Master in Human Language Technology and Interfaces

Machine Learning

Alessandro Moschitti

Department of information and communication technology
University of Trento
Email: moschitti@dit.unitn.it

Course Schedule

- prof **Moschitti**
- Today, 14:00-16:00

  Friday, Tomorrow: 14:00-16:00, Room 201
- Since November 12
- Thursday and Friday: 14:00-16:00, Room 106

Course Schedule (2)

- prof **Farid**
- A cycle of seminar lectures
  - Not completely sequential
  - Offer you to have a possibility to learn different topics
- **Monday,** 13:30 - 15:30, classroom 104

  Tuesday – sometime depending on the needs and on the availability of HL course

  (classroom 108, 13:30-14:30)

Content of Moschitti's lectures

- PAC Learning
  - VC dimension
- Perceptron
  - Vector Space Model
  - Representer Theorem
- Support Vector Machines (SVMs)
  - Hard/Soft Margin (Classification)
  - Regression and ranking
- Kernels Methods
  - Theory and Algebraic properties
  - Linear, Polynomial, Gaussian
  - Kernel construction,
- kernels for structured data (introduction)
  - Sequence, Tree Kernels

Moschitti's Lab

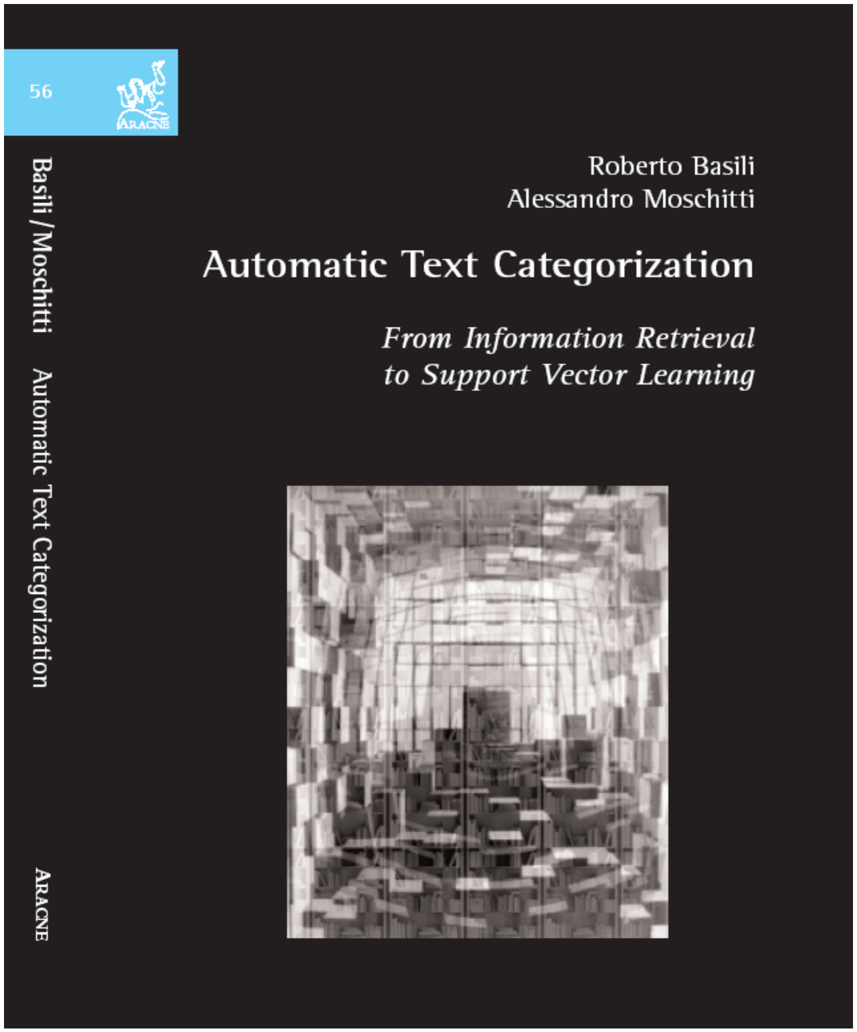
Minimal schedule

- Automated Text Categorization
- Question Classification (Question Answering)

Optional Topics

- Semantic Role Labeling
- Relation Extraction
- Named Entity Recognition
- Textual Entailment Recognition

Reference Book + some articles



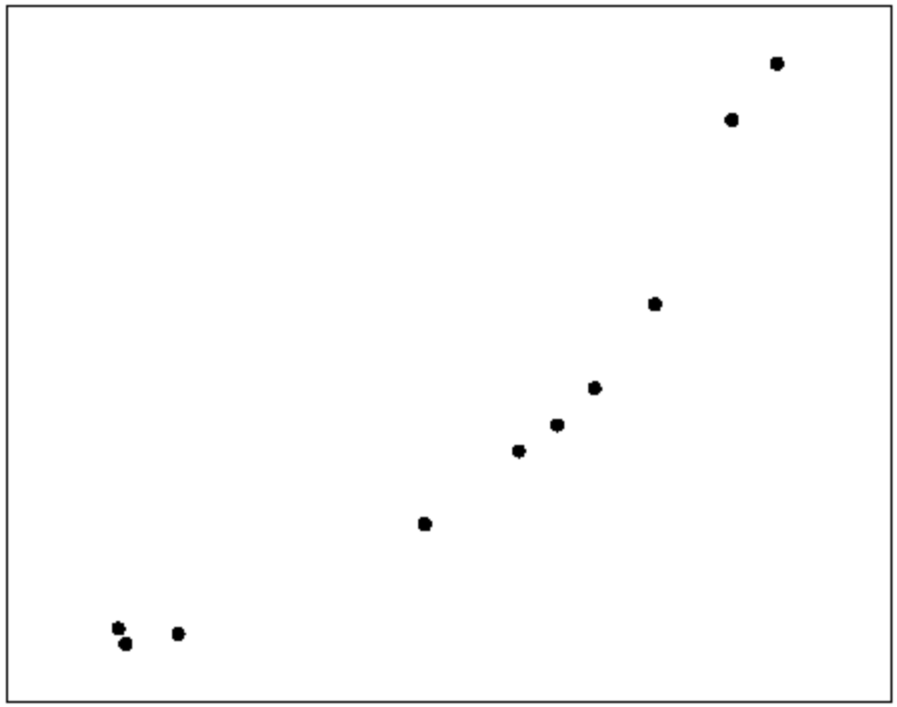
Today

- Introduction to Machine Learning
- Decision Trees
- Introduction to Probability

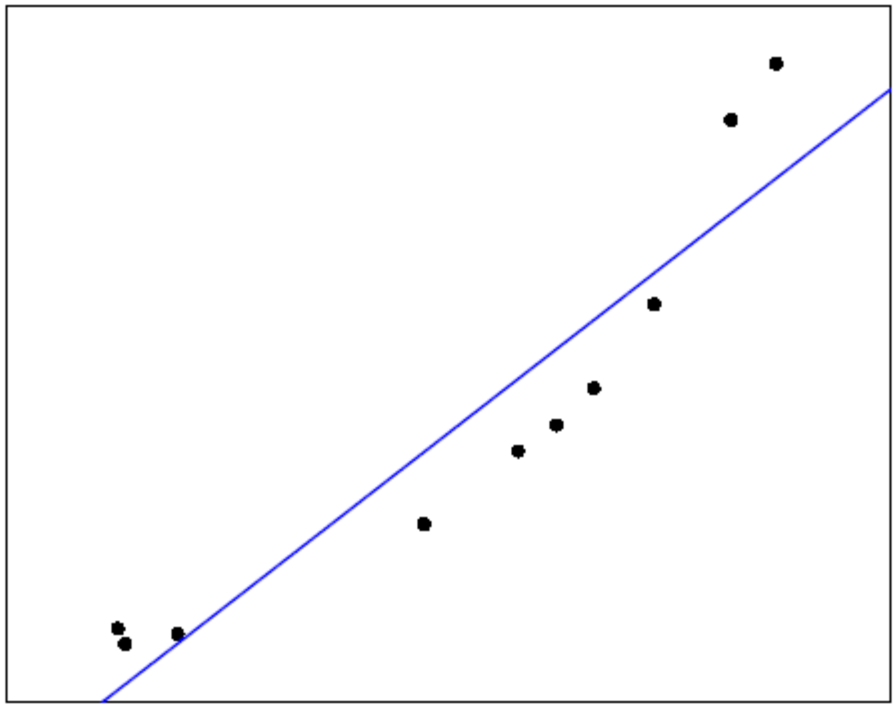
Why Learning Functions Automatically?

- Anything is a function
  - From the planet motion
  - To the input/output actions of your computer
- Therefore, any problem would be automatically solved

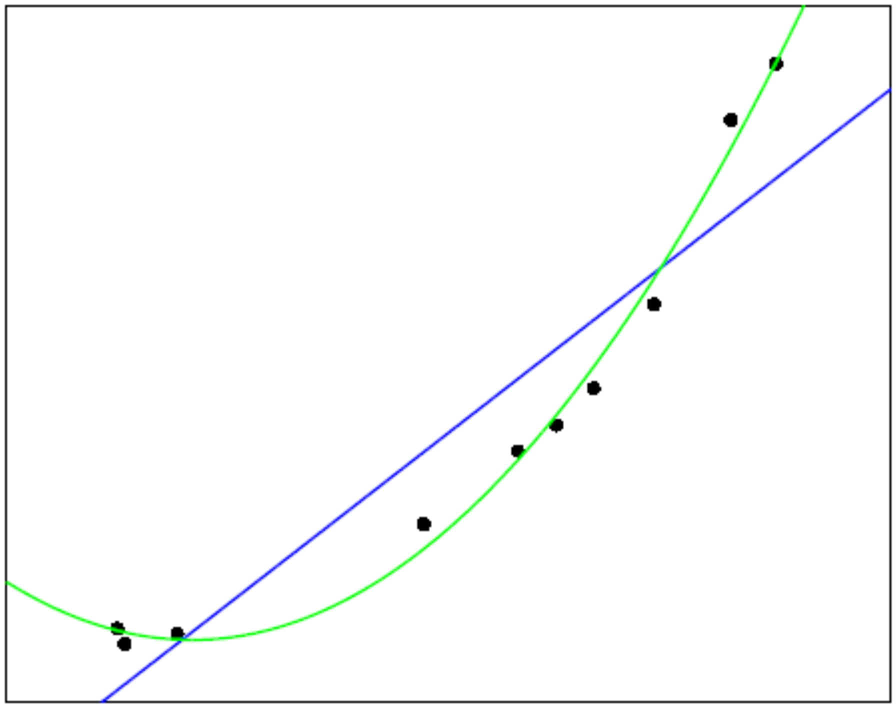
During your previous studies you have already tackled the learning problem



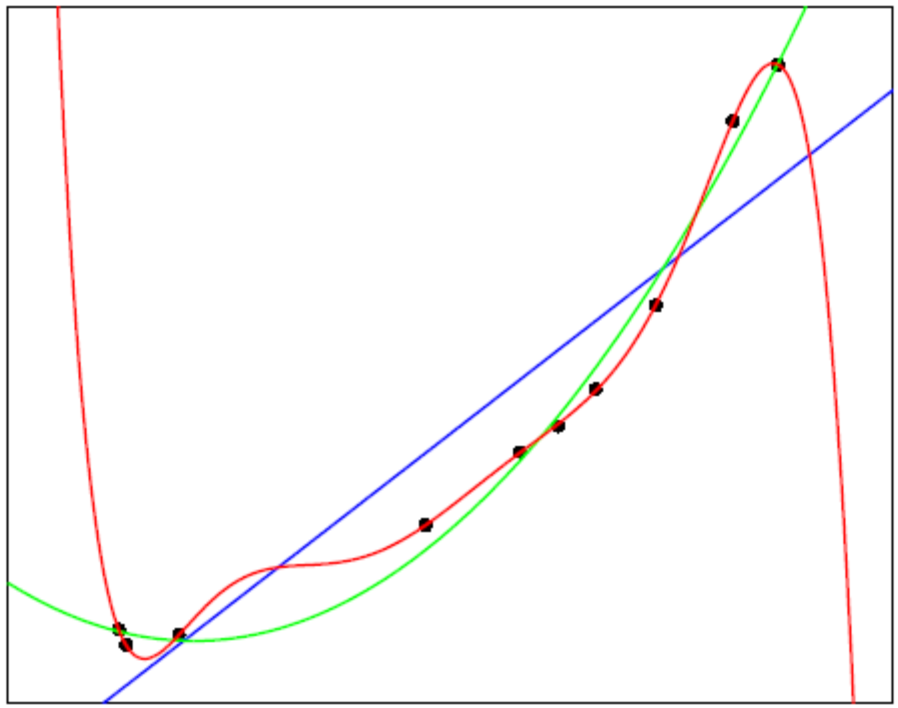
Linear Regression



Degree 2



Degree



Automatic Learning

- Real Values: *regression*
- Finite and integer: *classification*
- Suppose to design the classification function for cat and dog categories:
  - 2 classes
  - *f(x)* → {cats,dogs}
- Given a set of examples of the two categories:
  - Features are extracted (height, mustaches, tooth type, number of legs).
  - Apply a learning algorithm

Basic Learning Concepts

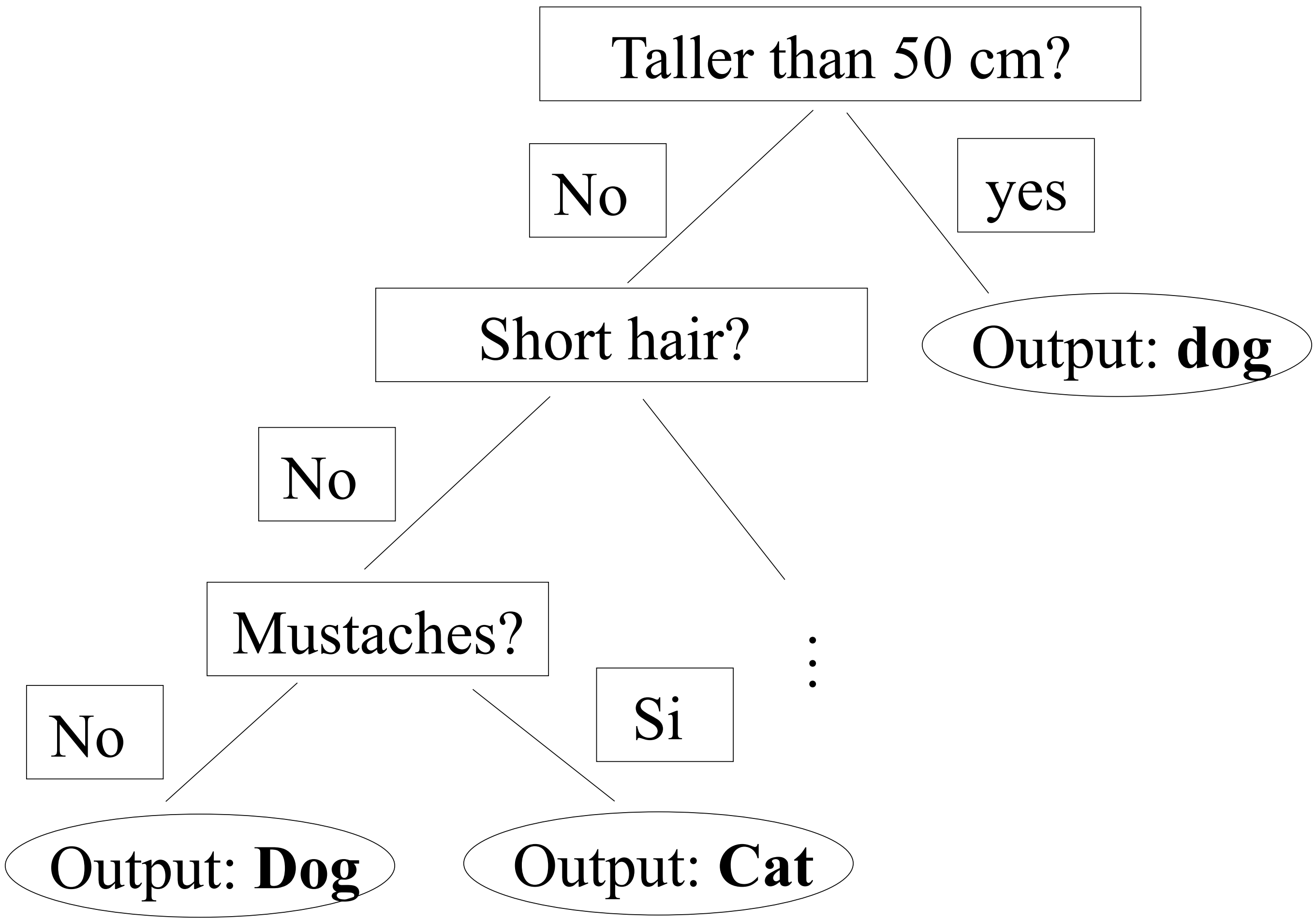
- Positive and Negative examples
- Feature representation
- Learning Algorithm
- Training and test set
- Accuracy measurement
- Can we learn any function?
- Statistical Learning Theory
  - PAC learning

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

Decision Trees

Decision Tree (between Dogs/Cats)



Entropy-based feature selection

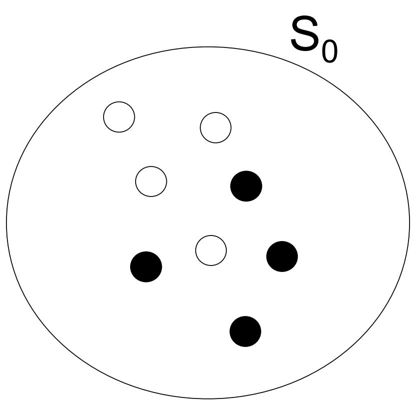
- Entropy of class distribution $P(C_i)$:

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

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

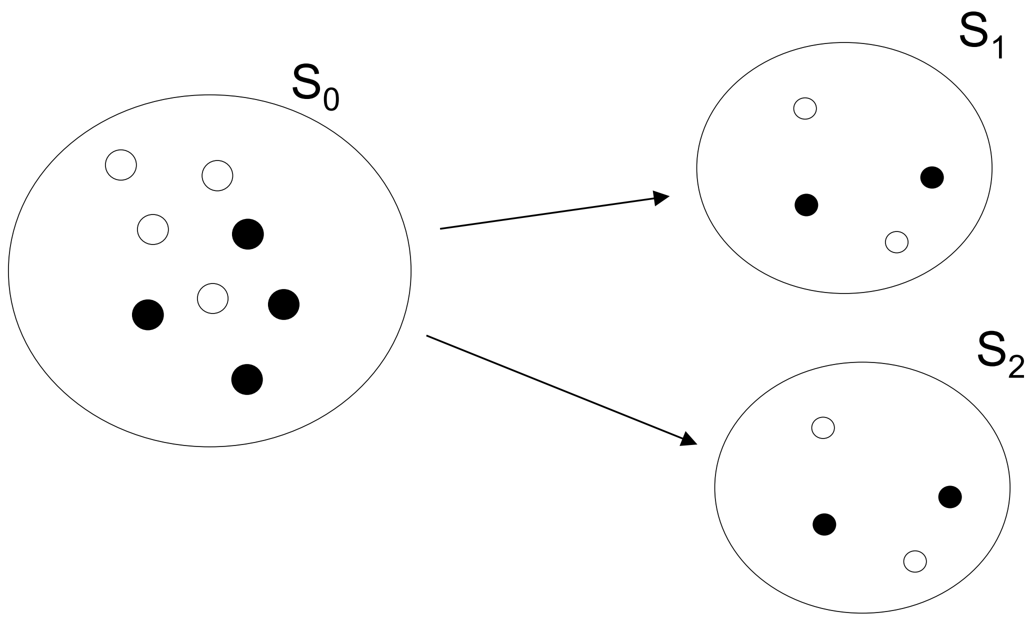
$$\bar{H}(P^{S_1}, .., P^{S_n}) = \sum_{i=1}^{m} \frac{H(P^{S_i})}{|S_i|}$$

Example: cats and dogs classification



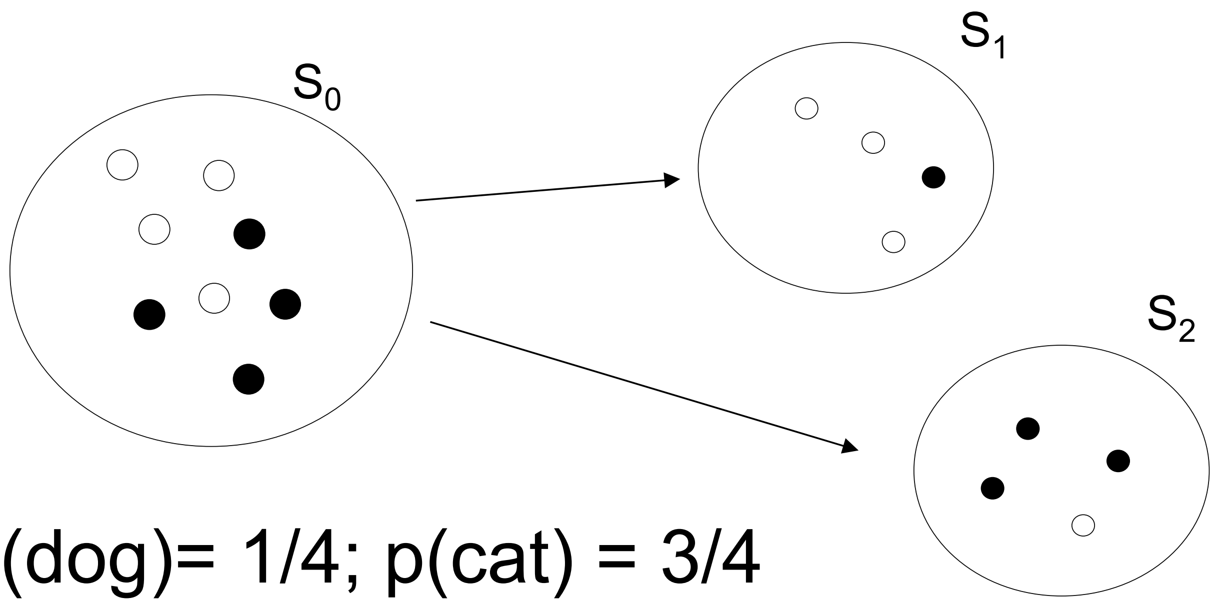
- p(dog)=p(cat) = 4/8 = ½ (for both dogs and cats)
- $H(S_0)$ = ½*log(2) * 2 = 1

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
- $H(S_2)=H(S_1) = ¼ * [(1/4)*\log(4) + (3/4)*\log(4/3)] = ¼ * [½ + 0.31] = ¼ * 0.81 = 0.20$
- $All(S_1,S_2) = 0.20*2 = 0.40$ (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:

$$P : \beta \rightarrow \lfloor 0,1 \rfloor$$

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:

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

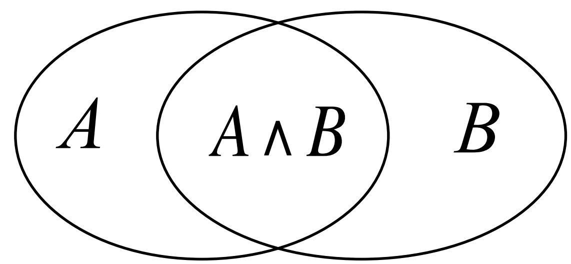
$$\sum_i P(E \wedge E_i) = \sum_{E_i \subset E} P(E_i) = \sum_{E_i \subset E} \frac{1}{n} =$$

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

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:

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

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

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 $P(e_j \mid c_i)$ 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
$P(\text{sneeze}\mid c_i)$	0.1	0.9	0.9
$P(\text{cough}\mid c_i)$	0.1	0.8	0.7
$P(\text{fever}\mid c_i)$	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={sneeze, cough, ¬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, $\forall c_i : P(e_k \mid c_i) = 0$.

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:

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

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

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** $\in$ *V*), and
- an operation *scalar* ***multiplication*** in V, denoted *a* * **v** (where **v** $\in$ V and *a* $\in$ F),
- the following properties hold for all *a*, *b* $\in$ F and **u**, **v**, and **w** $\in$ 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, [v1,v2,v3,...,vn] where vi, for i={1,2,3,...n} 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=[v1+w1,v2+w2,v3+w3,...,vn+wn].
  Define Scalar Multiplication:
  For all a in **F** and v in **V**, a*v=[a*v1,a*v2,a*v3,...,a*vn]. 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

- If V is a vector space 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

- (**v**, **0**) = 0
- $\| v \| = \sqrt{(v, v)}$
- If (**v**, **u**) = 0, **v**,**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
- $(x_1, x_2, \ldots, x_n)\,(y_1, y_2, \ldots, y_n) = (x_1 y_1, x_2 y_2, \ldots, x_n y_n)$
- $\cos(u, v) = \dfrac{(u, v)}{\| u \| \cdot \| v \|}$

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).
- For text, cosine similarity is typically most effective.

END

Training (version 2)

- 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 pairs $<w,d>$, $w \in d \in D_i$ *and* $w \in V$.
- For each $w_j \in V$,

  n_{ij} is the number of documents of c_i containing w_j that is the number of pairs $<w_j,d>$ such that $d \in D_i$

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