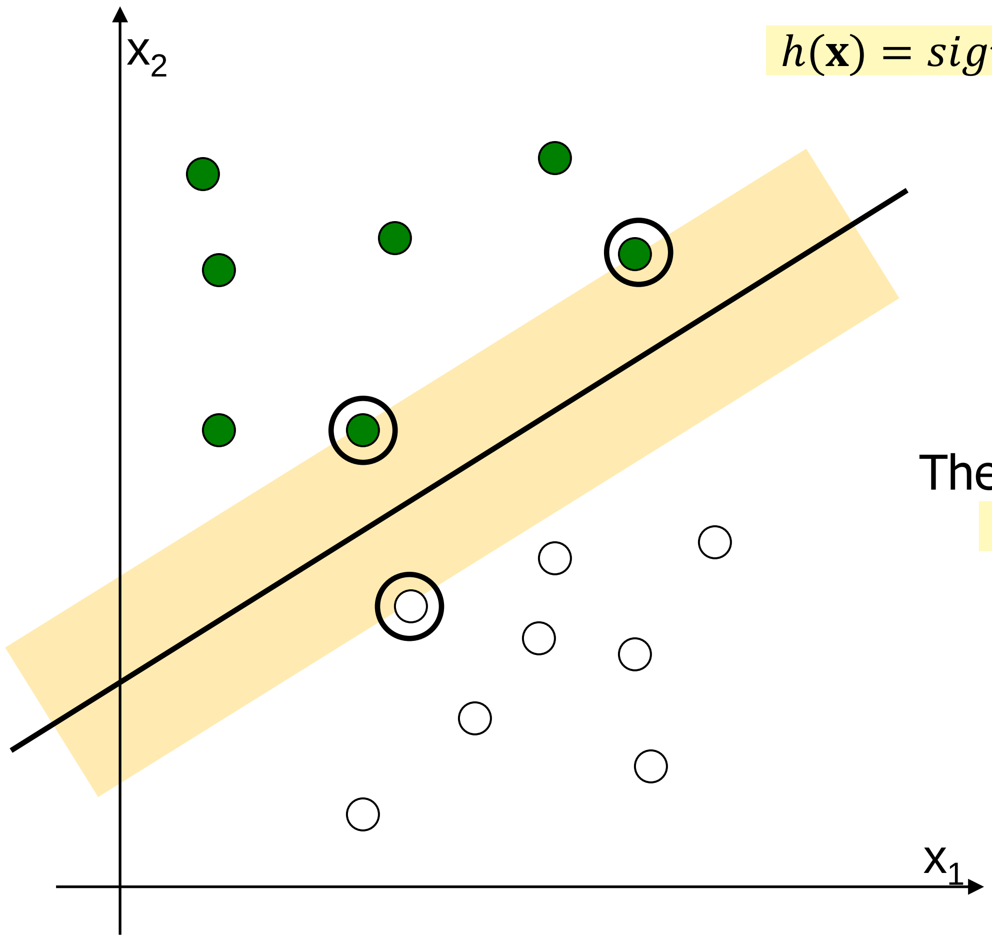
1) Alfa <>0 ↔ **support vectors** ($\alpha_p \neq 0 \rightarrow x_p$ is a support vector)

The solution is (often) sparse and formulated only in terms of the SVs
The hyperplane depends only from support vectors !

2) A special form of the solution: It is not even necessary to compute (w,b) explicitly to classify the points

Role of support vectors

Dip. Informatica
University of Pisa



$$h(\mathbf{x}) = sign\left(\sum_{p \in SV} \alpha_p y_p \, \mathbf{x}_p^T \mathbf{x} + b\right)$$

The hyperplane depends only from support vectors

Role of the inner product

Dip. Informatica
University of Pisa

- Data enter in form of dot products of pairs of points

$$h(\boldsymbol{x}) = sign\left(\sum_{p \in SV} \alpha_p y_p \boxed{\boldsymbol{x}_p^T \boldsymbol{x}} + b\right)$$

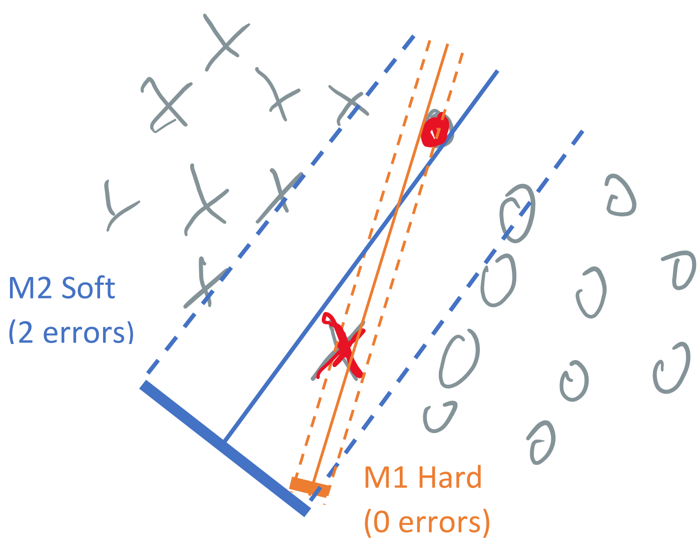
Dot product
among input patterns

Soft Margin

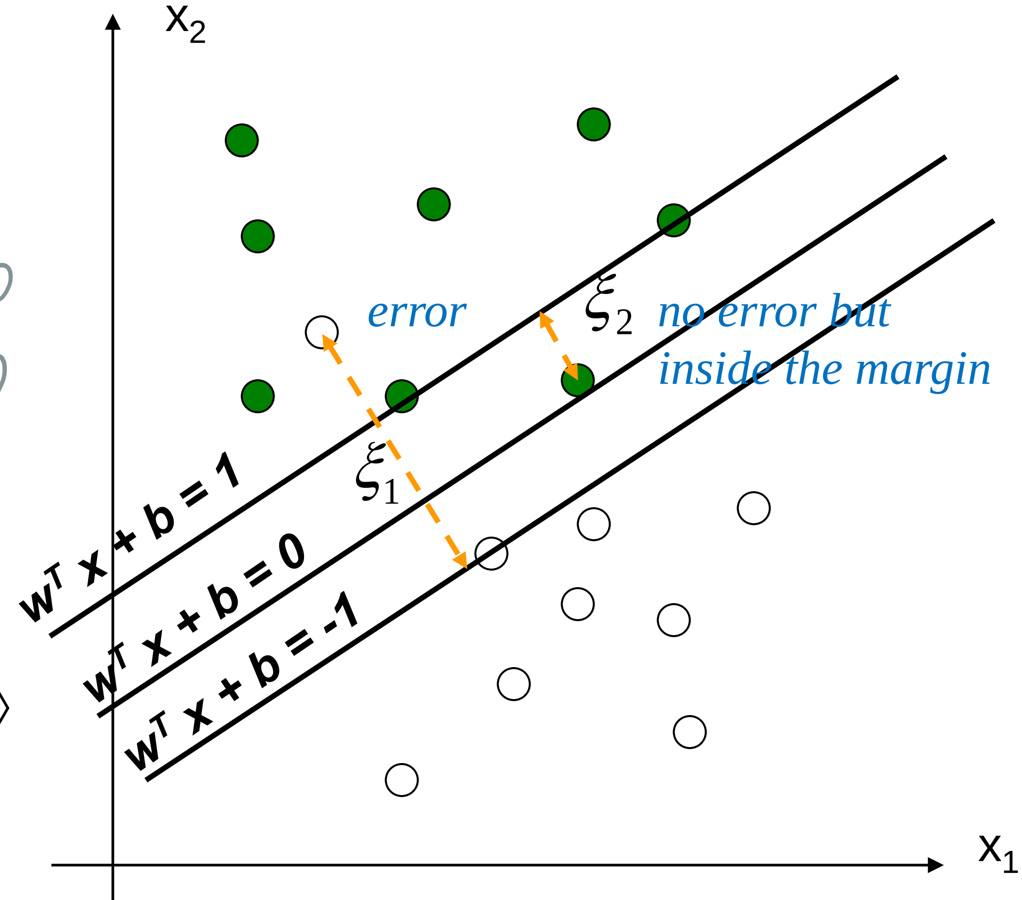
Hard margin for all the points may be too restrictive

Some errors can be allowed for *noise tolerance* and *to provide larger margin*

Solution: allow errors introducing slack-variables.



Slack variables ξ_i can be added to allow mis-classification of difficult or noisy data points



Soft Margin

Dip. Informatica
University of Pisa

Hard margin for all the points may be too restrictive

Some errors can be allowed for *noise tolerance and to provide larger margin*

Solution: allow errors introducing slack-variables.

Def

Primal training problem:

minimize $| \boldsymbol{w} |^2/2 + C \cdot \Sigma_p \xi_p$

such that $(\boldsymbol{w}\boldsymbol{x}_p + b)\, y_p \geq 1 - \xi_p$ and $\xi_p \geq 0$ for all p

positive ξ_p indicates an error or too small margin, C>0 guides the number of allowed errors

(the indices of the ξ_p are computed by the solver)

C: is a user defined hyperparameter (!)

Low C → many TR errors are allowed → possible underfitting

High C → no TR errors are allowed → small margin → possible overfitting

But we lost the *automatic* approximation of SRM of hard margin !

AIM 2)

Dip. Informatica
University of Pisa

Up to know only for linear separable problems…

And for non-linear cases?

Kernel

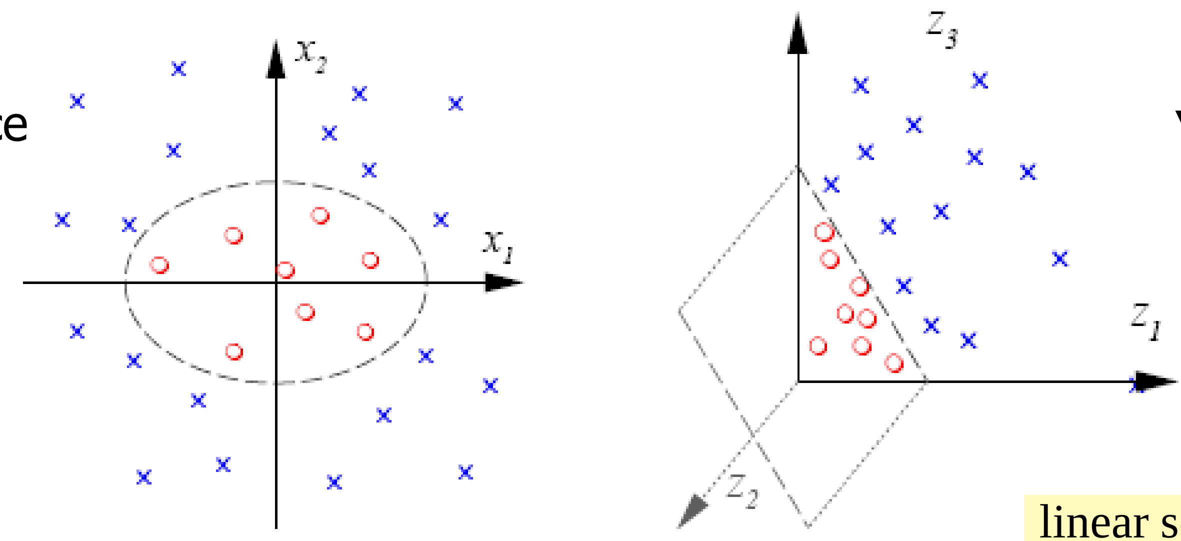
2) Use efficiently linear *basis expansion* via kernel
   - so we obtain another flexible approach for *non-linear* supervised learning

Mapping to a High-Dimensional Space

Dip. Informatica
University of Pisa

- Map the data points in the input space to a high-dimensional (so called in SVM) feature space, where they can be linearly separable

input space



"feature space"

linear separators here correspond to *non-linear* separators in original space.

$$\Phi : \mathbb{R}^2 \rightarrow \mathbb{R}^3$$

$$(x_1, x_2)^T \mapsto (x_1^2, \sqrt{2}x_1 x_2, x_2^2)^T$$

We already know it

Dip. Informatica
University of Pisa

- Using LBE $\Phi(\mathbf{x})$ instead of $\mathbf{x}$ to deal with non-linear tasks wrt to inputs
- However, we know that using high dimensional feature spaces (large basis function expansion) can be computationally unfeasible and more importantly can **easily** lead to *overfitting* without controlling the dimension of feature space and the *complexity* of the classier (complexity is in this case related to the input dimensionality, # of free parameters in the eq:)

$$h_{\mathbf{w}}(\mathbf{x}) = sign(\sum_k w_k \phi_k(\mathbf{x}) \ )$$

- We are going to propose the Kernel approach to manage (implicitly) the feature space in the context of regularized modeling (where the complexity is margin dependent, not strictly input dim. dependent):
- Thus, thanks to the automatized regularization of SVM, complexity of the classifier can be kept small regardless of dimensionality in the new feature space.

Use Φ(x) instead of x

Dip. Informatica
University of Pisa

- In SVM it is **not** necessary to compute ***w*** and
- the data enter in form of dot products of pairs of points

$$h(\mathbf{x}) = sign(\sum_{p \in SV} \alpha_p y_p \boxed{\mathbf{x}_p^T \mathbf{x}} + b)$$

$$\mathbf{w} = \sum_p \alpha_p y_p \mathbf{x}_p$$

$$h_{\mathbf{w}}(\mathbf{x}) = sign(\sum_k w_k \phi_k(\mathbf{x}) \ )$$

$$h(\mathbf{x}) = sign(\sum_{p \in SV} \alpha_p y_p \underbrace{\boxed{\phi^T(\mathbf{x}_p)\phi(\mathbf{x})}}_{Dot\ product})$$

and it is **not** even necessary to compute directly *phi*

Def

$$h(\mathbf{x}) = sign(\sum_{p \in SV} \alpha_p y_p \boxed{K(\mathbf{x}_p, \mathbf{x})})$$

- i.e. we can *implicitly manage* the feature space by a Kernel function

Kernel

Dip. Informatica
University of Pisa

#tech

A kernel k: $\mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ is a function such that some (possibly high dimensional) Hilbert space X and a function $\Phi:\mathbb{R}^n \rightarrow X$ exists with

$$k(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j) \qquad \boxed{\textit{Def}}$$

i.e. a kernel is a potential shortcut to compute the dot product efficiently also in high dimensional spaces

We use the K function to compute directly the dot products **in the feature space**

The example before can be efficiently computed in R^2 instead of R^3:

$$\Phi : \mathbb{R}^2 \rightarrow \mathbb{R}^3$$
$$(x_1, x_2)^T \mapsto (x_1^2, \sqrt{2}x_1 x_2, x_2^2)^T$$

$$\Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j)^2 = K(\mathbf{x}_i, \mathbf{x}_j)$$

Computed in 2-dim(not in 3-dim)

Dip. Informatica
University of Pisa

Well-know popular Kernels

Def • Linear: $K(\mathbf{x}_i,\mathbf{x}_j)= \mathbf{x}_i^T\mathbf{x}_j$

– Mapping Φ: $\mathbf{x} \rightarrow \phi(\mathbf{x})$, where $\phi(\mathbf{x})$ is $\mathbf{x}$ itself

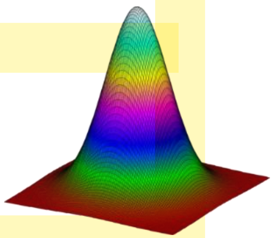
Def • Polynomial of power p: $K(\mathbf{x}_i,\mathbf{x}_j)= (1+ \mathbf{x}_i^T\mathbf{x}_j)^k$

– Mapping Φ: $\mathbf{x} \rightarrow \phi(\mathbf{x})$, where $\phi(\mathbf{x})$ has exponential dimension in k

Minus sign

Def • RBF (radial-basis function) Gaussian: $K(\mathbf{x}_i,\mathbf{x}_j) = e^{-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{2\sigma^2}}$

– Mapping Φ: $\mathbf{x} \rightarrow \phi(\mathbf{x})$, where $\phi(\mathbf{x})$ is *infinite-dimensional*



RBF is a very popular choice: note it has a hyperparameter (sigma)

Very flexible, it can be used to make decision boundary around each TR points!

Low sigma → narrow peaked Kernels → patterns are similar only if very close and the classifier replies with the class of the closest point

#tech

Can be very powerful on TR discrimination but of course prone to **overfitting**

Design of new kernel for special kind of data is a current research topic

SVM completed

Dip. Informatica
University of Pisa

- We choose the trade-off parameter C, and the kernel function K (and its hyper-parameters)
- solve the optimization problem to find alfa
  - Computational cost scales with l (num. of data) instead of n (feature space dimension)
  - Modularity: change just the Kernel (with the same solver)

#tech

- The final model

$$h(\mathbf{x}) = sign\Big(\sum_{p \in SV} \alpha_p y_p \, K(\mathbf{x}_p, \mathbf{x})\Big)$$

AIM 3)

Practice

3) Avoid misinterpretation

Dip. Informatica
University of Pisa

- Avoid typical misinterpretation in the use of SVM
  - Such are overoptimistic beliefs on overfitting and course-of-dim avoidance
- **Overfitting** can occur without a careful selection of SVM hyperparameters: C, kernel, kernel parameters ,….
- Implicit treatment of **High dim space** is in *the feature space* not in the input space (assuming we project therein significant inputs)
- **Validation** technique seen so far for model selection (e.g. C, kernel and kernel hyperparameters) and model evaluation should be used rigorously as well
- See guide SVM document for CV grid search suggestions

From LIBSVM guide

Dip. Informatica
University of Pisa

We propose that beginners try the following procedure first:

- Transform data to the format of an SVM software
- Conduct simple scaling on the data $\longrightarrow$ *See next slide*
- Consider the RBF kernel $K(x, y) = e^{-\gamma\|x-y\|^2}$
- Use cross-validation to find the best parameter C and γ
- Use the best parameter C and γ to train the whole training set
- Test *On a separate external set*

$\gamma = 1/(2\sigma^2)$

gamma=1/(2*sigma^2)

VALIDATION SET (from TR set)!!!

See next slide

Details

Dip. Informatica
University of Pisa

- Data preprocessing:
  - {red, green, blue} → (0,0,1), (0,1,0), (1,0,0) [1-of-k]
  - Linear scaling continuous values in range [-1,+1] or [0,1]
  - [e.g. v-min/(max-min)]

- Model selection grid-search : e.g. C and gamma in RBF
  - With a table on the combinations of all the possible growing (exponential) values too find good intervals, e.g.:

$$C = 2^{-5}, 2^{-3}, \ldots, 2^{15}, \gamma = 2^{-15}, 2^{-13}, \ldots, 2^{3}$$

  - Then a finer (nested) grid-search can be performed

Conclusion

Dip. Informatica
University of Pisa

- SVM is very useful and popular advanced ML tool
  - Performance: often good, but *not always* the comparison is favorable w.r.t. other ML methods (NN and others).
- Coming from theory (SRM) is a good way to provide new ML approaches
- Combing the use efficiently linear *basis expansion* via **kernel** within a **max margin** approach allow to combine flexible models and the control of complexity
- Modularity of the kernel open new possibilities

- Avoid to think SVM outside of the validation needs!

Bibliography

Dip. Informatica
University of Pisa

- AIMA , ed .3: **chap 18.9**
- A Practical Guide to Support Vector Classiffication

  http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf
  (related to LIBSVM)

Others

- Haykin. Neural Networks: A Comprehensive Foundation (2nd/ Edition) -chapter 6 (SVM)
- OPPURE nel S. Haykin: Neural Networks and Learning Machines, Prentice Hall; (3rd Edition, 2008)
- Paper: Muller, Mika, Ratsch, Tsuda, Scholkopf. An introduction to kernel-based learning algorithms. IEEE Trans. Neural Networks, 12(2):181-201, May 2001

Bibliography (others)

To have a very deep view of SVM and SLT (not needed!):

- V.N. Vapnik, An Overview of Statistical Learning Theory, IEEE Transactions on Neural Networks, 10 (5), 1999.
- C.J.C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition
- B. Scholkopf. Statistical Learning and Kernel Methods

BOOKS

- Shawe-Taylor, Cristianini: An introduction to Support Vector Machines Cambridge U.P., 2000
- Shawe-Taylor, Cristianini: Kernel Methods for Pattern Analysis. Cambridge U.P. 2004
- Scholkopf, Smola: Learning with kernels. MIT Press. Cambridge, MA, 2002.
- (The SLT bible) Vapnik. Statistical Learning Theory. Wiley, N.Y., 1998.
- (Compact form) Vapnik. The Nature of Statistical Learning Theory. Springer, N.Y., 1995

DRAFT, please do not circulate!

For information

http://www.di.unipi.it/~micheli/DID

Alessio Micheli

micheli@di.unipi.it

https://ciml.di.unipi.it/

Dipartimento di Informatica
Università di Pisa - Italy

Computational Intelligence & Machine Learning Group