Computational Models for Data Analysis

Introduction

Alessandra Giordani

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

Course Schedule

- April 16: 15:45 - 18:15
- May 7: 15:45 - 18:15
- May 9: 14:30 - 17:00
- May 16: 14:30 - 17:00
- May 23: 14:30 - 17:00
- May 28: 14:30 - 17:00
- May 30: 14:30 - 17:00

Exams

- June 13
- July 8
- September 11

Lectures

- Introduction to ML
  - Decision Tree
  - Bayesian Classifiers
  - Vector spaces
- Vector Space Categorization
  - Feature design, selection and weighting
  - Document representation
  - Category Learning: Rocchio and KNN
  - Measuring of Performance
  - From binary to multi-class classification

Lectures

- Probably Approximately Correct Learning (PAC)
  - VC dimension
- Perceptron
  - Vector Space Model
  - Representer Theorem
- Support Vector Machines (SVMs)
  - Hard/Soft Margin (Classification)
  - Regression and ranking

Lectures

- Kernels Methods
  - Theory and Algebraic properties
  - Linear, Polynomial, Gaussian
  - Kernel construction,
- Kernels for structured data
  - Sequence, Tree Kernels
- Structured Output

Slides of former professor also here:

Computational Methods for Data Analysis

- Introduction to Machine Learning: Decision Tree and Bayesian Classifiers
- Vector Space Learning
- Introduction to Statistical Learning Theory
- VC-dimension
- Perceptron
- Support Vector Machines
- Kernel Methods for Structured Data

As referring text please use my new chapter:

Kernel-Based Machines for Abstract and Easy Modeling of Automatic Learning

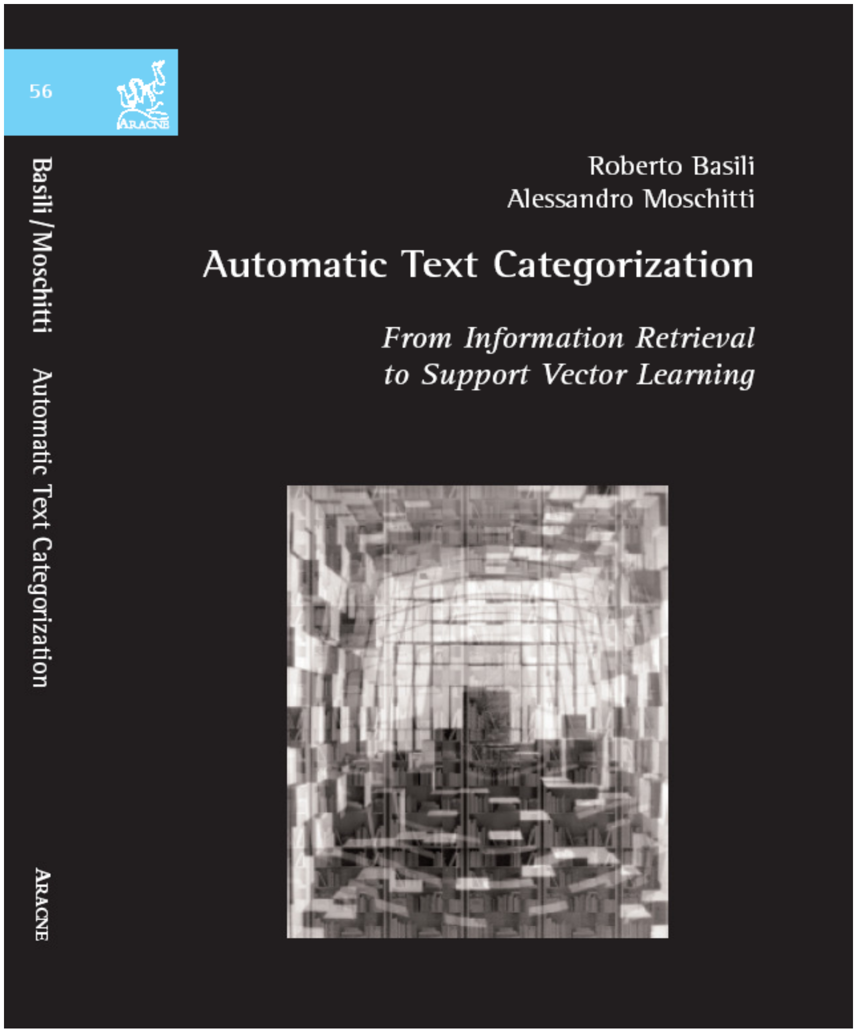
along with the old book (with some typos)

Roberto Basili and Alessandro Moschitti, *Automatic Text Categorization: from Information Retrieval to Support Vector Learning*. Aracne editrice, Rome, Italy.

http://disi.unitn.it/moschitti/teaching.html

http://disi.unitn.it/~agiordani/teaching.htm

Reference Book + some articles



Today

- Introduction to Machine Learning
- Vector Spaces

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

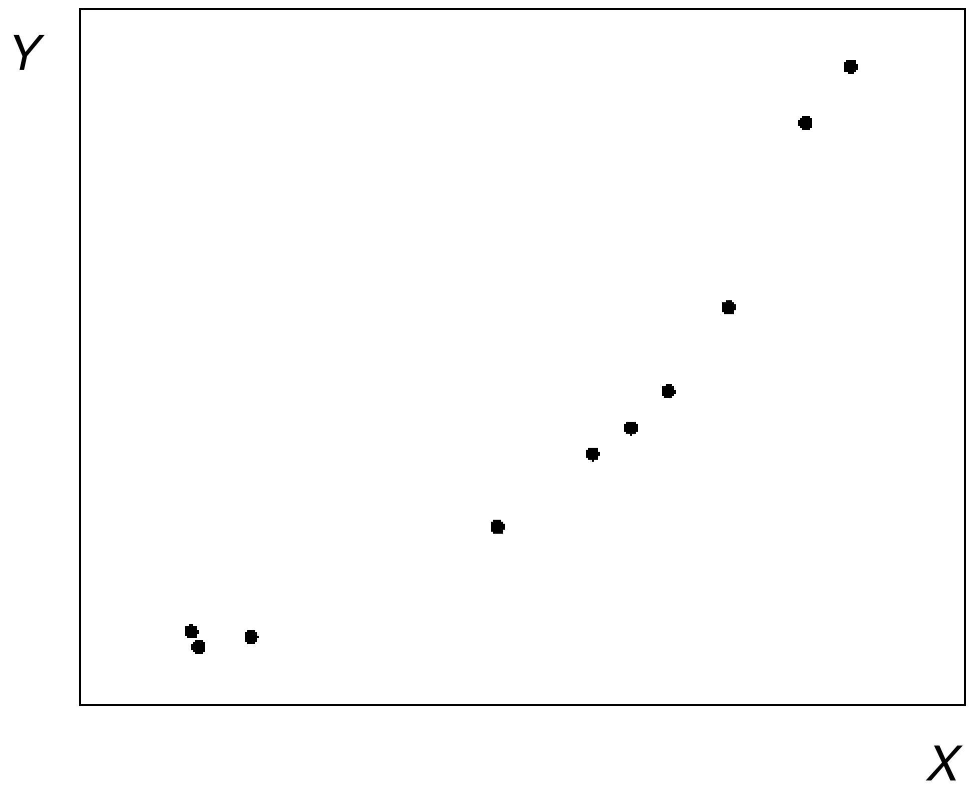
More concretely

- 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

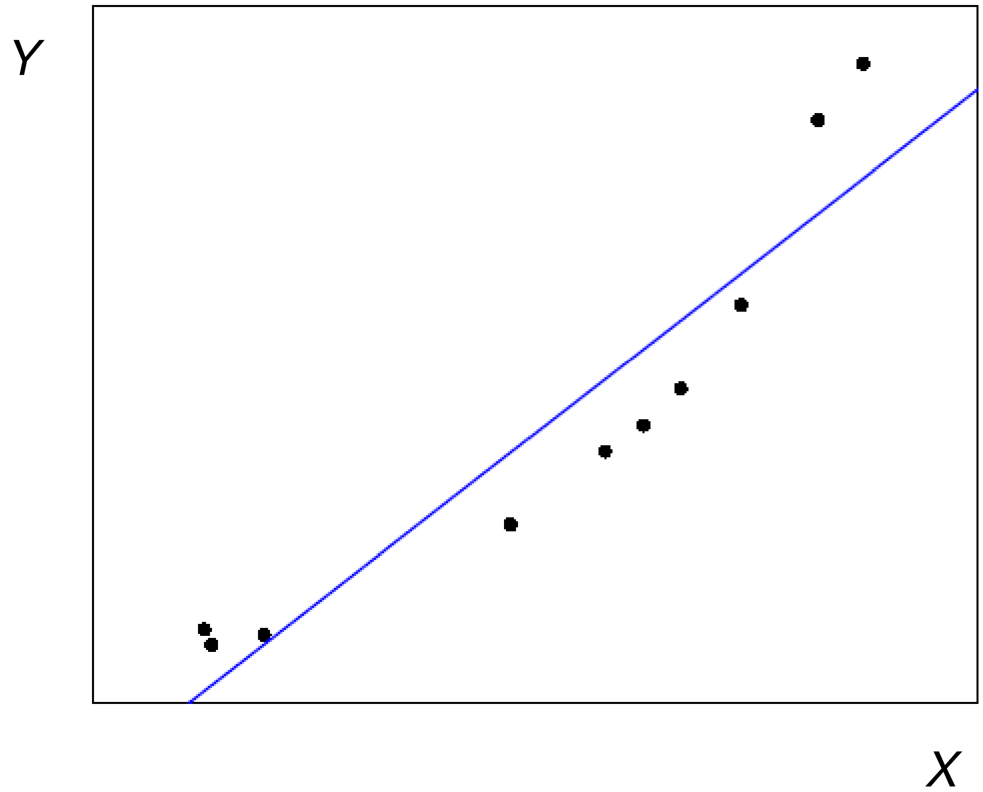
What is Statistical Learning?

- 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. $\langle x^*, f(x^*)\rangle = \langle x^*, y^*\rangle$

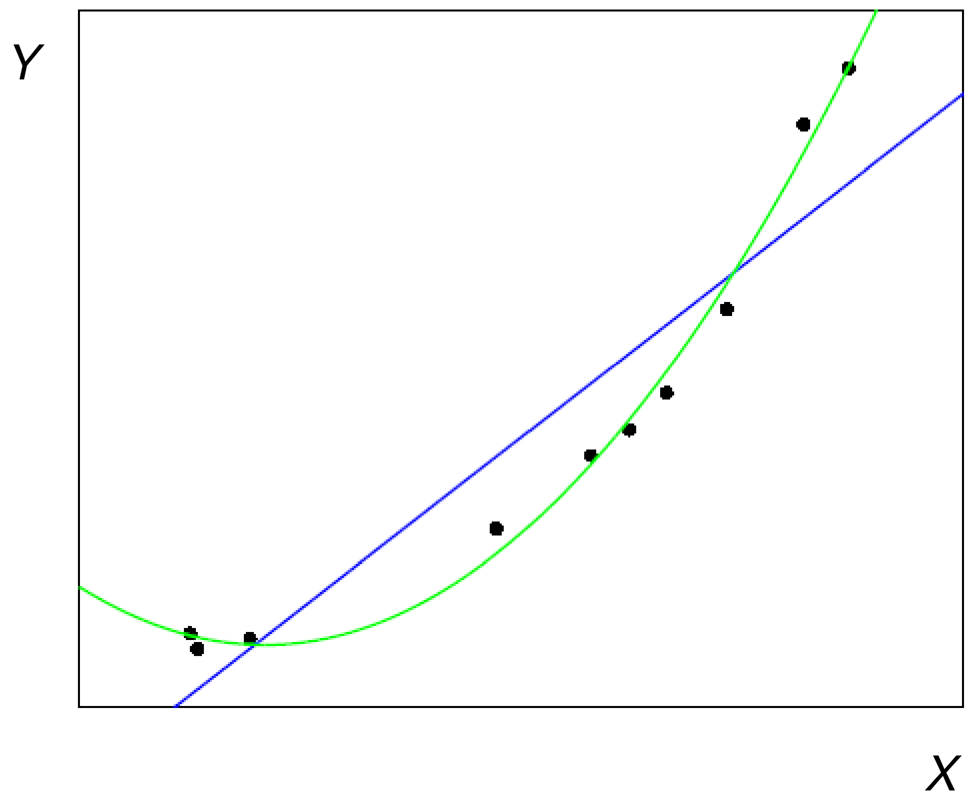
You have already tackled the learning problem



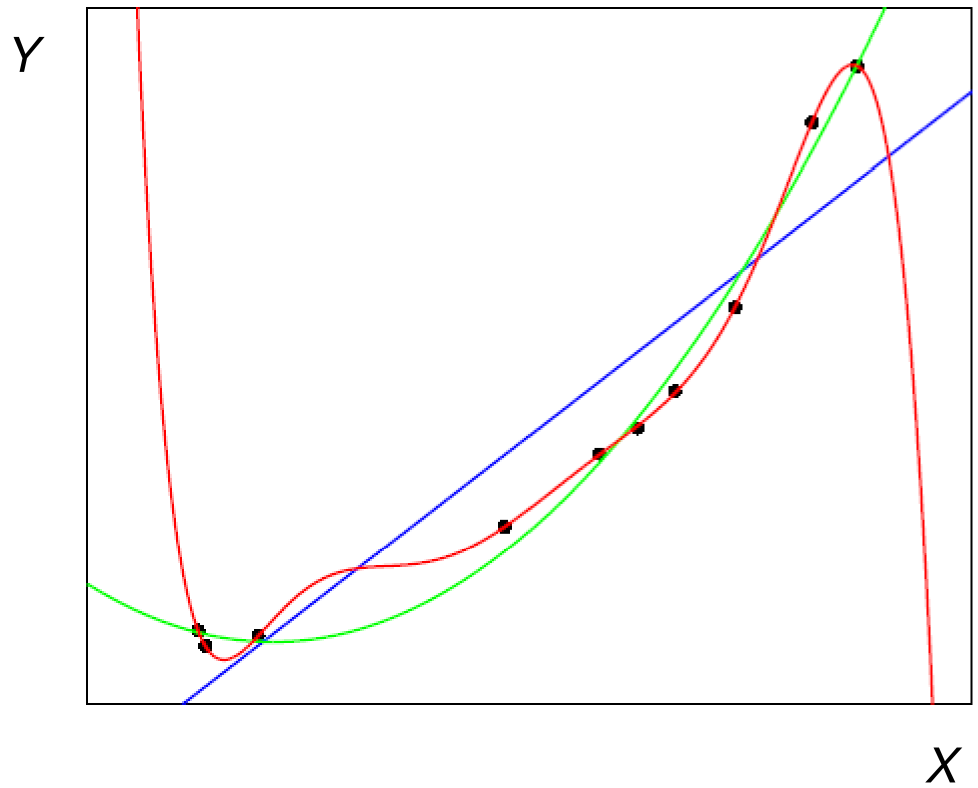
Linear Regression



Degree 2



Degree



Machine Learning Problems

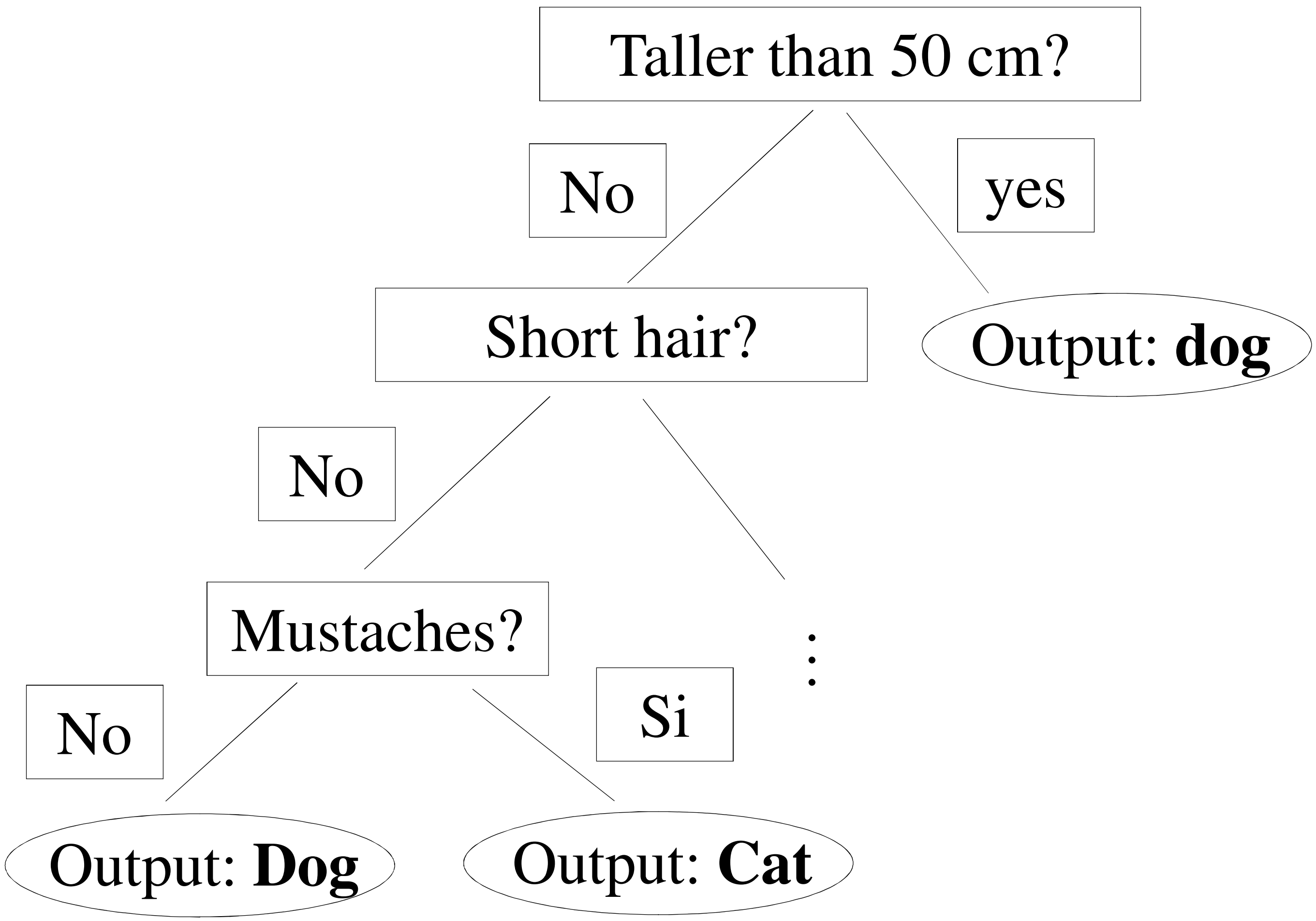
- Overfitting
- How dealing with millions of variables instead of only two?
- How dealing with real world objects instead of real values?

Learning Models

- Real Values: *regression*
- Finite and integer: *classification*
- Binary Classifiers:
  - 2 classes, e.g.
    $f(x) \rightarrow \{\text{cats}, \text{dogs}\}$

Decision Trees

Decision Tree (between Dogs/Cats)



Mustaches or Whiskers

- Whiskers are an important orientation tools for both dogs and cats
- all dogs and cats have them

$\Longmapsto$ not good features

- We may use their length
- What about mustaches?

Mustaches?

Entropy-based feature selection

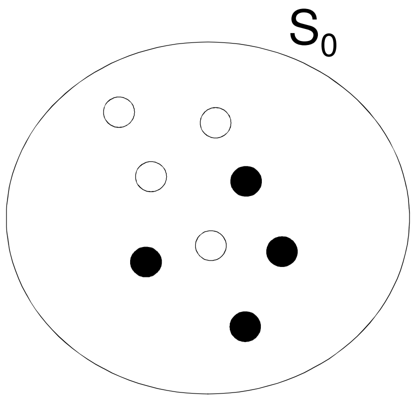
- Entropy of class distribution $P(C_i)$:

$$H(P) = \sum_{i=1}^{m} -P(C_i) log_2(P(C_i))$$

- Measure "how much the distribution is uniform"
- Given $S_1 \ldots S_n$ sets partitioned wrt a feature the overall entropy is:

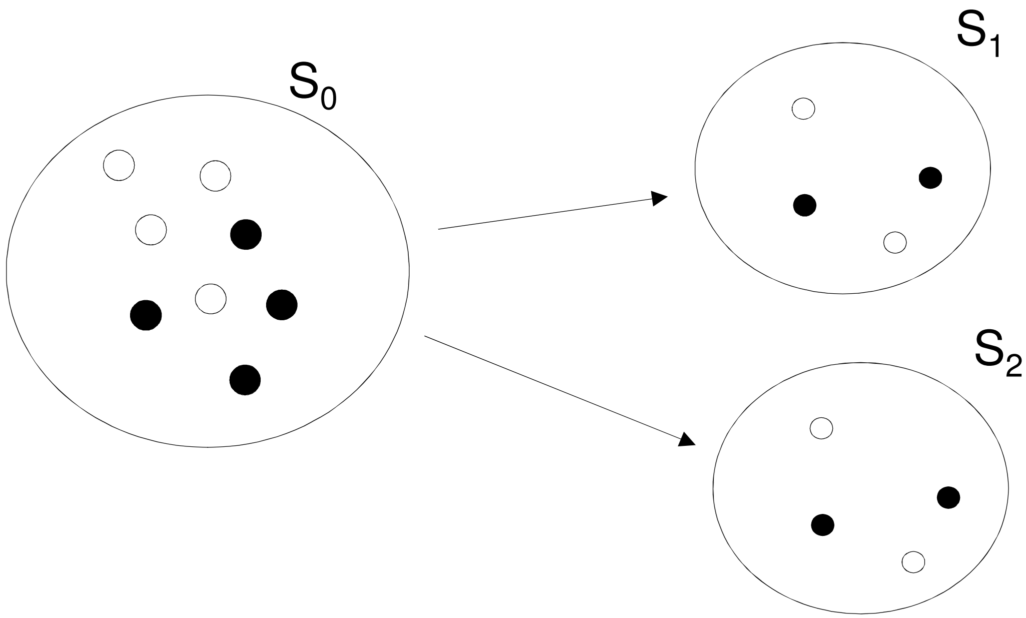
$$\bar{H}(P^{S_1}, .., P^{S_n}) = \sum_{i=1}^{m} \frac{H(P^{S_i})}{|S_i|}$$

Example: cats and dogs classification



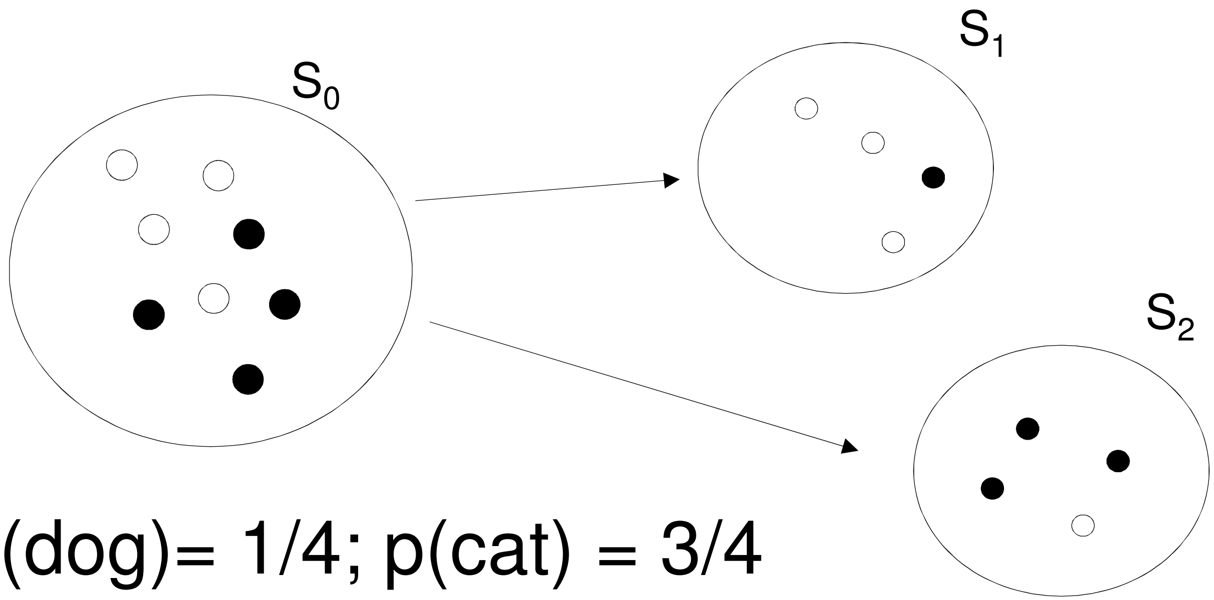
- p(dog)=p(cat) = 4/8 = ½ (for both dogs and cats)
- $H(S_0)$ = ½*log(2) * 2 = 1

Has the animal more than 6 siblings?



- p(dog)=p(cat) = 2/4 = ½ (for both dogs and cats)
- $H(S_1) = H(S_2) = ¼ * [½*log(2) * 2] = 0.25$
- $All(S_1, S_2) = 2*.25 = 0.5$

Does the animal have short hair?



- p(dog)= 1/4; p(cat) = 3/4
- $H(S_2)=H(S_1) = \frac{1}{4} * [(1/4)*\log(4) + (3/4)*\log(4/3)] = \frac{1}{4} * [\frac{1}{2} + 0.31] = \frac{1}{4} * 0.81 = 0.20$
- $All(S_1,S_2) = 0.20*2 = 0.40$ (note that $|S1| = |S2|$)

Follow up

- ***hair length feature*** is better than ***number of siblings*** since 0.40 is lower than 0.50
- Test all the features
- Choose the best
- Start with a new feature on the collection sets induced by the best feature

Probabilistic Classifier

Probability (1)

- Let Ω be a space and β a collection of subsets of Ω
- β is a collection of events
- A probability function *P* is defined as:

$$P : \beta \to [0,1]$$

Definition of Probability

- *P* is a function which associates each event *E* with a number *P(E)* called probability of *E* as follows:

$$1)\ 0 \leq P(E) \leq 1$$

$$2)\ P(\Omega) = 1$$

$$3)\ P(E_1 \vee E_2 \vee \ldots \vee E_n \vee \ldots) = \\ = \sum_{i=1}^{\infty} P(E_i) \text{ if } E_i \wedge E_j = 0,\ \forall i \neq j$$

Finite Partition and Uniformly Distributed

- Given a partition of *n* events uniformly distributed (with a probability of 1/*n*); and
- given an event *E*, we can evaluate its probability as:

$$P(E) = P(E \wedge E_{tot}) = P(E \wedge (E_1 \vee E_2 \vee \ldots \vee E_n)) =$$

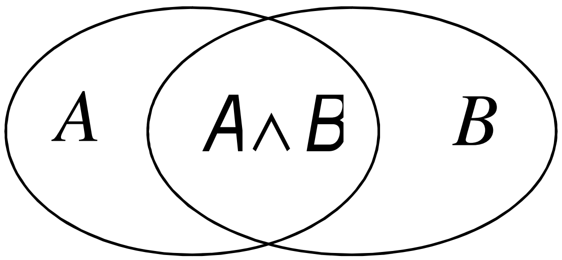
$$\sum_i P(E \wedge E_i) = \sum_{E_i \subset E} P(E_i) = \sum_{E_i \subset E} \frac{1}{n} =$$

$$\frac{1}{n} \sum_{E_i \subset E} 1 = \frac{1}{n}(\left|\{i : E_i \subset E\}\right|) = \frac{\text{Target Cases}}{\text{All Cases}}$$

Conditioned Probability

- *P(A / B)* is the probability of *A* given *B*
- *B* is the piece of information that we know
- The following rule holds:

$$P(A \mid B) = \frac{P(A \wedge B)}{P(B)}$$



Indipendence

- *A* and *B* are indipedent *iff*:

$$P(A \mid B) = P(A)$$

$$P(B \mid A) = P(B)$$

- If *A* and *B* are indipendent:

$$P(A) = P(A \mid B) = \frac{P(A \land B)}{P(B)}$$

$$P(A \land B) = P(A)P(B)$$

Bayes's Theorem

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

Proof:

$$P(A \mid B) = \frac{P(A \wedge B)}{P(B)}$$ (Def. of. Cond. prob)

$$P(B \mid A) = \frac{P(A \wedge B)}{P(A)}$$ Def. of. Cond. prob

$$P(A \mid B) = \frac{[P(B \mid A)P(A)]}{P(B)}$$

Bayesian Classifier

- Given a set of categories $\{c_1, c_2, \ldots c_n\}$
- Let E be a description of a classifying example.
- The category of E can be derived by using the following probability:

$$P(c_i \mid E) = \frac{P(c_i)P(E \mid c_i)}{P(E)}$$

$$\sum_{i=1}^{n} P(c_i \mid E) = \sum_{i=1}^{n} \frac{P(c_i)P(E \mid c_i)}{P(E)} = 1$$

$$P(E) = \sum_{i=1}^{n} P(c_i)P(E \mid c_i)$$

Bayesian Classifier (cont)

- We need to compute:
  - the posterior probability: P(c_i)
  - the conditional probability: P(E | c_i)
- P(c_i) can be estimated from the training set, D.
  - given n_i examples in D of type c_i, then P(c_i) = n_i / $|D|$
- Suppose that an example is represented by *m features*:

$$E = e_1 \wedge e_2 \wedge \cdots \wedge e_m$$

- The elements will be exponential in *m* so there are not enough training examples to estimate P(E | c_i)

Naïve Bayes Classifiers

- The *features* are assumed to be indipendent given a category (c_i).

$$P(E \mid c_i) = P(e_1 \wedge e_2 \wedge \cdots \wedge e_m \mid c_i) = \prod_{j=1}^{m} P(e_j \mid c_i)$$

- This allows us to only estimate $P(e_j / c_i)$ for each *feature* and category.

An example of the Naïve Bayes Clasiffier

- C = {Allergy, Cold, Healthy}
- e_1 = sneeze; e_2 = cough; e_3 = fever
- E = {sneeze, cough, ¬fever}

Prob	**Healthy**	**Cold**	**Allergy**
P(c_i)	0.9	0.05	0.05
P(sneeze|c_i)	0.1	0.9	0.9
P(cough|c_i)	0.1	0.8	0.7
P(fever|c_i)	0.01	0.7	0.4

An example of the Naïve Bayes Clasiffier (cont.)

Probability	Healthy	Cold	Allergy
P(c_i)	0.9	0.05	0.05
P(sneeze | c_i)	0.1	0.9	0.9
P(cough | c_i)	0.1	0.8	0.7
P(fever | c_i)	0.01	0.7	0.4

$E=\{\text{sneeze, cough, } \neg\text{fever}\}$

P(Healthy| E) = (0.9)(0.1)(0.1)(0.99)/P(E)=0.0089/P(E)

P(Cold | E) = (0.05)(0.9)(0.8)(0.3)/P(E)=0.01/P(E)

P(Allergy | E) = (0.05)(0.9)(0.7)(0.6)/P(E)=0.019/P(E)

The most probable category is allergy

P(E) = 0.0089 + 0.01 + 0.019 = 0.0379

P(Healthy| E) = 0.23, P(Cold | E) = 0.26, P(Allergy | E) = 0.50

Probability Estimation

- Estimate counts from training data.
- Let n_i be the number of examples in c_i
- let n_{ij} be the number of examples of c_i containing the feature e_j, then:

$$P(e_j \mid c_i) = \frac{n_{ij}}{n_i}$$

- Problems: the data set may still be too small.
- For rare features we may have, e_k, $\forall c_i$:P(e_k | c_i) = 0.

Smoothing

- The probabilities are estimated even if they are not in the data
- Laplace smoothing
  - each feature has a priori probability, *p*,
  - We assume that such feature has been observed in an example of size *m*.

$$P(e_j \mid c_i) = \frac{n_{ij} + mp}{n_i + m}$$

Naïve Bayes for text classification

- "bag of words" model
  - The examples are category documents
  - Features: Vocabulary $V = \{w_1, w_2, \ldots w_m\}$
  - $P(w_j \mid c_i)$ is the probability to have w_j in a category i
- Let us use the Laplace's smoothing
  - Uniform distribution ($p = 1/|V|$) and $m = |V|$
  - That is each word is assumed to appear exactly one time in a category

Training (version 1)

- V is built using all training documents D
- For each category $c_i \in C$

  Let D_i the document subset of D in c_i

  $\Rightarrow P(c_i) = |D_i| \,/\, |D|$

  n_i is the total number of words in D_i

  for each $w_j \in V$, n_{ij} is the counts of w_j in c_i

  $\Rightarrow P(w_j \mid c_i) = (n_{ij} + 1) \,/\, (n_i + |V|)$

Testing

- Given a test document *X*
- Let *n* be the number of words of *X*
- The assigned category is:

$$\operatorname*{argmax}_{c_i \in C} P(c_i) \prod_{j=1}^{n} P(a_j \mid c_i)$$

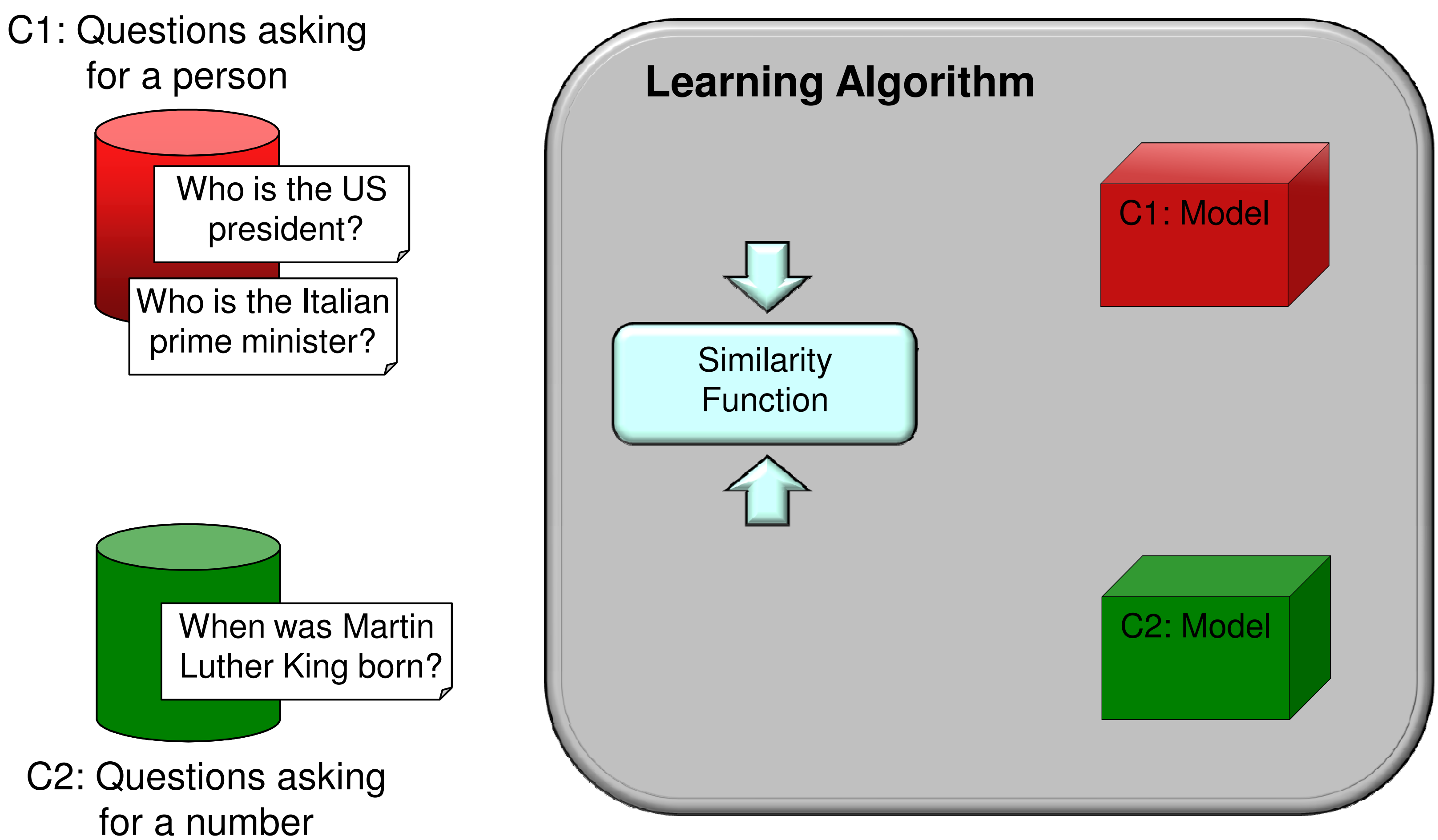
where a_j is a word at the j-th position in *X*

Part I: Abstract View of Statistical Learning Theory

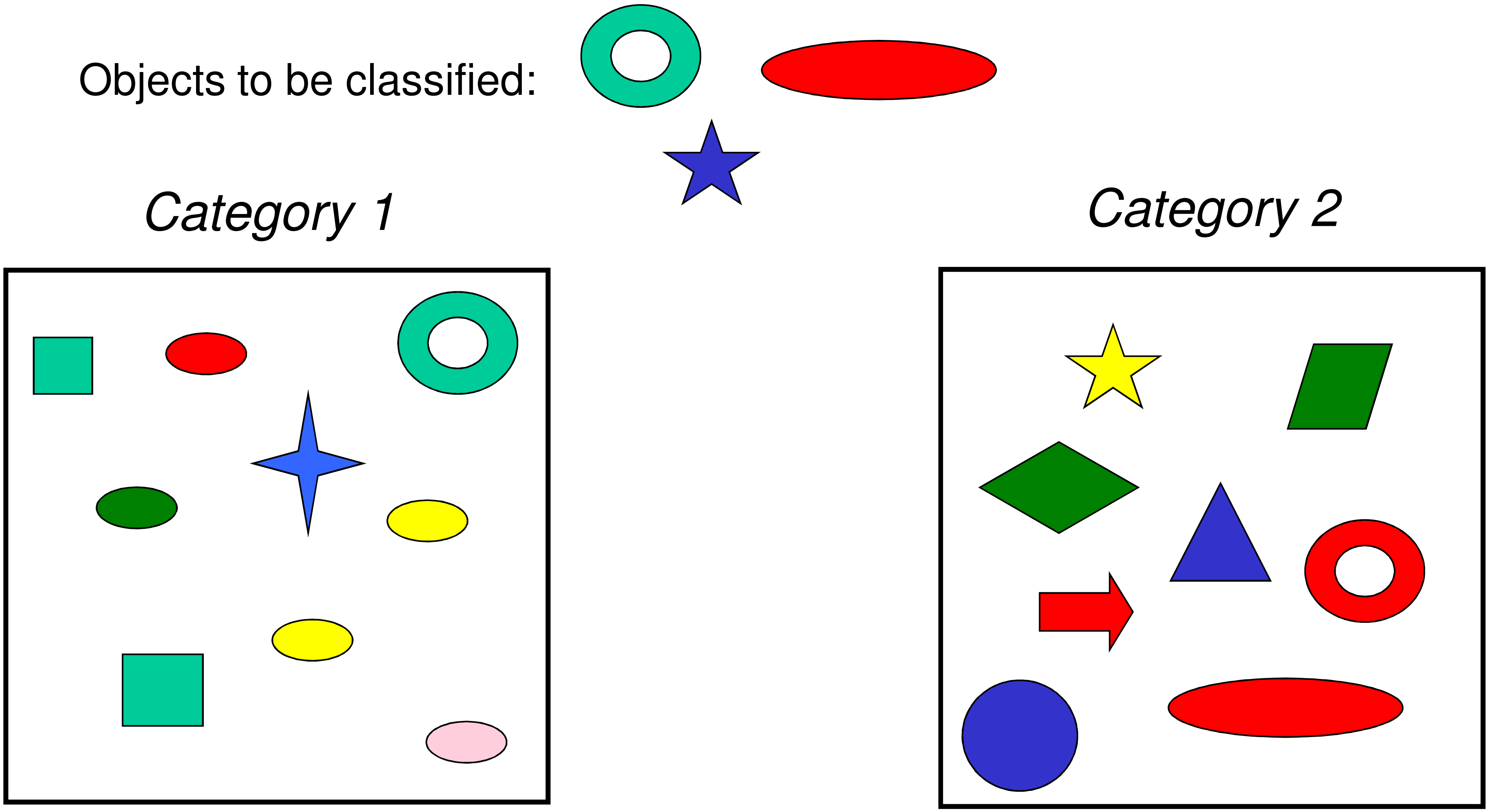
Main Ingredients of Statistical Learning

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

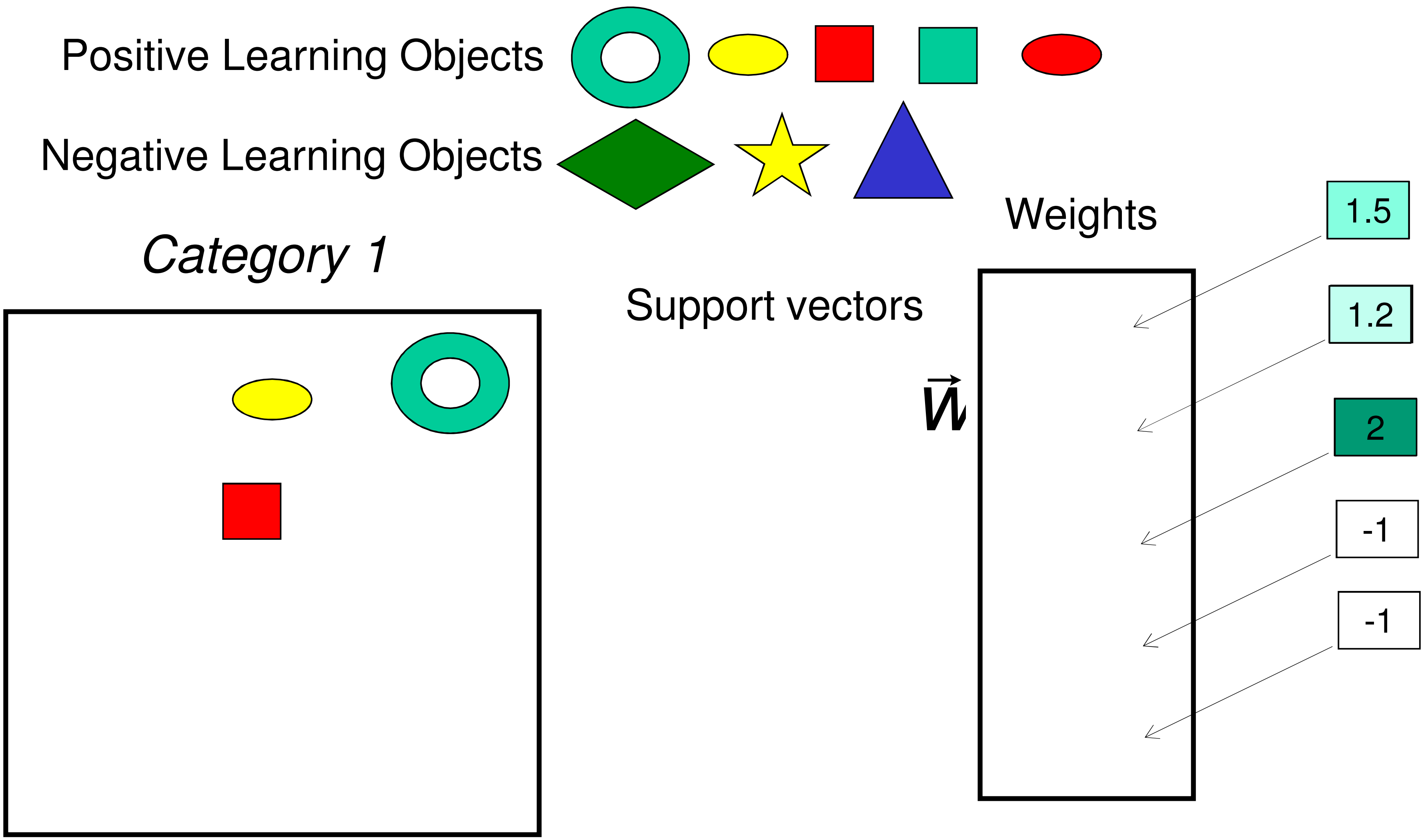
Intuitions on Machine Learning (kernel machines)



Example based Classifiers



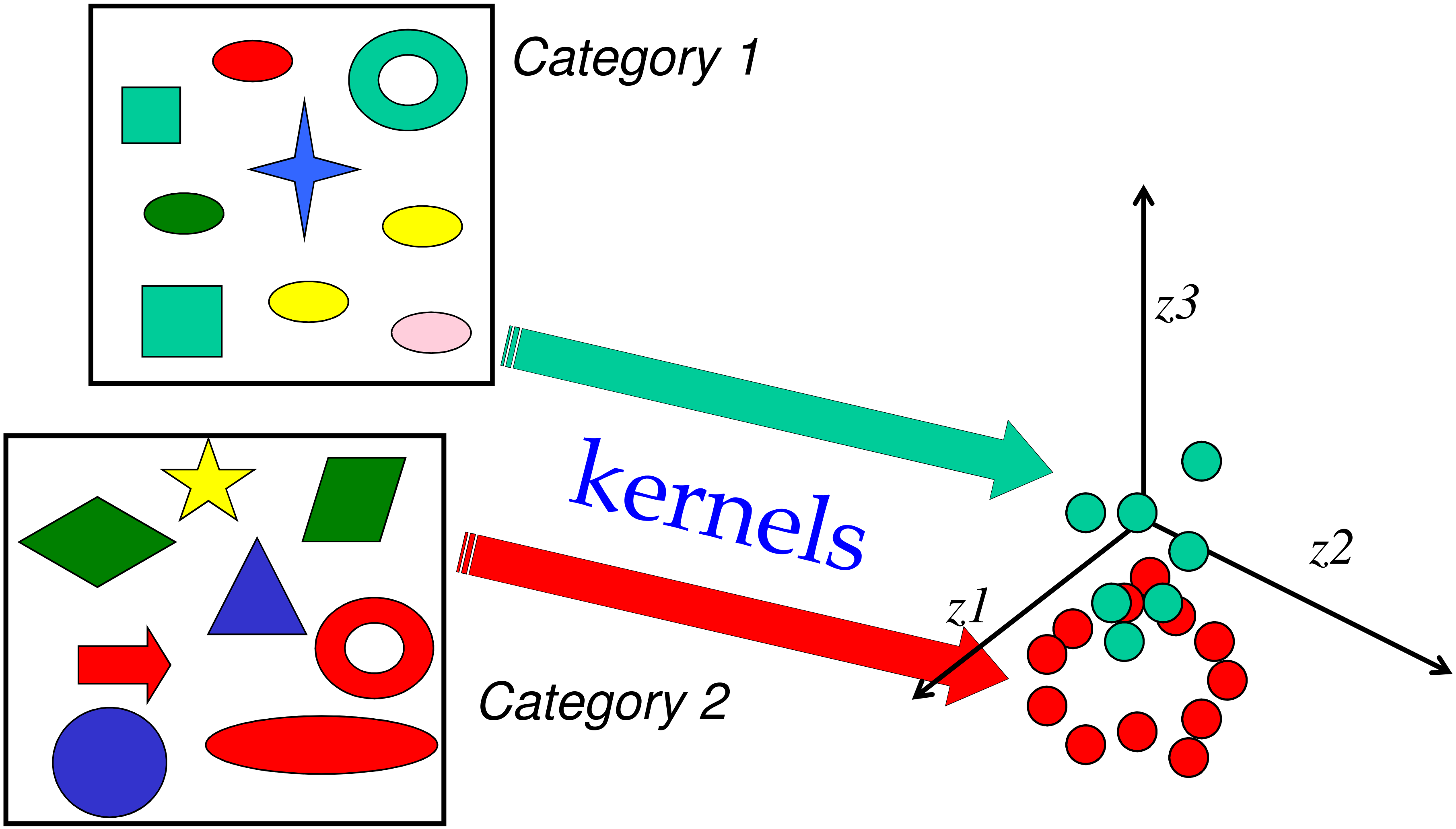
Learning phase



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



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 $\mathbf{v} + \mathbf{w}$ (where $\mathbf{v}, \mathbf{w} \in V$), and
- an operation, *scalar* ***multiplication*** in V, denoted $a * \mathbf{v}$ (where $\mathbf{v} \in V$ and $a \in F$),
- the following properties hold for all $a, b \in F$ and $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w} \in V$:
- $\mathbf{v} + \mathbf{w}$ belongs to V.
(Closure of V under vector addition)
- $\mathbf{u} + (\mathbf{v} + \mathbf{w}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w}$
(Associativity of vector addition in V)
- There exists a neutral element $\mathbf{0}$ in V, such that for all elements $\mathbf{v}$ in V, $\mathbf{v} + \mathbf{0} = \mathbf{v}$
(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:

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

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

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

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
- $(x_1, x_2, \ldots, x_n)\,(y_1, y_2, \ldots, y_n) = x_1 y_1 + x_2 y_2 + \ldots + x_n y_n$
- $\cos(u, v) = \dfrac{(u, v)}{\| u \| \cdot \| v \|}$

Projection

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

- It follows that

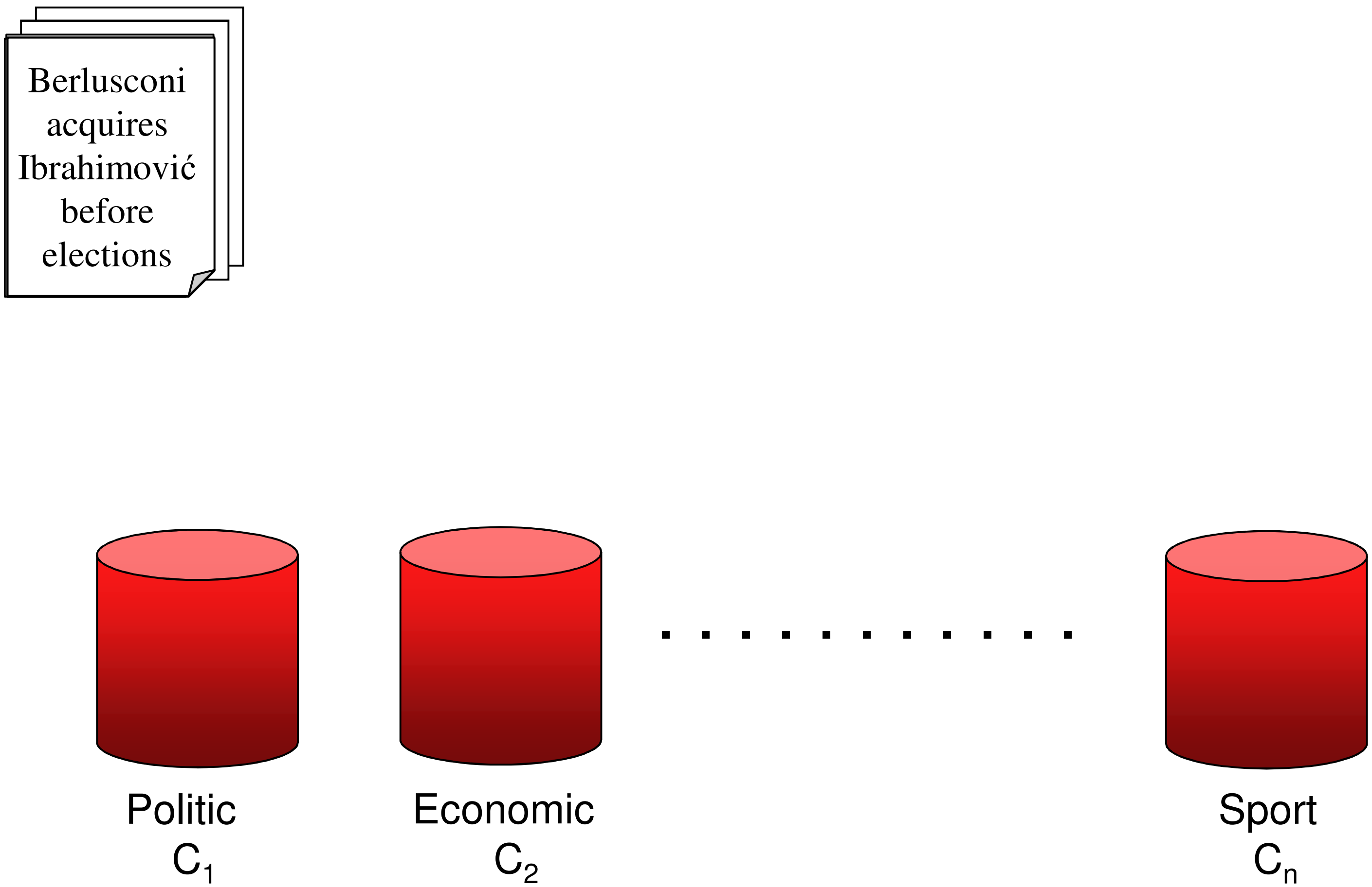
$$\|\vec{x}\|\cos(\vec{x},\vec{w}) = \frac{\vec{x}\cdot\vec{w}}{\|\vec{w}\|} = \vec{x}\cdot\frac{\vec{w}}{\|\vec{w}\|}$$

- 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

A Simple Example: Text Categorization



Text Classification Problem

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

  define $f : T \rightarrow 2^{C}$

The Vector Space Model (VSM)

d_1: Politic

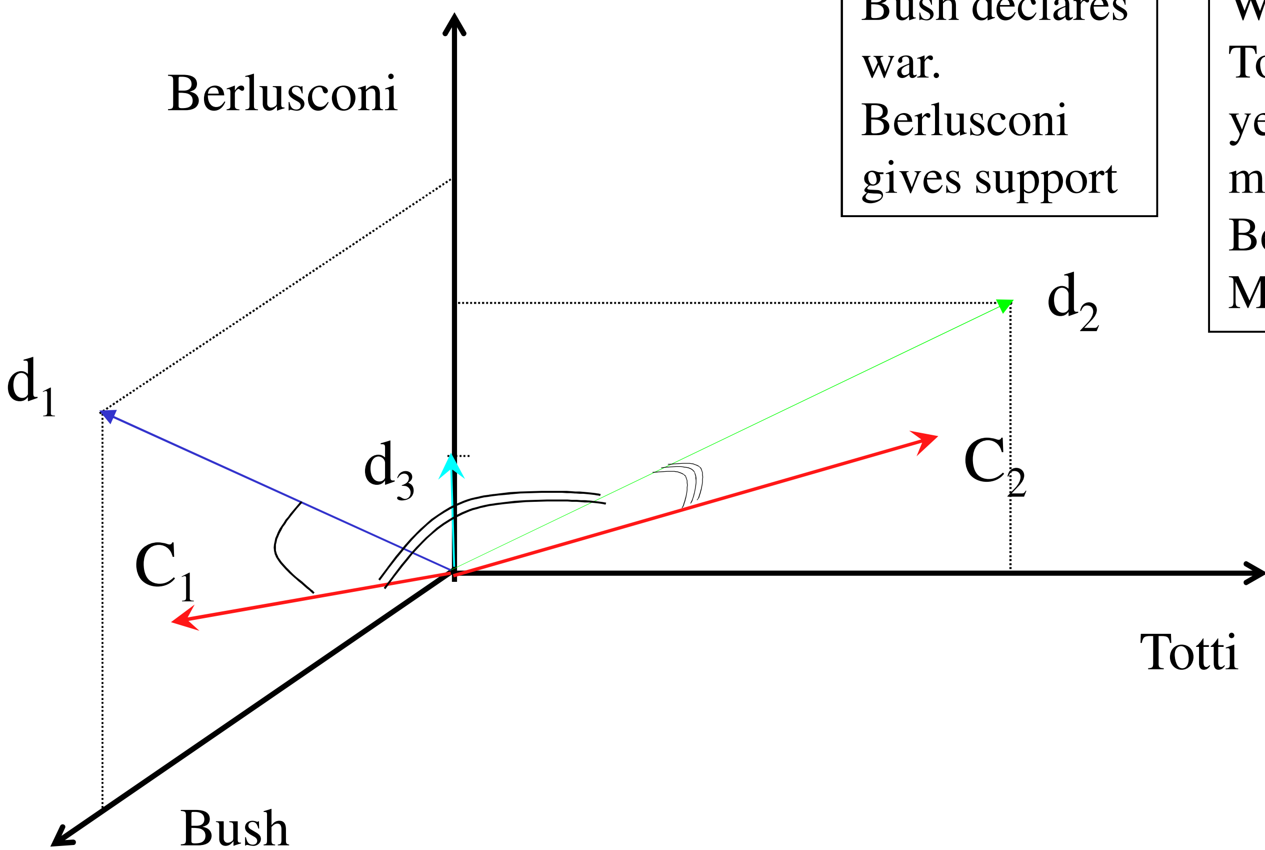
Bush declares war. Berlusconi gives support

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

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

Summary of Today Machine Learning Concepts

- 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

What Next?

- Can we learn any function?
- Statistical Learning Theory
  - PAC learning