Introduction to Machine Learning

Alessandro Moschitti

Department of Computer Science and Information Engineering
University of Trento
Email: moschitti@disi.unitn.it

Workshop on Machine Learning for Software Construction
Isola 2011
Vienna, October 17-18

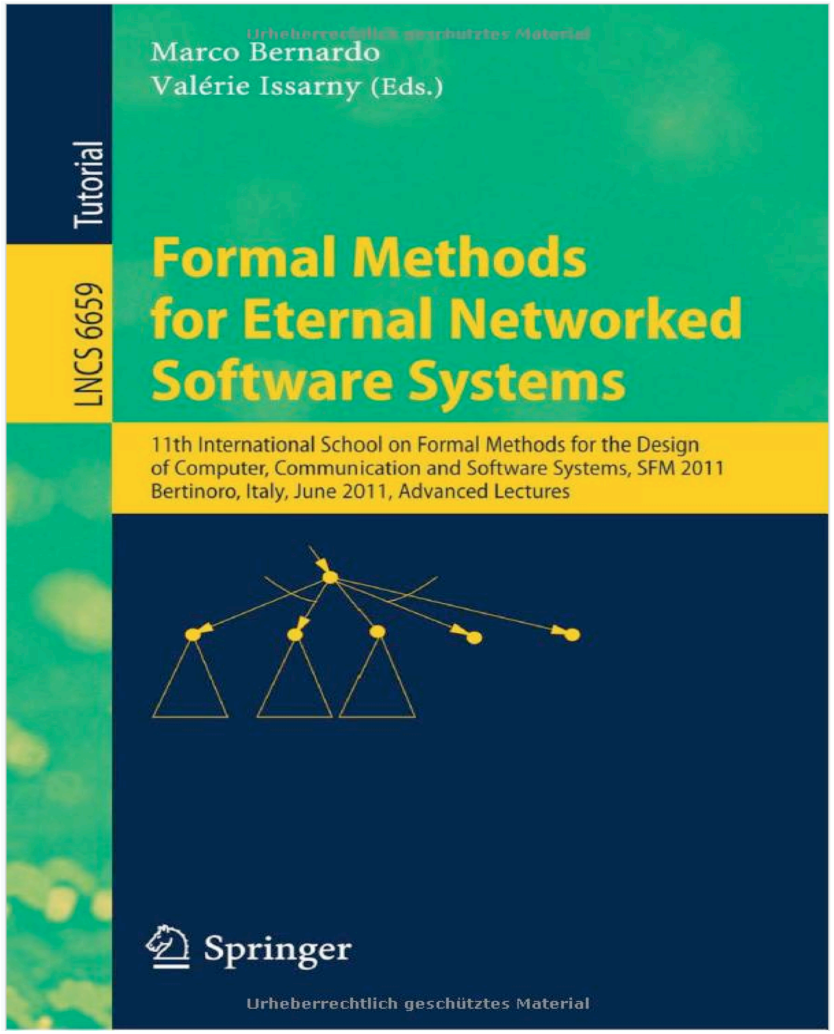
Outline

- What is Machine Learning?
- Abstract view of statistical learning theory
  - Training
  - Testing
  - Kernel functions
- Vector Space Model for text categorization
- Support Vector Machines (SVMs)
- Phases of applied Machine Learning
- Ranking and Regression
- Accuracy, Precision, Recall and F1

Practical Exercise with Aliaksei

- Automated Text Categorization
- Structural Kernels
  - Question Classification
- Classification of XML files of Web Services Description Language (WSDL)

Referring Material



Alessandro Moschitti,
Kernel-based Machines for Abstract and Easy Design of Automatic Learning
Part VI, Pag 458

Why Learning Functions Automatically?

- Anything is a function
  - From the planet motion
  - To the input/output actions in your computer
- Any problem would be automatically solved

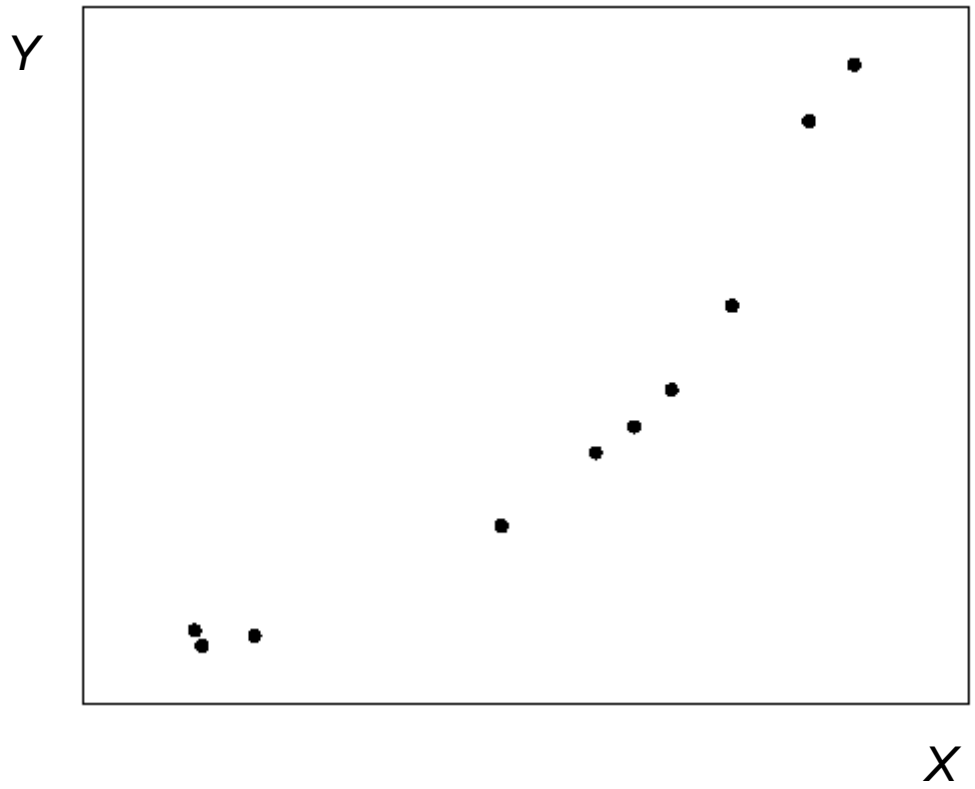
More concretely

- Given the user requirement (input/output relations) we write programs
- Different cases typically handled with *if-then* applied to input variables
- What happens when
  - millions of variables are present and/or
  - values are not reliable (e.g. noisy data)
- Machine learning writes the *program* (rules) for you

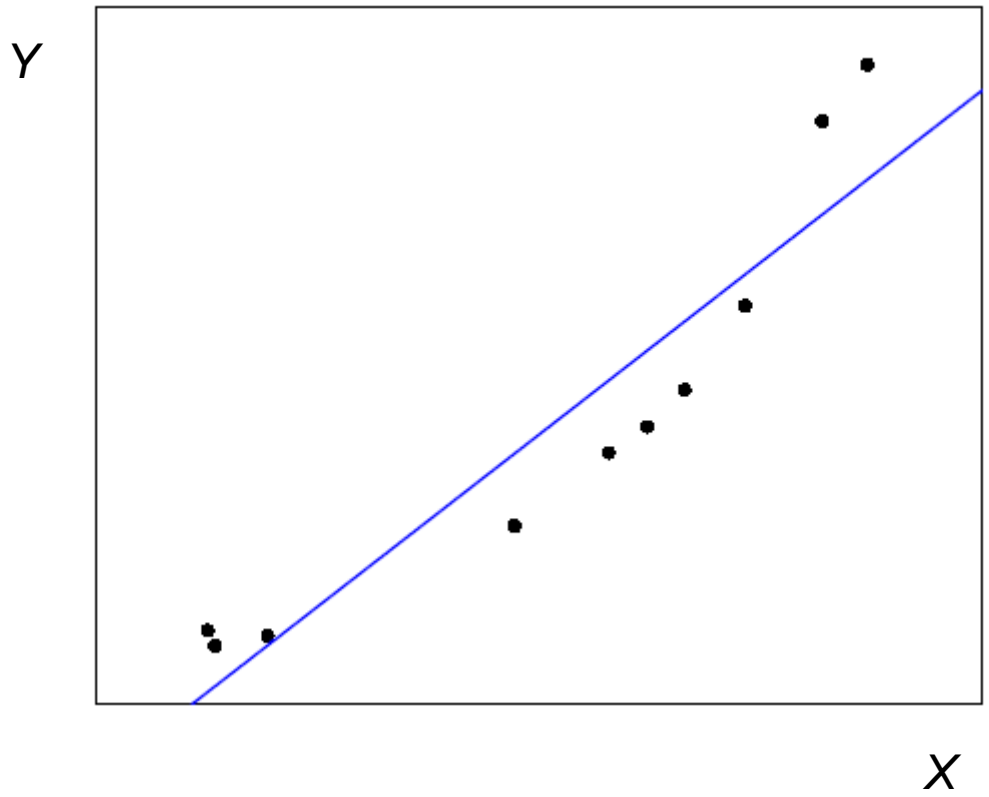
What is Statistical Learning?

- Statistical Methods – Algorithms that learn relations in the data from examples
- Simple relations are expressed by pairs of variables: $\langle x_1, y_1 \rangle$, $\langle x_2, y_2 \rangle$,…, $\langle x_n, y_n \rangle$
- Learning f such that evaluate y^* given a new value x^*, i.e. $\langle x^*, f(x^*) \rangle = \langle x^*, y^* \rangle$

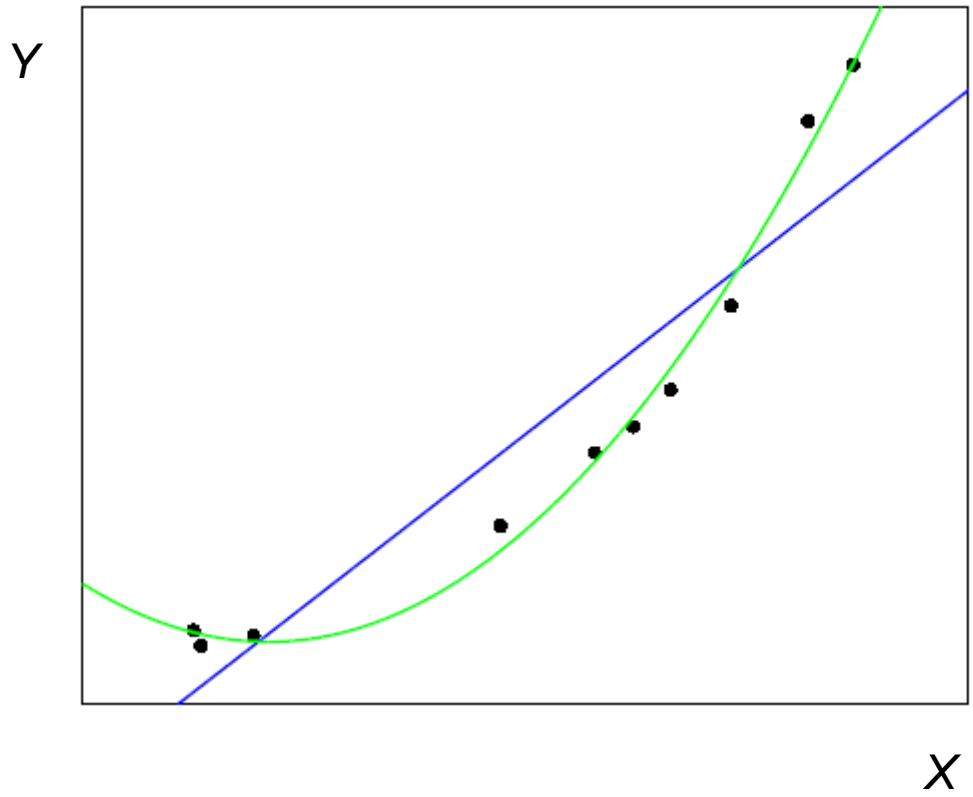
You have already tackled the learning problem



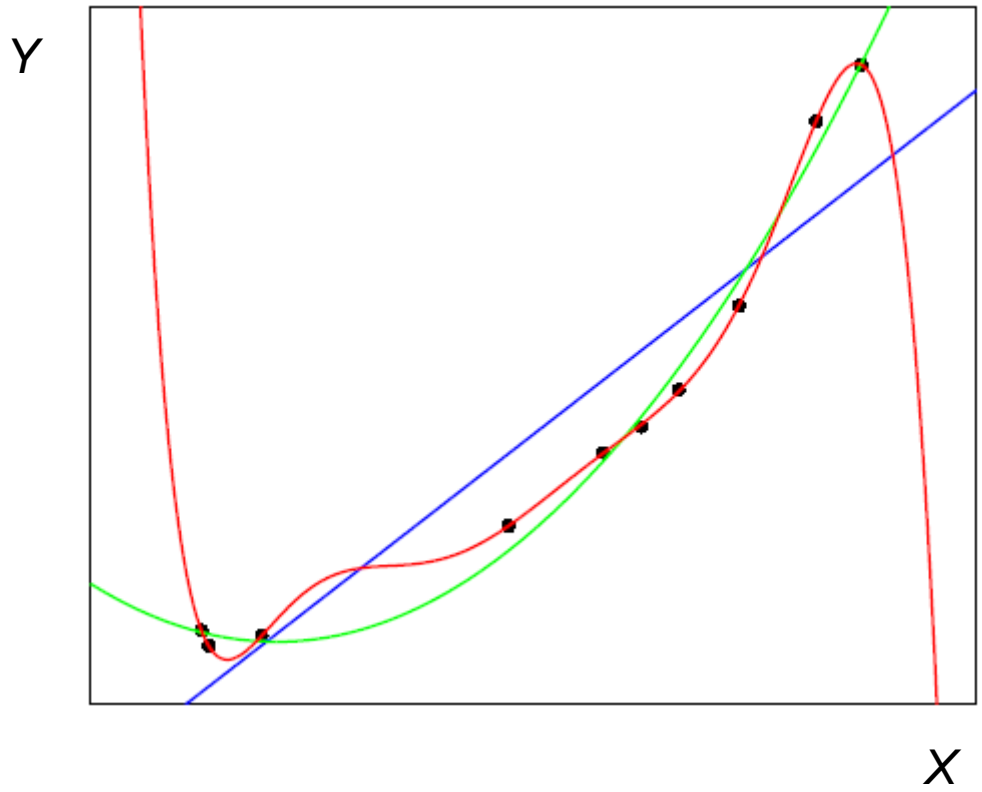
Linear Regression



Degree 2



Degree



Machine Learning Problems

- Overfitting
- How dealing with millions of variables instead of only two?
- How dealing with real world objects instead of real values?

Learning Models

- Real Values: *regression*
- Finite and integer: *classification*
- Binary Classifiers:
  - 2 classes, e.g.

    f(x) → {cats,dogs}

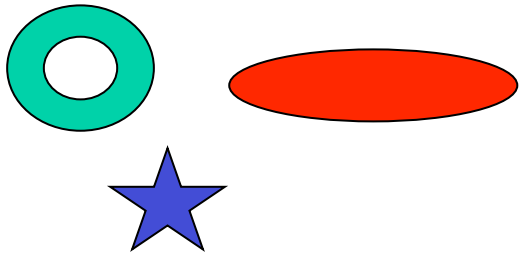
Abstract View of Statistical Learning Theory

Main Ingredients of Statistical Learning

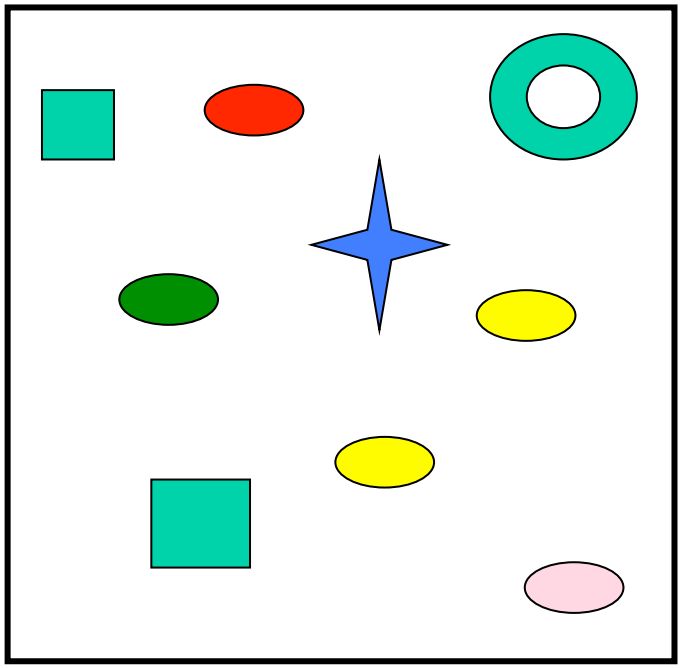
- Training set
  - Set of objects associated with a label
- Similarity function between the domain objects
- A learning algorithm
  - loss function tells the algorithm if it is doing well

Example based Classifiers

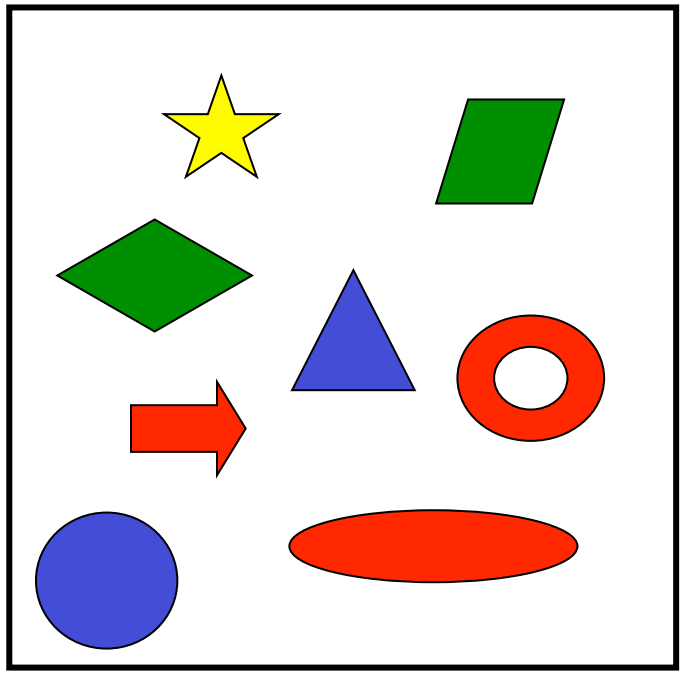
Objects to be classified:



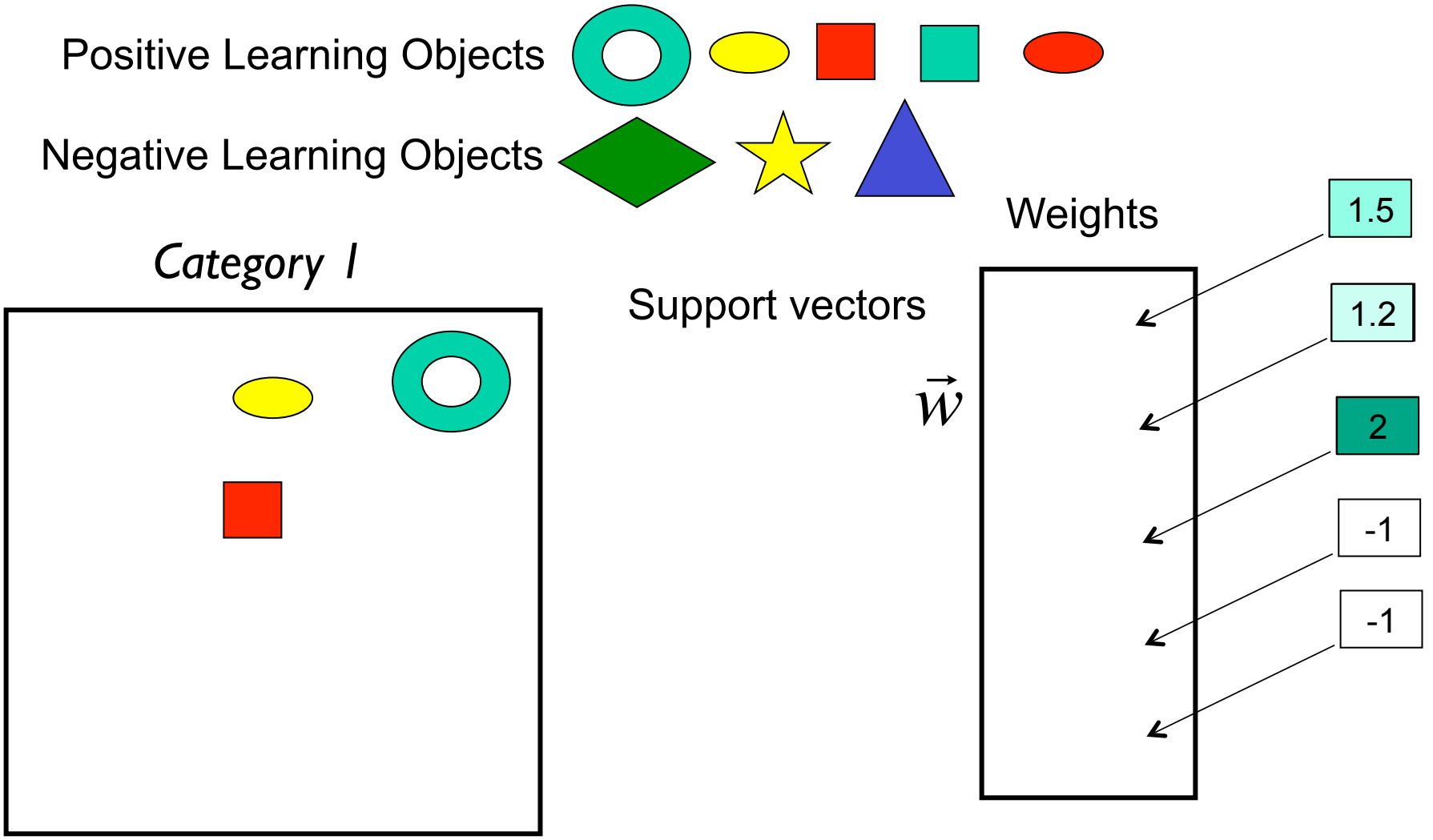
Category 1



Category 2



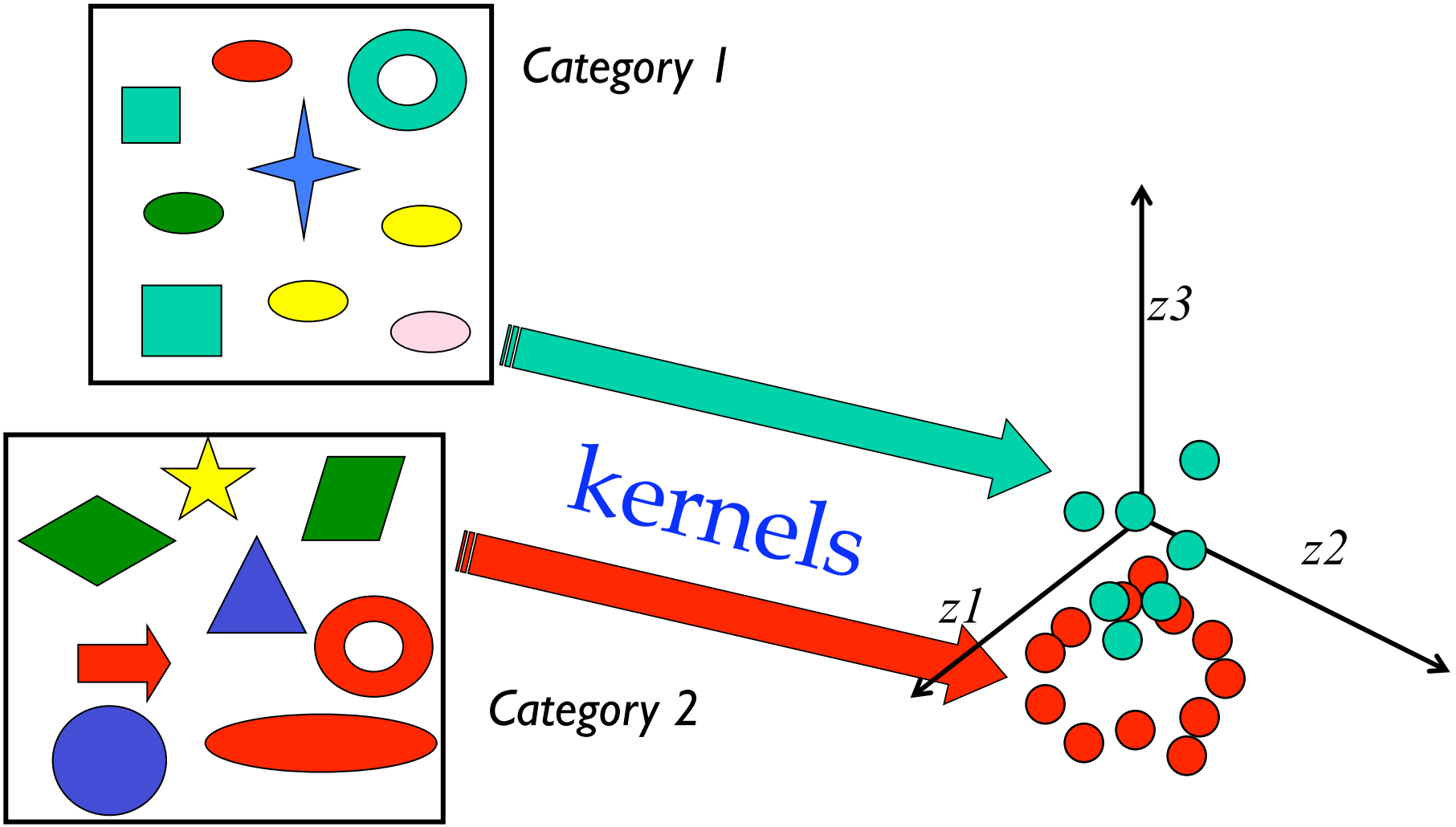
Learning phase



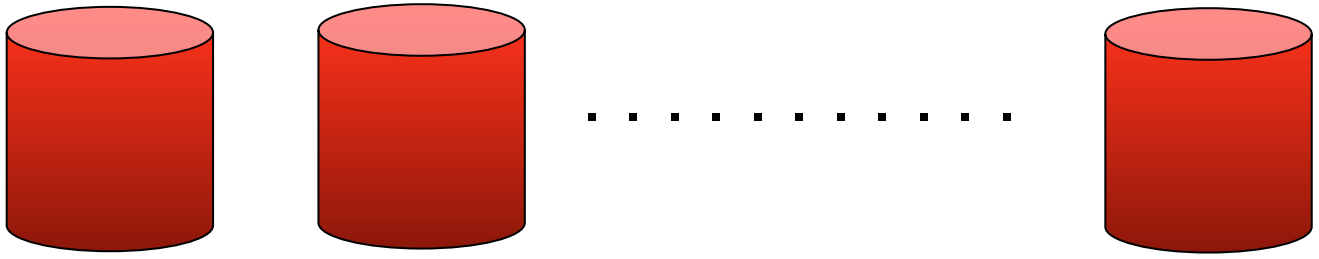
Similarity in Statistical Learning Theory

- Similarity is intuitively useful to learn and implement the classification function
- NB: *This does not lead to heuristic models*
- In statistical learning theory valid similarities are called ***Kernel Functions***
  - Kernels map examples in vector spaces
  - Examples are classified based on geometric properties
- Formally proved upperbound to the system error

In other words



A Simple Example: Text Categorization



Text Classification Problem

- Given: $C = \{C^1, \ldots, C^n\}$
  - a set of target categories:
  - the set *T* of documents,

  define $f : T \rightarrow 2^C$

Learning by counting words in common

- To learn and classify let us use a simple similarity function:
  - $S(d_1,d_2)$="number of words shared by the documents"
  - = $L_1 \cap L_2$, where L_1 and L_2 are the word lists of the two documents, respectively
- An efficient algorithm is list merging
- Interestingly, S can be evaluated as scalar product between two vectors

Word Vectors

- Each document is associated with a vector of *n* word types (i.e., *n* is the size of the vocabulary)

$$\text{Index}: \quad 1,..,23,.....,2560,.....,23456,.....,42546,..,62325,..,n$$

$$\vec{x} = (0,..,1,..,0,..,0,..,1,..,0,..,0,..,1,..,0,..,0,..,1,..,0,..,1,0,...,0)$$

acquisition buy market sell stocks

$$\text{Index}: \quad 1,..,20,.....,2560,.....,23456,.....,42560,..,62325,..,n$$

$$\vec{z} = (0,..,1,..,0,..,0,..,1,..,0,..,0,..,1,..,0,..,0,..,1,..,0,..,1,0,...,0)$$

acquired buy market sold stocks

- The dot product $\vec{x} \cdot \vec{z}$ counts the number of features in common, e.g., 3

The Vector Space Model (VSM)

d_1: Politic

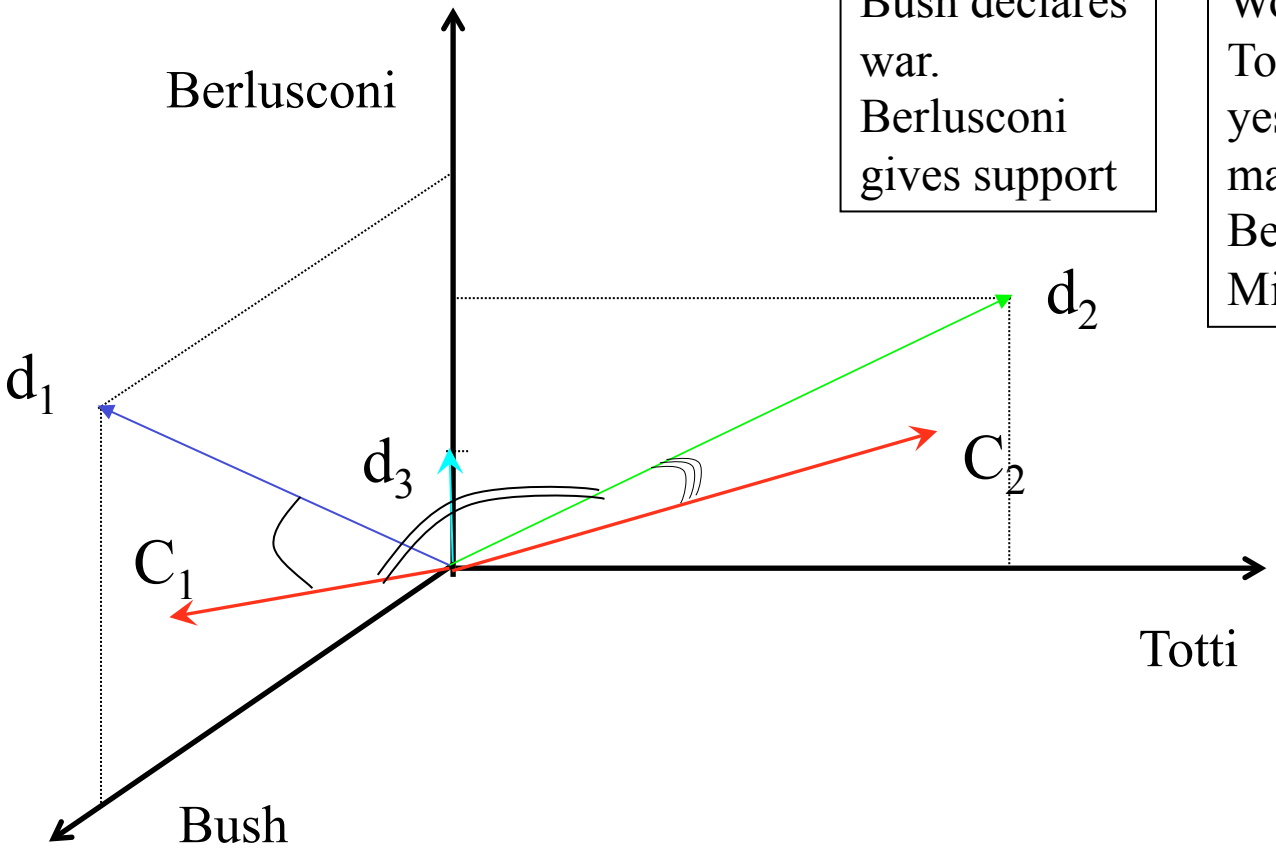
Bush declares war. Berlusconi gives support

d_2: Sport

Wonderful Totti in the yesterday match against Berlusconi's Milan

d_3:Economic

Berlusconi acquires Ibrahimović before elections



C_1 : Politics Category

C_2 : Sport Category

Summary of VSM

- VSM (Salton89')
  - Features are dimensions of a Vector Space
    Linear Kernel
  - Documents and Categories are vectors of feature weights.
  - *d* is assigned to C^i if $\vec{d} \cdot \vec{C}^i > th$
- Changing symbols

$$\vec{w} \cdot \vec{x} - th > 0 \Rightarrow \vec{w} \cdot \vec{x} + b > 0$$

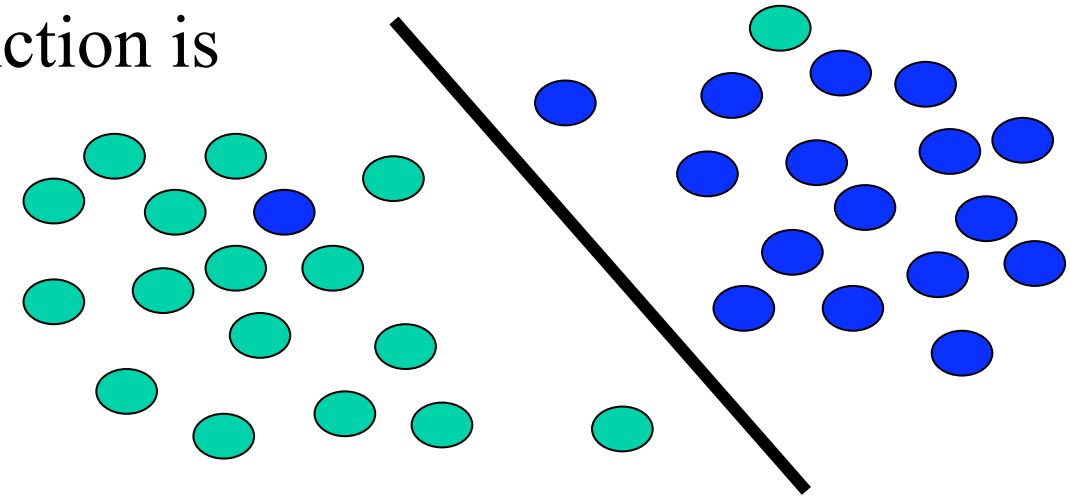
Linear Classifier

- The equation of a hyperplane is

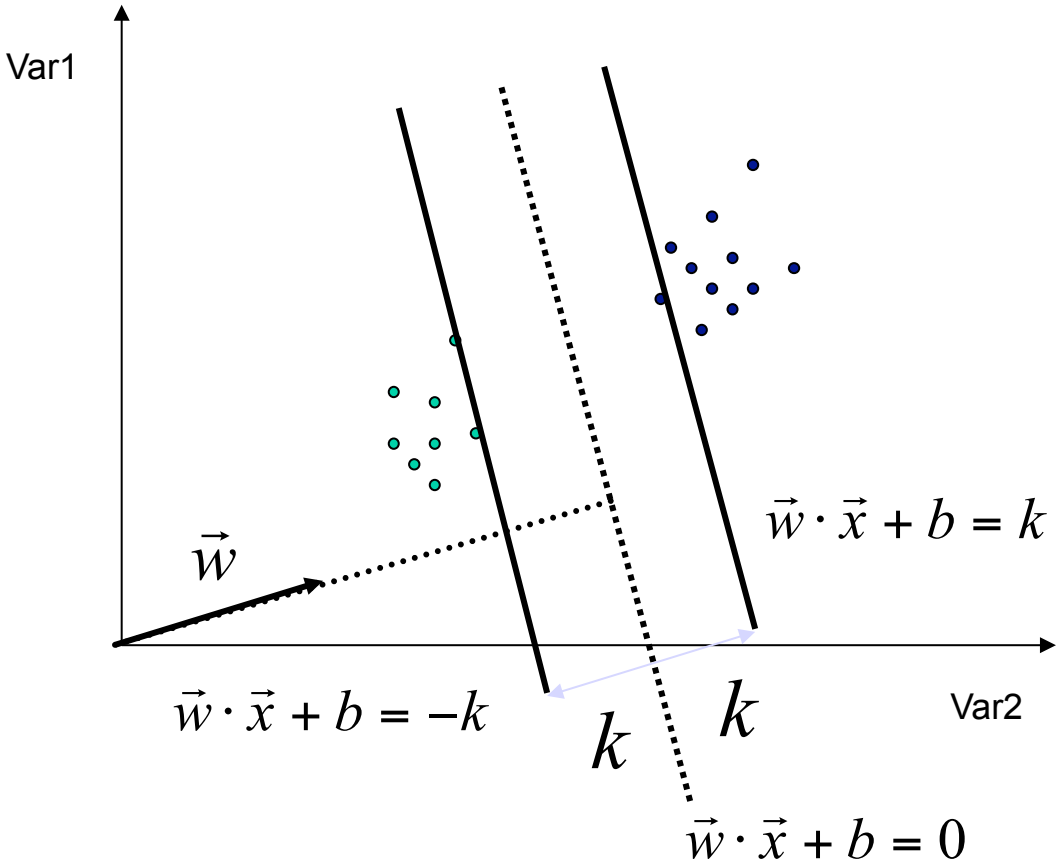
$$f(\vec{x}) = \vec{x} \cdot \vec{w} + b = 0, \quad \vec{x}, \vec{w} \in \Re^n, b \in \Re$$

- $\vec{x}$ is the vector representing the classifying example
- $\vec{w}$ is the gradient to the hyperplane
- The classification function is

$$h(x) = \text{sign}(f(x))$$

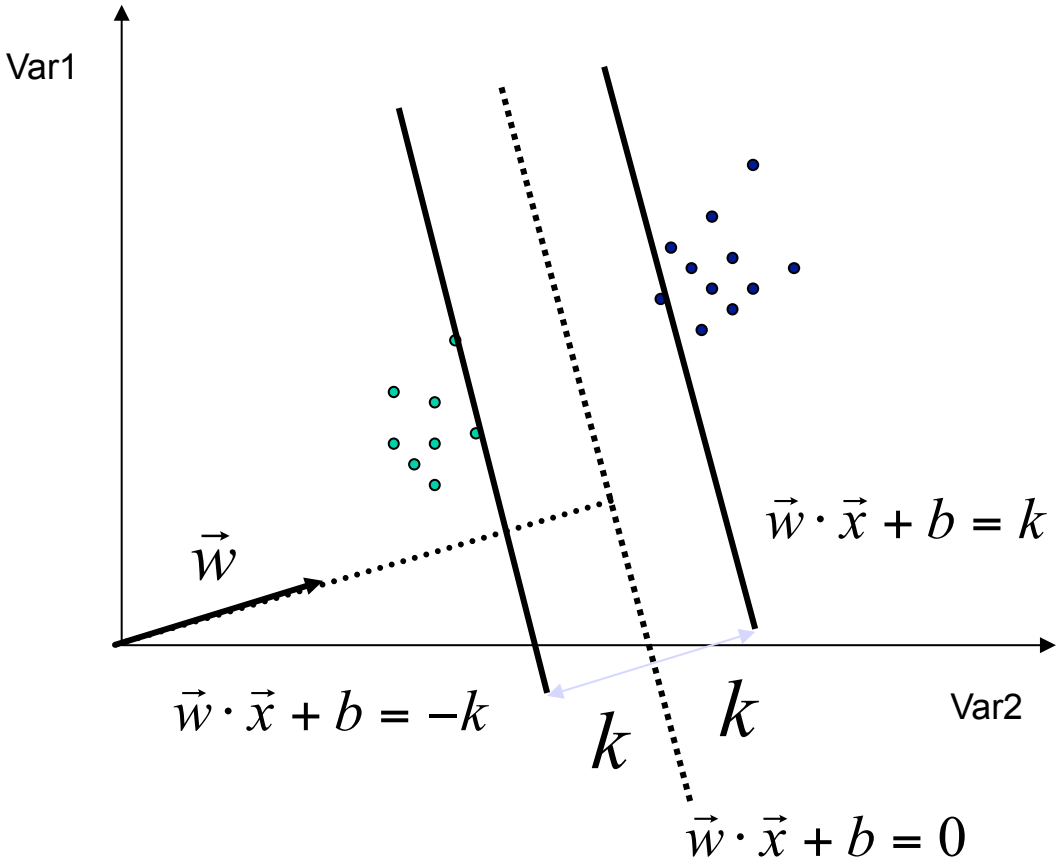


Learning the SVM hyperplane



The margin is equal to $\frac{2|k|}{\|w\|}$

Learning the SVM hyperplane



The margin is equal to $\frac{2|k|}{\|w\|}$

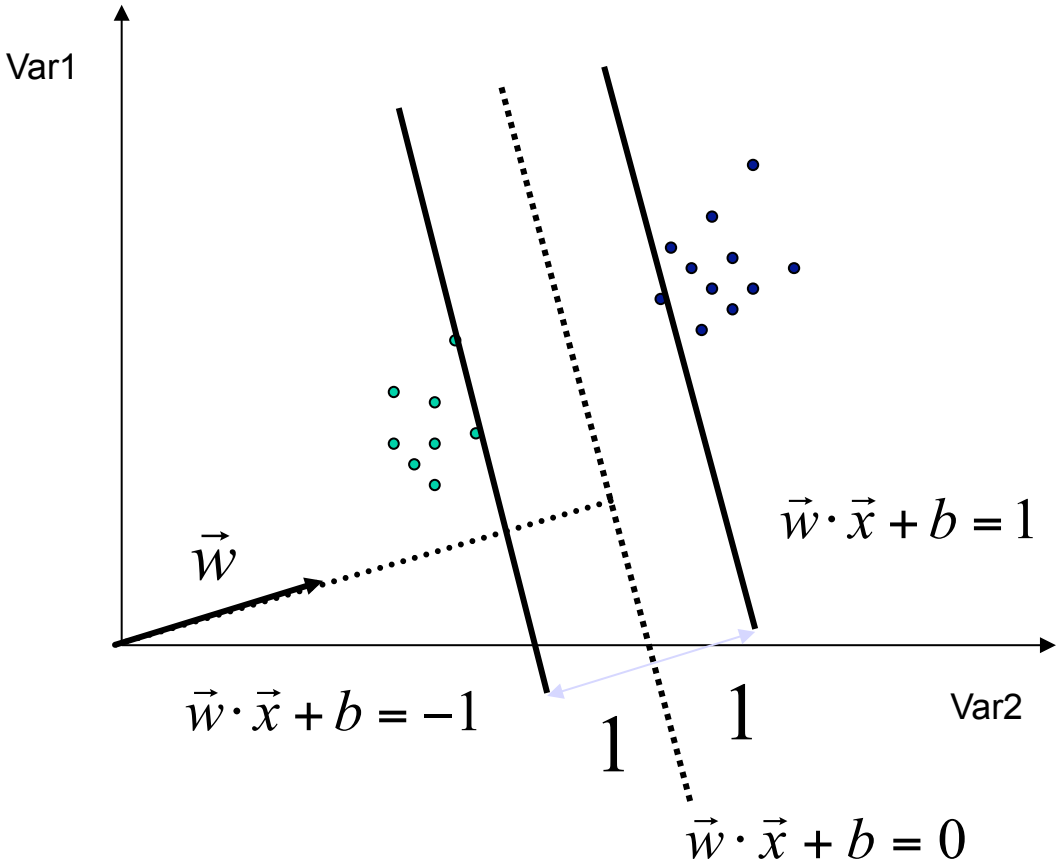
We need to solve

$$\max \frac{2|k|}{\| \vec{w} \|}$$

$$\vec{w} \cdot \vec{x} + b \geq +k, \quad \text{if } \vec{x} \text{ is positive}$$

$$\vec{w} \cdot \vec{x} + b \leq -k, \quad \text{if } \vec{x} \text{ is negative}$$

Learning the SVM hyperplane



There is a scale for which *k=1*.

The problem transforms in:

$$\max \frac{2}{\| \vec{w} \|}$$

$$\vec{w}\cdot\vec{x}+b \geq +1, \quad \text{if } \vec{x} \text{ is positive}$$

$$\vec{w}\cdot\vec{x}+b \leq -1, \quad \text{if } \vec{x} \text{ is negative}$$

Optimization Problem

- Optimal Hyperplane:
  - Minimize $\tau(\vec{w}) = \frac{1}{2}\|\vec{w}\|^2$
  - Subject to $y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1, i = 1,\ldots,m$
- The dual problem is simpler

Dual Transformation

- Given the Lagrangian associated with our problem

$$L(\vec{w}, b, \vec{\alpha}) = \frac{1}{2}\vec{w} \cdot \vec{w} - \sum_{i=1}^{m} \alpha_i [y_i(\vec{w} \cdot \vec{x}_i + b) - 1]$$

- To solve the dual problem we need to evaluate:

$$\theta(\vec{\alpha}, \vec{\beta}) = inf_{w \in W} \; L(\vec{w}, \vec{\alpha}, \vec{\beta})$$

- Let us impose the derivatives to 0, with respect to $\vec{w}$

$$\frac{\partial L(\vec{w}, b, \vec{\alpha})}{\partial \vec{w}} = \vec{w} - \sum_{i=1}^{m} y_i \alpha_i \vec{x}_i = \vec{0} \quad \Rightarrow \quad \vec{w} = \sum_{i=1}^{m} y_i \alpha_i \vec{x}_i$$

Kernels in Support Vector Machines

- In Soft Margin SVMs we maximize:

$$\sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y_i y_j \alpha_i \alpha_j \left(\vec{x_i} \cdot \vec{x_j} + \frac{1}{C} \delta_{ij} \right)$$

- By using kernel functions we rewrite the problem as:

$$\begin{cases} maximize \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y_i y_j \alpha_i \alpha_j \left(k(o_i, o_j) + \frac{1}{C} \delta_{ij} \right) \\ \alpha_i \geq 0, \quad \forall i = 1, .., m \\ \sum_{i=1}^{m} y_i \alpha_i = 0 \end{cases}$$

Dual Representation for Classification

- Given the relation we found:

$$\vec{w} = \sum_{j=1..\ell} \alpha_j y_j \vec{x}_j$$

- the classification function will be:

$$\text{sgn}(\vec{w} \cdot \vec{x} + b) = \text{sgn}\left(\sum_{j=1..\ell} \alpha_j y_j \vec{x}_j \cdot \vec{x} + b \right)$$

- So we can apply a general kernel function

$$\text{sgn}(\vec{w} \cdot \vec{x} + b) = \text{sgn}\left(\sum_{j=1..\ell} \alpha_j y_j k(o_j \cdot o) + b \right)$$

Phases of applied ML

- Data pre-processing (e.g. tokenization, stemming)
- Feature Selection (optionally)
  - Document Frequency, Information Gain, χ_2 , mutual information,...
- Feature weighting
  - for examples and profiles
- Similarity measure
  - between example and profile (e.g. scalar product)
- Statistical Inference
  - threshold application
- Performance Evaluation
  - Accuracy, Precision/Recall, BEP, f-measure,..

Machine Learning Jargon

- Positive and Negative examples
- Feature representation
  - Kernels
- Learning Algorithm
- Training and test set
- Accuracy measurement
- Generalization/Empirical error Trade-off

Several Kinds of Learning Algorithms

- Logic boolean expressions, (e.g. Decision Trees).
- Probabilistic Functions, (Bayesian Classifier).
- Separating Functions working in vector spaces
  - Non linear: KNN, neural network multiple-layers,…
  - **Linear**: **SVMs**, neural network with one neuron,…
- These approaches are largely applied in language technology
- Very Simple Example: Text Categorization

Error Characterization

- From PAC-learning Theory (***Vapnik***):

$$R(\alpha) \leq R_{emp}(\alpha) + \varphi\left(\frac{d}{m}, \frac{\log(\delta)}{m}\right)$$

$$\varphi\left(\frac{d}{m}, \frac{\log(\delta)}{m}\right) = \sqrt{\frac{d\left(\log\frac{2m}{d}+1\right)-\log\left(\frac{\delta}{4}\right)}{m}}$$

where d is theVC-dimension, m is the number of examples, δ is a bound on the probability to get such error and α is a classifier parameter.

Ranking, Regression and Multiclassification

The Ranking SVM

[Herbrich et al. 1999, 2000; Joachims et al. 2002]

- The aim is to classify instance pairs as correctly ranked or incorrectly ranked
  - This turns an ordinal regression problem back into a binary classification problem
- We want a ranking function f such that

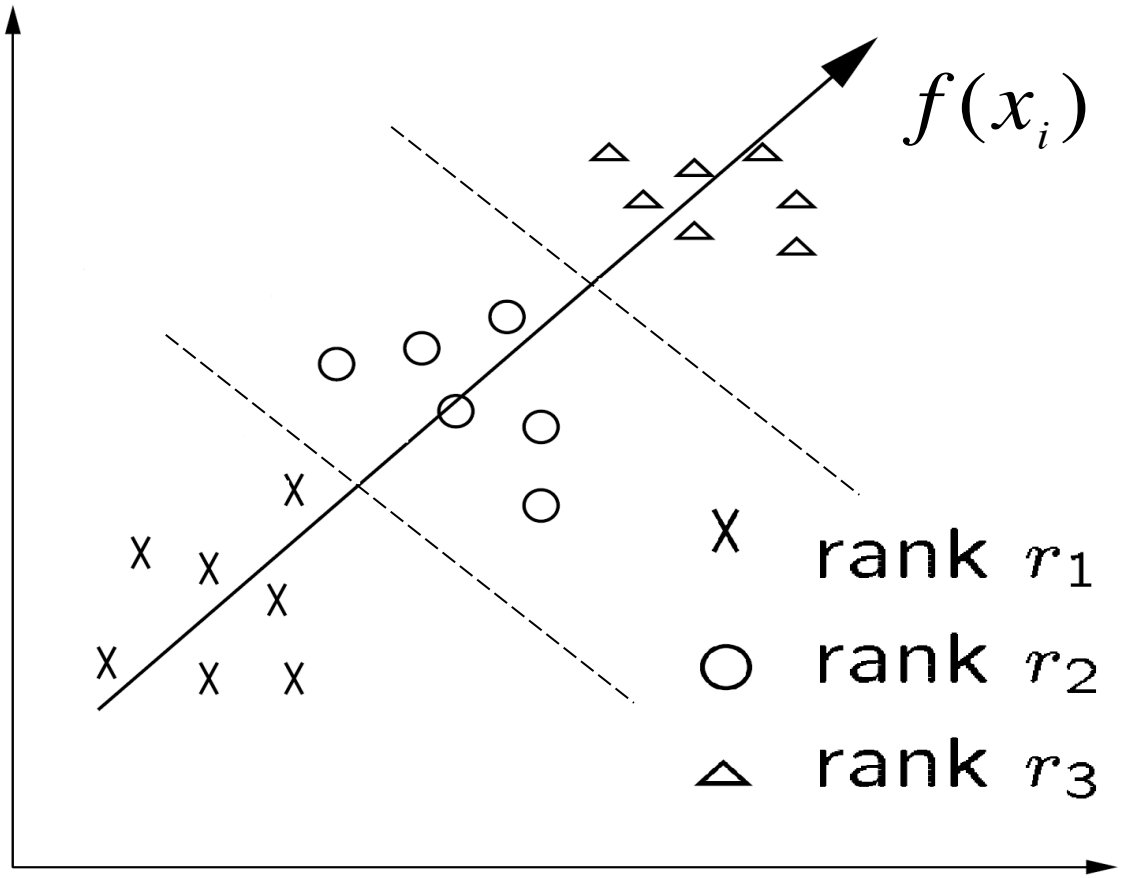
$$\boldsymbol{x}_i > \boldsymbol{x}_j \text{ iff } f(\boldsymbol{x}_i) > f(\boldsymbol{x}_j)$$

- … or at least one that tries to do this with minimal error
- Suppose that f is a linear function

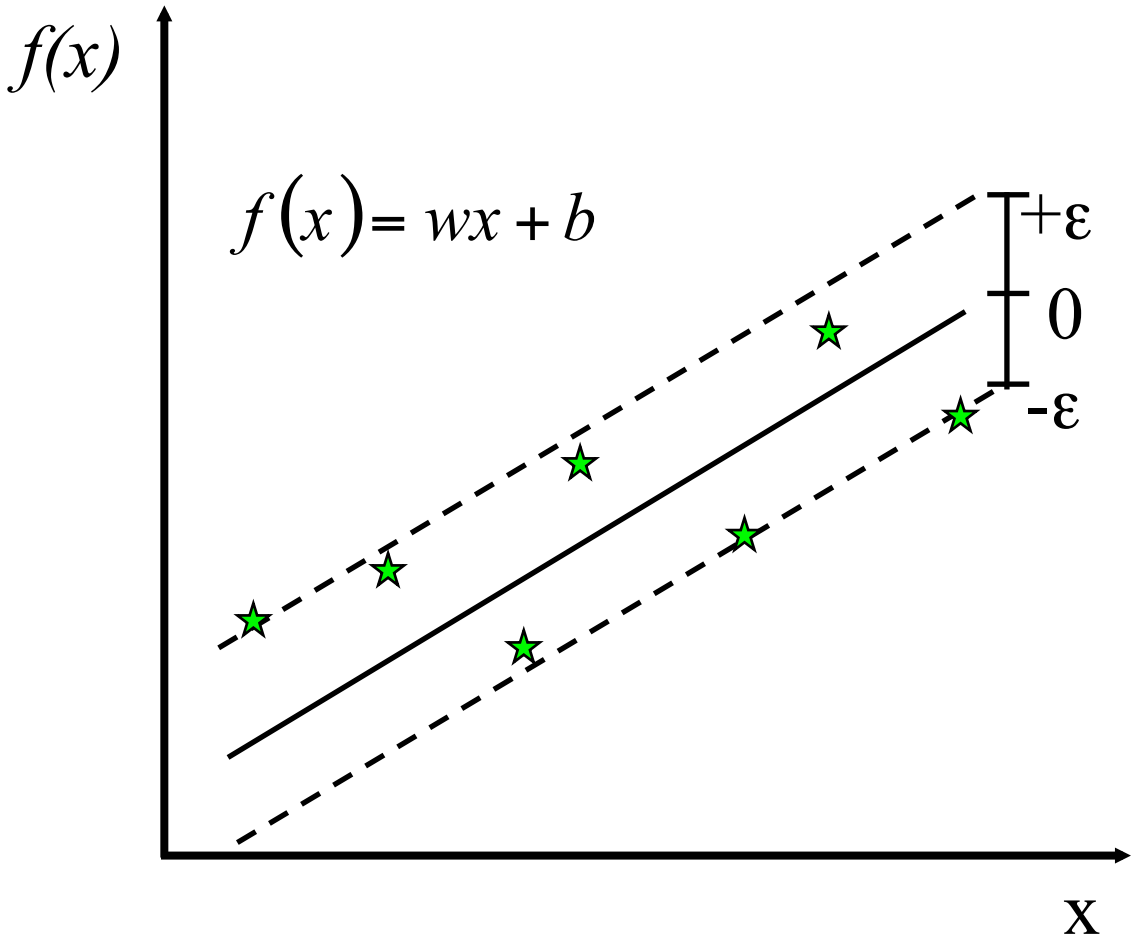
$$f(\boldsymbol{x}_i) = \mathbf{w} \bullet \boldsymbol{x}_i$$

The Ranking SVM

- Ranking Model: $f(\boldsymbol{x}_i)$



Support Vector Regression (SVR)



Solution:

$$Min \frac{1}{2} w^T w$$

Constraints:

$$y_i - w^T x_i - b \leq \varepsilon$$

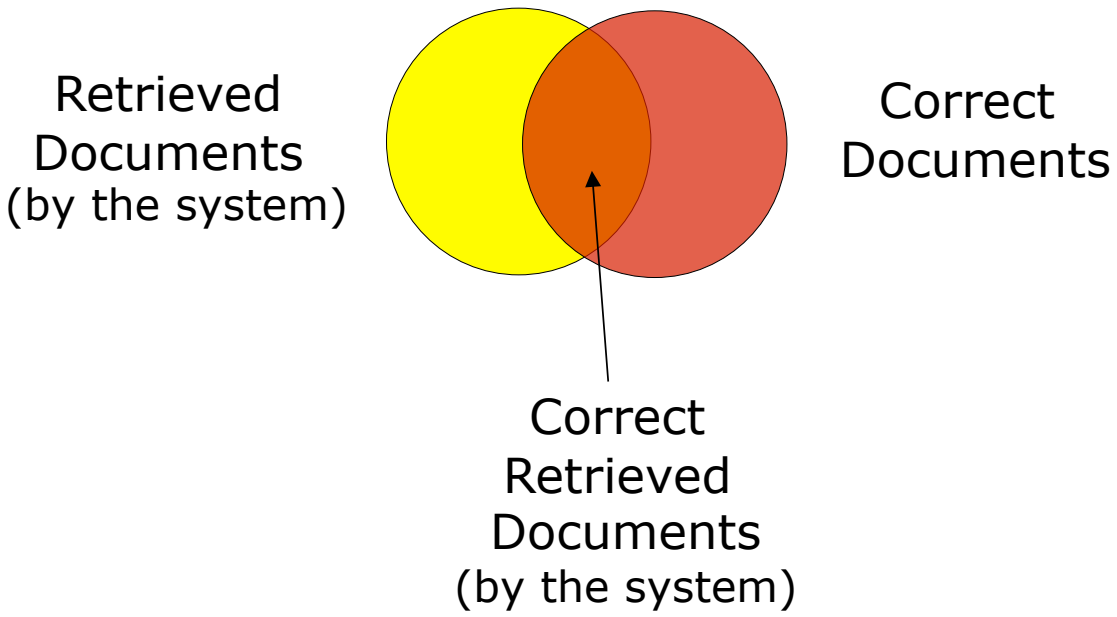
$$w^T x_i + b - y_i \leq \varepsilon$$

From Binary to Multiclass classifiers

- Three different approaches:
- **ONE-vs-ALL (OVA)**
  - Given the example sets, {E1, E2, E3, …} for the categories: {C1, C2, C3,…} the binary classifiers: {b1, b2, b3,…} are built.
  - For b1, E1 is the set of positives and E2∪E3 ∪… is the set of negatives, and so on
  - For testing: given a classification instance x, the category is the one associated with the maximum margin among all binary classifiers

Performance Measurements

- Given a set of document *T*
- Precision = # Correct Retrieved Document / # Retrieved Documents
- Recall = # Correct Retrieved Document/ # Correct Documents



Precision and Recall of C_i

- a, corrects
- b, mistakes
- c, not retrieved

The *Precision* and *Recall* are defined by the above counts:

$$Precision_i = \frac{a_i}{a_i + b_i}$$

$$Recall_i = \frac{a_i}{a_i + c_i}$$

Performance Measurements (cont'd)

- Breakeven Point
  - Find thresholds for which
    Recall = Precision
  - Interpolation
- f-measure
  - Harmonic mean between precision and recall
- Global performance on more than two categories
  - Micro-average
    - The counts refer to classifiers
  - Macro-average (average measures over all categories)

F-measure e MicroAverages

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

$$\mu Precision = \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} a_i + b_i}$$

$$\mu Recall = \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} a_i + c_i}$$

$$\mu BEP = \frac{\mu Precision + \mu Recall}{2}$$

$$\mu f_1 = \frac{2 \times \mu Precision \times \mu Recall}{\mu Precision + \mu Recall}$$

The End

- Machine Learning can write program for you
  - Huge amount of variables
  - Uncertainty
  - Noise in the data
- Needs:
  - Training data
  - Representation of the objects
- Kernels help with representation
- SVMs are efficient and accurate algorithms

One of EternalS mission: How to apply ML for Software-related Tasks

Thank You

Vector Spaces

Definition (1)

- A set V is a **vector space** over a field F (for example, the field of real or of complex numbers) if, given
- an operation *vector* ***addition*** defined in V, denoted **v** + **w** (where **v**, **w** ∈ *V*), and
- an operation, *scalar* ***multiplication*** in V, denoted *a* * **v** (where **v** ∈ V and *a* ∈ F),
- the following properties hold for all *a*, *b* ∈ F and **u**, **v**, and **w** ∈ V:
- **v** + **w** belongs to V.
  (Closure of V under vector addition)
- **u** + (**v** + **w**) = (**u** + **v**) + **w**
  (Associativity of vector addition in V)
- There exists a neutral element **0** in V, such that for all elements **v** in V, **v** + **0** = **v**
  (Existence of an additive identity element in V)

Definition (2)

- For all **v** in V, there exists an element **w** in V, such that **v** + **w** = **0**
  (Existence of additive inverses in V)
- **v** + **w** = **w** + **v**
  (Commutativity of vector addition in V)
- *a* * **v** belongs to V
  (Closure of V under scalar multiplication)
- *a* * (*b* * **v**) = (*ab*) * **v**
  (Associativity of scalar multiplication in V)
- If 1 denotes the multiplicative identity of the field F, then 1 * **v** = **v**
  (Neutrality of one)
- *a* * (**v** + **w**) = *a* * **v** + *a* * **w**
  (Distributivity with respect to vector addition.)
- (*a* + *b*) * **v** = *a* * **v** + *b* * **v**
  (Distributivity with respect to field addition.)

An example of Vector Space

- For all *n,* $\mathbf{R}^n$ forms a vector space over **R**, with component-wise operations.
- Let **V** be the set of all n-tuples, $[v_1,v_2,v_3,...,v_n]$ where v_i is a member of **R**={real numbers}
- Let the field be **R**, as well
- Define Vector Addition:

  For all v, w, in **V**, define $v+w=[v_1+w_1,v_2+w_2,v_3+w_3,...,v_n+w_n]$
- Define Scalar Multiplication:

  For all a in **F** and v in **V**, $a*v=[a*v_1,a*v_2,a*v_3,...,a*v_n]$
- Then **V** is a Vector Space over **R**.

Linear dependency

- Linear combination:
- $\alpha_1 \mathbf{v}_1 + \ldots + \alpha_n \mathbf{v}_n = 0$ for some $\alpha_1 \ldots \alpha_n$ not all zero

  $\Rightarrow y = \alpha_1 \mathbf{v}_1 + \ldots + \alpha_n \mathbf{v}_n$ has a unique expression
- In case $\alpha_i > 0$ and the sum is 1 it is called convex combination

Normed Vector Spaces

- Given a vector space V over a field K, a norm on V is a function from V to $\mathbf{R}$,
- it associates each vector $\mathbf{v}$ in V with a real number, $||\mathbf{v}||$
- The norm must satisfy the following conditions:
  - For all a in K and all $\mathbf{u}$ and $\mathbf{v}$ in V,
    1. $||\mathbf{v}|| \geq 0$ with equality if and only if $\mathbf{v} = \mathbf{0}$
    2. $||a\mathbf{v}|| = |a|\ ||\mathbf{v}||$
    3. $||\mathbf{u} + \mathbf{v}|| \leq ||\mathbf{u}|| + ||\mathbf{v}||$
- A useful consequence of the norm axioms is the inequality
  - $||\mathbf{u} \pm \mathbf{v}|| \geq |\ ||\mathbf{u}|| - ||\mathbf{v}||\ |$
- for all vectors $\mathbf{u}$ and $\mathbf{v}$

Inner Product Spaces

- Let V be a vector space and **u**, **v**, and **w** be vectors in V and c be a constant.
- Then, an *inner product* (,) on V is
  - a function with domain consisting of pairs of vectors and
  - range real numbers satisfying
  - the following properties:
    1. $(\mathbf{u}, \mathbf{u}) \geq 0$ with equality if and only if $\mathbf{u} = \mathbf{0}$.
    2. $(\mathbf{u}, \mathbf{v}) = (\mathbf{v}, \mathbf{u})$
    3. $(\mathbf{u} + \mathbf{v}, \mathbf{w}) = (\mathbf{u}, \mathbf{w}) + (\mathbf{v}, \mathbf{w})$
    4. $(c\mathbf{u}, \mathbf{v}) = (\mathbf{u}, c\mathbf{v}) = c(\mathbf{u}, \mathbf{v})$

Example

- Let V be the vector space consisting of all continuous functions with the standard + and *. Then define an inner product by

$$(f,g)=\int_0^1 f(t)g(t)dt$$

- For example: $(x,x^2) = \int_0^1 (x)(x^2)dx = \frac{1}{4}$
- The four properties follow immediately from the analogous property of the definite integral:

$$(f+g,h)=\int_0^1 (f+g)(t)h(t)\,dt$$

$$=\int_0^1 \{f(t)h(t)+g(t)h(t)\}\,dt=\int_0^1 f(t)h(t)\,dt+\int_0^1 g(t)h(t)\,dt$$

$$=(f,h)+(g,h)$$

Inner Product Properties

- $(\mathbf{v}, \mathbf{0}) = 0$
- $\| v \| = \sqrt{(v, v)}$
- If $(\mathbf{v}, \mathbf{u}) = 0$, $\mathbf{v}, \mathbf{u}$ are called orthogonal
- Schwarz Inequality:
  - $[(\mathbf{v}, \mathbf{u})]^2 \leq (\mathbf{v}, \mathbf{v})\,(\mathbf{u}, \mathbf{u})$
- The classical scalar product is the component-wise product
- $(x_1, x_2, \ldots, x_n)\,(y_1, y_2, \ldots, y_n) = x_1 y_1 + x_2 y_2 + \ldots + x_n y_n$
- $$\cos(u, v) = \frac{(u, v)}{\| u \| \cdot \| v \|}$$

Projection

- From $\cos(\vec{x}, \vec{w}) = \dfrac{\vec{x} \cdot \vec{w}}{\| \vec{x} \| \cdot \| \vec{w} \|}$

- It follows that

$$\| \vec{x} \| \cos(\vec{x}, \vec{w}) = \frac{\vec{x} \cdot \vec{w}}{\| \vec{w} \|} = \vec{x} \cdot \frac{\vec{w}}{\| \vec{w} \|}$$

- Norm of $\vec{x}$ times the cosine between $\vec{x}$ and $\vec{w}$, i.e. the projection of $\vec{x}$ on $\vec{w}$

Similarity Metrics

- The simplest distance for continuous *m*-dimensional instance space is *Euclidian distance*.
- The simplest distance for *m*-dimensional binary instance space is *Hamming distance* (number of feature values that differ).
- Cosine similarity is typically the most effective