Machine Learning for Lexical Information Acquisition

Núria Bel and Muntsa Padró
Universitat Pompeu Fabra
muntsa.padro@upf.edu

UNIVERSITAT POMPEU FABRA
GRUP DE TECNOLOGIES DELS RECURSOS LINGÜÍSTICS (TRL)
INSTITUT UNIVERSITARI DE LINGÜÍSTICA APLICADA

Outline

- Machine Learning: Introduction
- An example of cue based lexical classification
- Supervised Learning:
  - Naïve Bayes
  - Decision Trees
- Unsupervised Learning
  - Clustering
- Evaluation measures

Outline

- Machine Learning: Introduction
- An example of cue based lexical classification
- Supervised Learning:
  - Naïve Bayes
  - Decision Trees
- Unsupervised Learning
  - Clustering
- Evaluation measures

Machine Learning

- Machine learning is an area of artificial intelligence concerned with the study of computer algorithms that improve automatically through experience.
- The algorithms infer models and or parameters to approximately represent data.

A Machine Learns?

Tom M. Mitche

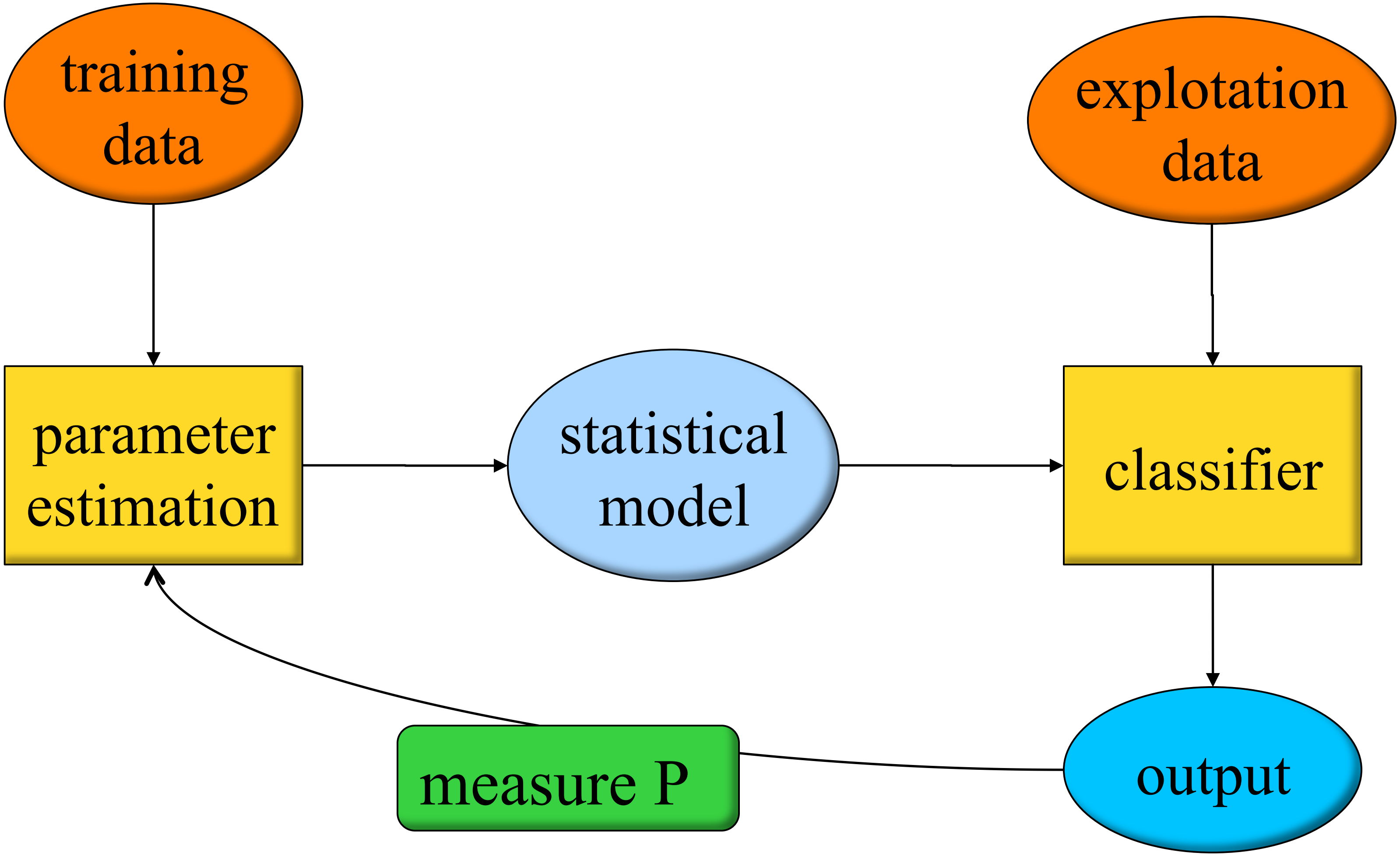
Training data

What do we want to do?

How do we evaluate the results?

"A computer program is said to learn from **experience E** with respect to some class of **tasks T** and performance **measure P,** if its performance at tasks in T, as measured by P, improves with experience E."

Developing a Classifier



Types of Machine Learning algortihms

- **Supervised learning:** The algorithm is first presented with training data which consists of examples which include both the inputs and the desired outputs, thus enabling it to learn a function. The learner should then be able to generalize from the presented data to unseen examples.
- **Unsupervised learning:** The algorithm is presented with examples from the input space only and a model is fit to these observations.
- **Semi-supervised learning:** combines both labeled and unlabeled examples to generate an appropriate function or classifier.

Outline

- Machine Learning: Introduction
- An example of cue based lexical classification
- Supervised Learning:
  – Naïve Bayes
  – Decision Trees
- Unsupervised Learning
  – Clustering
- Evaluation measures

Example: Eventive names in English

- Detection of eventive names in English: *accident* vs *family*
- Cues for eventive nouns:
  1. During
  2. After / before
  3. End / beginning
  4. Happen / begin / start /occur
  5. Subjects of verb with temporal modifiers: the cocktel lasted for 4 hours (day / month / year / second…)
  6. Initiate / carry out
  7. After frequency of, occurrence of, period of.

Example: Eventive names in English

8. external argument realized as genitive
9. external argument realized as adjective
10. singular form

- Cues for non-eventive

11. Demostrative, undefinite nor possessive determiners (usually, events do not appear with them)
12. Some / any / the whole
13. Locative preposition

Example: Eventive names in English

- We search for each cue in the sentences were the target noun occurs.

  <s>the/A666 new/JA deal/NN6S be/V6A6S able/JA to/P expropriate/VI666 the/A666 upper/JA income/NN6S bracket/ NN6P even/D6 **before/P the/A666 ##war/NN6S##**</s>

- The number of times each word occurs are stored in a vector:

	during	after	end	happen	day	carry out	genitive	adjective	singular	possessive	Locative prep	Indeterminate det
War	5	5	4	1	0	0	0	4	77	0	0	0
Family	0	0	0	0	0	0	4	39	0	16	13	0

Example: Eventive names in English

- In weka file, we add the total number of occurrences, correct class and lemma:

  164,5,5,4,1,0,0,0,4,77,0,0,0,1,war

  1434,0,0,0,0,0,0,4,39,0,16,13,0,0,family

- We can use frequencies instead of absolute counts:

  164,0.03,0.03,0.02,0.006,0,0,0,0.02,0.47,0,0,0,1,war

  1434,0,0,0,0,0,0,0.003,0.027,0,0.011,0.009,0,0,family

- This is the input for Machine Learning algorithms.

Outline

- Machine Learning: Introduction
- An example of cue based lexical classification
- Supervised Learning:
  - Naïve Bayes
  - Decision Trees
- Unsupervised Learning
  - Clustering
- Evaluation measures

Supervised Learning

- General idea of feature based classifiers:
  - Given the observed features (cues) for the word we want to classify, compute the probability of belonging to each class having seen these features:

$$P(class|n_1, \dots, n_L)$$

  - Compare these probabilities and choose as the correct class the one that is more likely (maximizes probability)

Number of times we have seen cue_j

- Examples:
  - Naïve Bayes
  - Maximum Entropy Models
  - Suport Vector Machines
  - …

UNIVERSITAT POMPEU FABRA
GRUP DE TECNOLOGIES DELS RECURSOS LINGÜÍSTICS (TRL)
INSTITUT UNIVERSITARI DE LINGÜÍSTICA APLICADA

Naïve Bayes Classifier

- A Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem

$$\boxed{P(class|n_1,\dots,n_L)} = \frac{P(n_1,\dots,n_L|class)P(class)}{\sum_k P(n_1,\dots,n_L|class_k)P(class_k)}$$

$$= \boxed{\frac{1}{Z}P(class)\prod_i P(n_i\,|class)})$$

- Independence (naïve) assumption: the presence (or absence) of a particular feature is unrelated to the presence (or absence) of any other feature:

$$P(n_1,\dots,n_L|class) = \prod_i P(n_i\,|class)$$

In our case: Bernouilli distribution

- The probability of s ined number of times n_i in N occu depends of the probability of seeing this cue in each occurrence:

Number of times we have seen the word

$$P(n_i|class) = \binom{N}{n_i} P(cue_i|class)^{n_i} \cdot (1 - P(cue_i|class))^{(N-n_i)}$$

E.g. War: Probability of seeing “duri 64, assu event.

Number of times we have seen cue_j

Number of times we

Probability of not seeing “during” cue when an eventive (or non-eventive) noun is seen.

Probability of seeing “during” cue when an eventive (or non-eventive) noun is seen.

In our case: Bernouilli distribution

- The probability of seeing a cue a determined number of times n_i in N occurrences of the word, depends of the probability of seeing this cue in each occurrence:

$$P(n_i|class) = \binom{N}{n_i} \underline{P(cue_i|class)^{n_i}} \cdot \left(1 - \underline{P(cue_i|class)}\right)^{(N-n_i)}$$

- Recall that we wanted to compute:

$$P(class|n_1, \ldots, n_L) = \frac{1}{Z} \underline{P(class)} \prod_i P(n_i\,|class)$$

- That can be expressed in terms of *P(cue$_i$|class)*, *P(class)* and observed counts *(n_i and N)*.

Parameter Estimation for Naïve Bayes

- We need to estimate:
  - *$P(cue_i|class)$* (cue likelihood)
  - *$P(class)$* (class prior)

- Two main approaches:
  - Maximum Likelihood Estimation (MLE): Compute relative frequencies from the training set
  - Bayesian modeling: Use both, frequencies of the training set and *a priori* knowledge about the system

Estimating $P(cue_i|class)$

- Relative frequency:

$$P(cue_i|class) = \frac{\textit{Number of times we have seen } cue_i \textit{ with elements of the class}}{\textit{Number of occurrences of words in the class}}$$

$$= \frac{n_i(word_1) + n_i(word_2) + \cdots + n_i(word_M)}{N(word_1) + N(word_2) + \cdots + N(word_M)}$$

Number of times we have seen *cue*$_i$ with *word*$_1$

We sum over all words that belong to the studied class

Total number of times we have seen *word*$_2$

Estimating *P(class)*

- Estimate it from the training set with MLE:

$$P(class) = \frac{Number\ of\ instances\ in\ this\ class}{Total\ number\ of\ instances}$$

- Usually equiprobable classes are assumed:

$$P(class) = \frac{1}{Number\ of\ classes}$$

Problems with MLE

- P=0 for unseen events!
- This is a big problem for sparse data
- Smoothing techniques need to be applied
- For example, Laplace Law

$$P_{Lap} = \frac{n_i(word_1) + n_i(word_2) + \cdots + n_i(word_M) + 1}{N(word_1) + N(word_2) + \cdots + N(word_M) + 2}$$

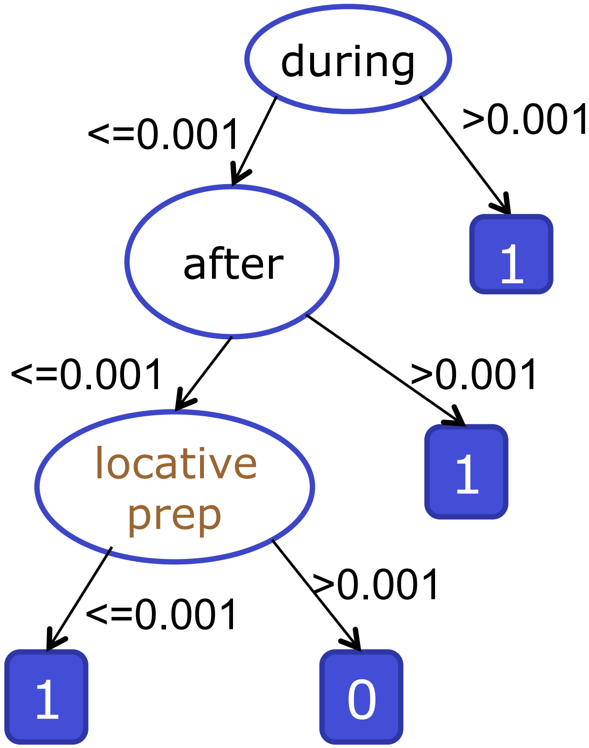
- This is not very adequate for silent data, since we are adding too many evidence to data
- More informed smoothing techniques may be more suitable.

Outline

- Machine Learning: Introduction
- An example of cue based lexical classification
- Supervised Learning:
  - Naïve Bayes
  - Decision Trees
- Unsupervised Learning
  - Clustering
- Evaluation measures

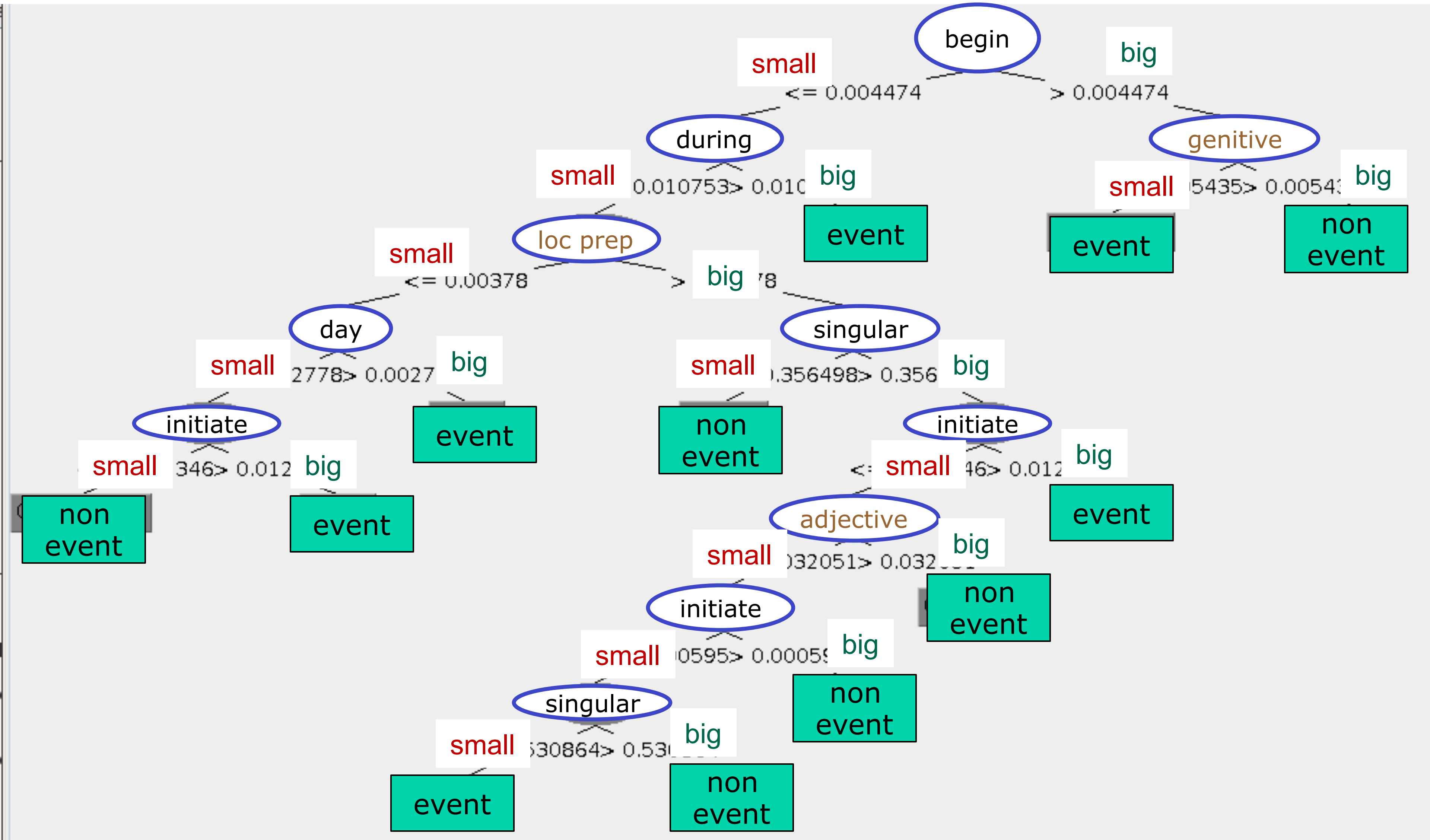
UNIVERSITAT POMPEU FABRA
GRUP DE TECNOLOGIES DELS RECURSOS LINGÜÍSTICS (TRL)
INSTITUT UNIVERSITARI DE LINGÜÍSTICA APLICADA

Decision Trees



- Each node in the tree specifies a test of some attribute of the instance.
- Each branch decending from that node corresponds to one ot he possible values for this attribute.
- To classify an instance: start at the root node, test the attribute for this node and move down the tree branch corresponding to the observed value of this attribute.

DT Example: Eventive names in English

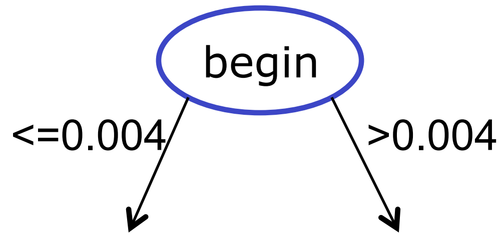


Decision Trees: Learning Algorithms

- Learning a DT: from all possible Decision Trees, choose the one that better fits the data.
- Different ML algorithms infere Decision Trees.
  - ID3 (Quinlan 1986)
  - C4.5 (Quinlan 1993), J48
  - etc
- Most of them are based on ID3

ID3 algorithm for learning DTs

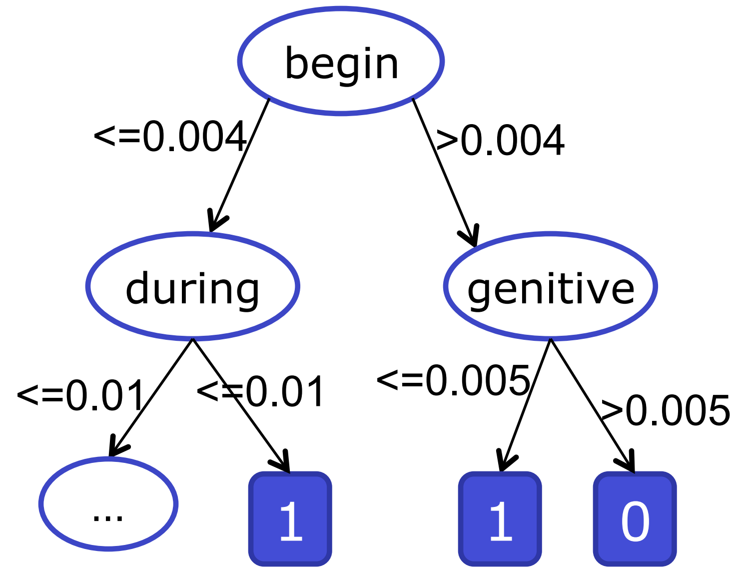
- First step: decide which attribute should be in the root node.
  - Intuitively: choose the attribute that better separates the space. Note: in our case, we need to decide also the treshold



  - Formally:
    - compute “information gain” of each attribute
    - choose the attribute that has greater information gain.

ID3 algorithm for learning DTs

- Once we know the root node, study the examples that are under each condition.
  - If all examples are in the same class: create a leaf node
  - Else: repeat process to look for the most informative attribute.



- Do it recursivelly until all examples are classified

Considerations about IB3

- It uses only the attributes that it needs to build the tree. There may be unused attributes
- No backtracking to reconsider its choices.
- Very sensible to overfitting → pruning

Pruning Trees

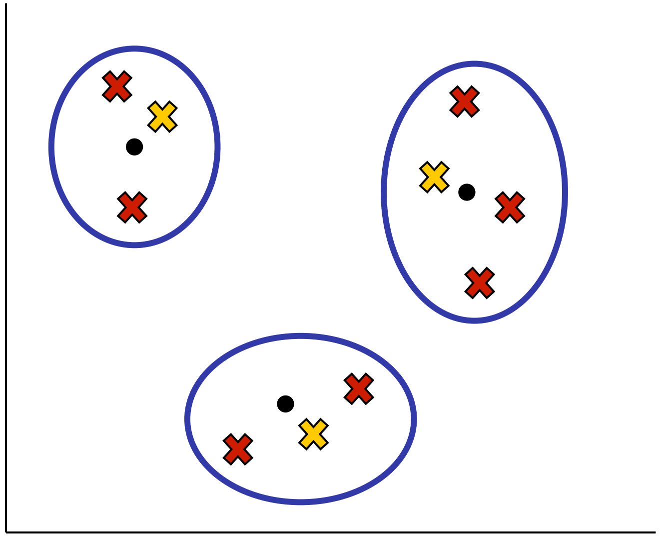
- Tool to correct for potential overfitting: relax the specificity of the decision tree
- Pruning produces fewer, more easily interpreted results.
- Pruning always reduces the accuracy of a model on training data, but tries to impove its results on test data.
- The overall concept is to gradually generalize a decision tree until it gains a balance of flexibility and accuracy.
- To do so...
  - Use a separate test set to evaluate different pruned trees independent on the training data
  - Look for leaves that represent very few instances
  - Convert Tress to rules an generalize rules
  - ...

Outline

- Machine Learning: Introduction
- An example of cue based lexical classification
- Supervised Learning:
  - Naïve Bayes
  - Decision Trees
- Unsupervised Learning
  - Clustering
- Evaluation measures

Clustering

- Clustering of data is a method by which large sets of data are grouped into clusters of smaller sets of similar data.
- Similarity measure: distance
- Representatives of the cluster: Centroid and/or medoid



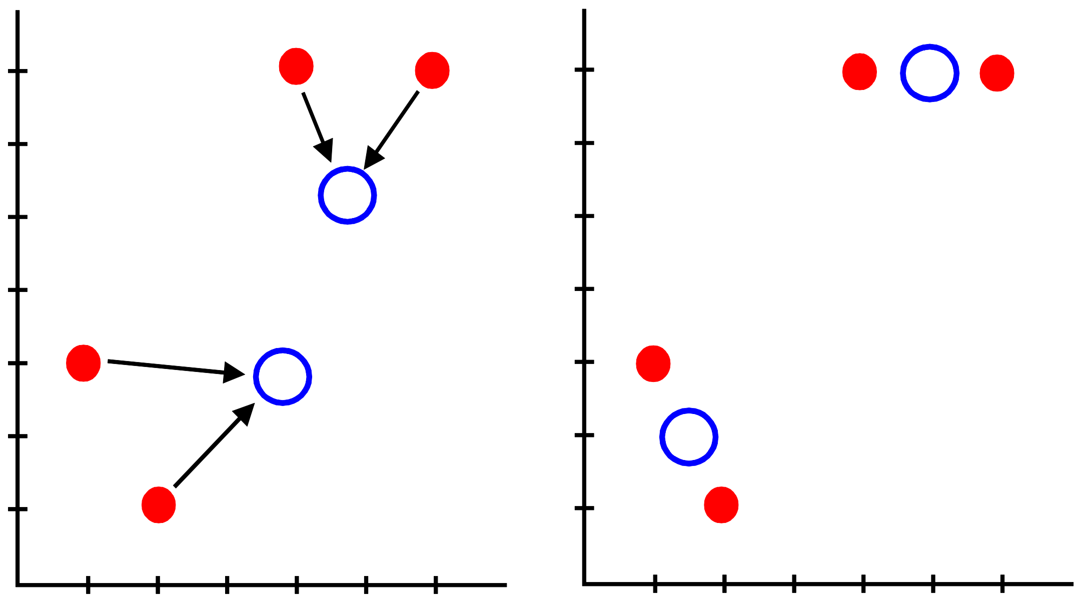
Clustering

- Utility:
  - Generalization (learning). Ex: on Monday, on Sunday, ? Friday
  - Unsupervised classification
- Object assignment to clusters
  - Hard: one cluster per object.
  - Soft: distribution $P(c_i \mid x_j)$. Degree of membership.

UNIVERSITAT POMPEU FABRA
GRUP DE TECNOLOGIES DELS RECURSOS LINGÜÍSTICS (TRL)
INSTITUT UNIVERSITARI DE LINGÜÍSTICA APLICADA

K-means: non-hierarchical cluestering

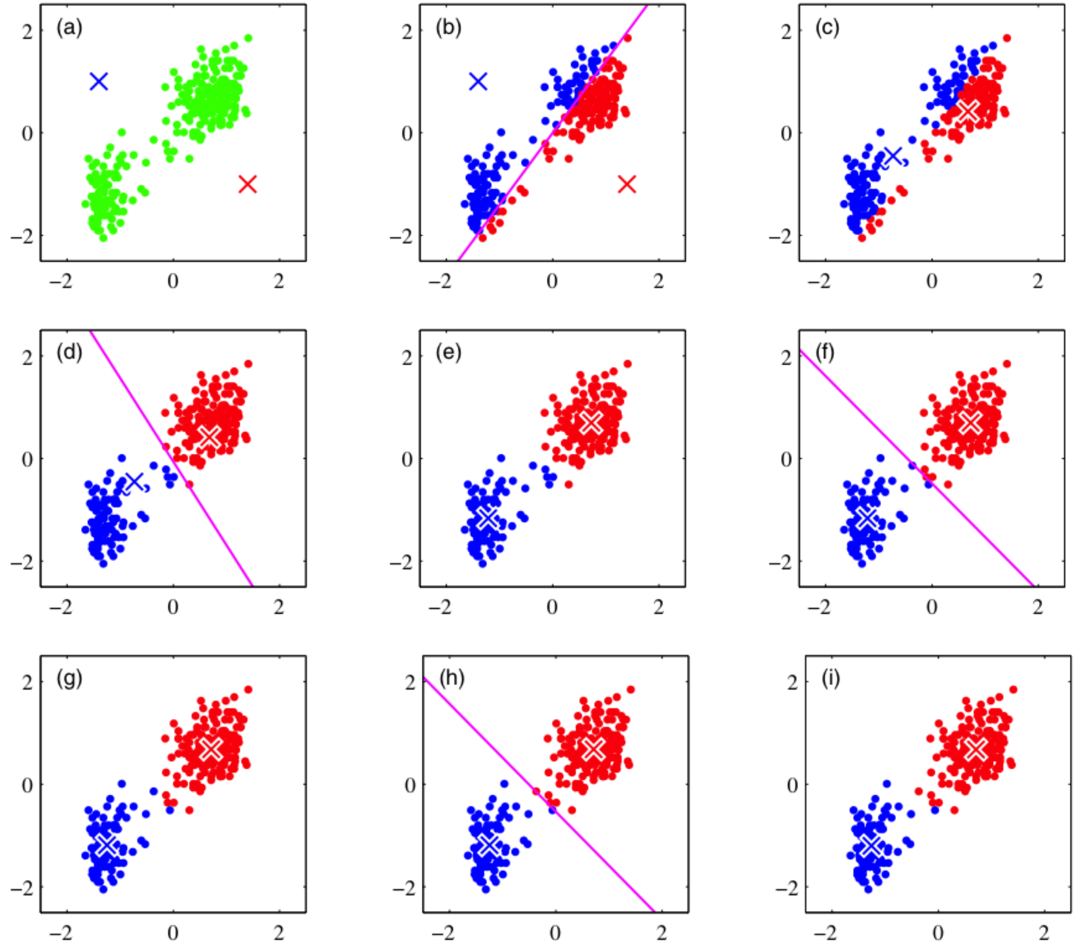
- Choose *k* initial centers
- Each element is associated to the closest center.
- Recompute the centroid of the cluster and repeat.



Assignment

Recomputation of means

K-means: Example



Outline

- Machine Learning: Introduction
- An example of cue based lexical classification
- Supervised Learning:
  - Naïve Bayes
  - Decision Trees
- Unsupervised Learning
  - Clustering
- Evaluation measures

Evaluation Measures

- To evaluate the results: count how many instances the system classifies correctly over the total number of instances.

- Accuracy: $$Acc = \frac{correctly\ classified\ instances}{total\ number\ of\ instances}$$

- To better study the errors:
  - Fa… classified as members of the class
  - False negatives: members of the class classified as outside the class

Accuracy may be computed for all instances or separatelly for each class

Evaluation Measures: Precision and Recall

- Precision and Recall are computed for each class
- Precision: "correctness" of the instances we have classified.

$$P = \frac{class\ instances\ correctly\ classified}{proposed\ class\ instances}$$

- Recall: how many of the correct instances did we classified?

$$R = \frac{class\ instances\ correctly\ classified}{gold - standard\ class\ instances}$$

- F1: combines P and R with harmonic mean

$$F_1 = \frac{2 \cdot P \cdot R}{P + R}$$

Evaluation Measures: Examples

$$P = \frac{class\ instances\ correctly\ classified}{proposed\ class\ instances}$$

$$R = \frac{class\ instances\ correctly\ classified}{gold-standard\ class\ instances}$$

- Supose a gold-s eventive nouns.
- Supose two classifi
  - Classifier 1: cl
  - Classifier 2: classifies one instance as eventive (and it is correct) and all other instances as non-eventive.

Cla-ssifier	Event	Non event	Total ok	Total ok event	Total ok non-event	Accu-racy	FP	FN	P event	R event	P non-event	R non-event
All event	100	0	50	50	0	50%	50	0	50%	100%	0%	0%
One event	1	99	51	1	50	51%	0	49	100%	2%	50.5%	100%

Hands-on exercice

https://sites.google.com/site/cuebasedlia/

Thank you!

muntsa.padro@upf.edu

Bibliography

Tom M. Mitchell (1997). Machine Learning, McGraw Hill. ISBN 0-07-042807-7.