Special Topics in Text Mining

Manuel Montes y Gómez

http://ccc.inaoep.mx/~mmontesg/

mmontesg@inaoep.mx

Instituto Nacional de Astrofísica, Óptica y Electrónica, Mexico

General agenda

- Introduction to text classification
- Beyond the BoW representation
- Non conventional classification methods
- Non thematic text classification applications
- Introduction to document clustering

Introduction to text classification

Agenda

- The problem of text classification
- Machine learning approach for TC
- Construction of a classifier
  - Document representation
  - Dimensionality reduction
  - Classification methods
- Evaluation of a TC method
- Description of the module project

Classification

Given a universe of objects and a pre-defined set of classes assign each object to its correct class

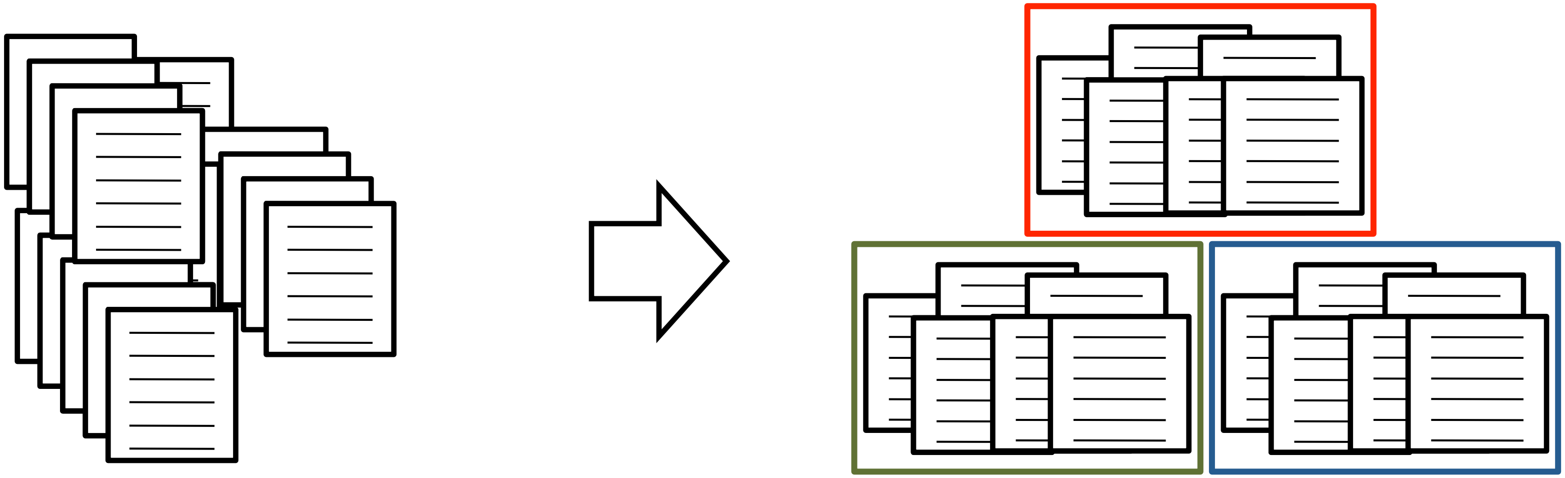
- Input:
  - A description of an instance, $x \in X$, by a vector of measurements; where X is the instance space.
  - A fixed set of categories: $C=\{c_1, c_2, \ldots c_n\}$
- Output:
  - The category of x: $c(x) \in C$, where $c(x)$ is a categorization function whose domain is X and whose range is C.

Textual-related tasks

Problem	Objects (instances)	Categories
Tagging	words in context	POS tags
WSD	words in context	word senses
PP attachment	sentences	parse trees
Language identification	Text	languages
Text classification	**documents**	**topics**

Text classification

- It is the assignment of free-text documents to one or more predefined categories based on their content.



Documents (e.g., news articles)

Categories/classes
(e.g., sports, religion, economy)

Text classification applications

- Journal articles indexed by subject categories
- Patents archived using International Patent Classification
- Patient records coded using international insurance categories

Other applications?

- E-mail messages filtering (spam detection)
- Product reviews organized by polarity

Other non-thematic applications?

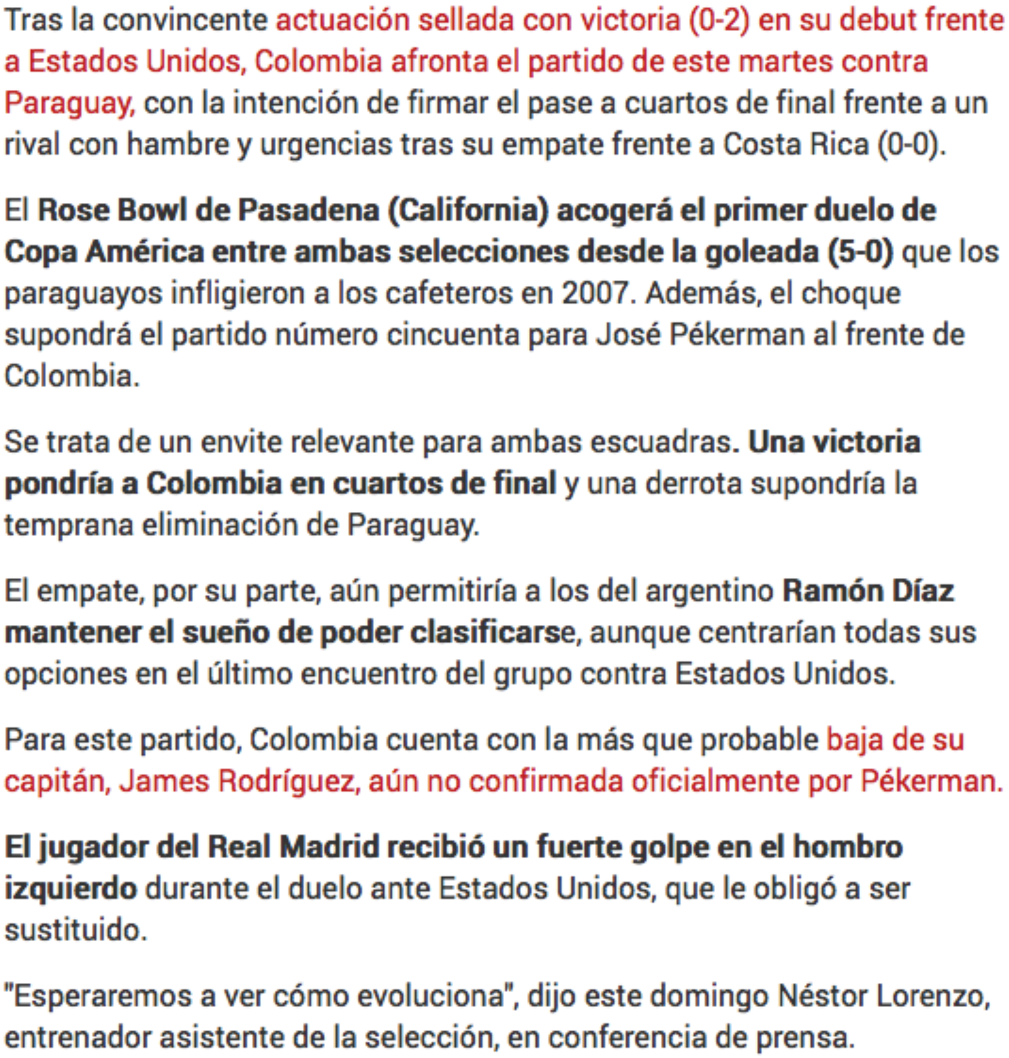
Manual classification

- Very accurate when job is done by experts
  - Different to classify news in general categories than biomedical papers into subcategories.
- But difficult and expensive to scale
  - Different to classify thousands than millions
- Used by Yahoo!, Looksmart, about.com, ODP, Medline, etc.

Ideas for building an automatic classification system?

How to define the classification function?

What is the topic of this document?



Tras la convincente actuación sellada con victoria (0-2) en su debut frente a Estados Unidos, Colombia afronta el partido de este martes contra Paraguay, con la intención de firmar el pase a cuartos de final frente a un rival con hambre y urgencias tras su empate frente a Costa Rica (0-0).

El **Rose Bowl de Pasadena (California) acogerá el primer duelo de Copa América entre ambas selecciones desde la goleada (5-0)** que los paraguayos infligieron a los cafeteros en 2007. Además, el choque supondrá el partido número cincuenta para José Pékerman al frente de Colombia.

Se trata de un envite relevante para ambas escuadras. **Una victoria pondría a Colombia en cuartos de final** y una derrota supondría la temprana eliminación de Paraguay.

El empate, por su parte, aún permitiría a los del argentino **Ramón Díaz mantener el sueño de poder clasificarse**, aunque centrarían todas sus opciones en el último encuentro del grupo contra Estados Unidos.

Para este partido, Colombia cuenta con la más que probable baja de su capitán, James Rodríguez, aún no confirmada oficialmente por Pékerman.

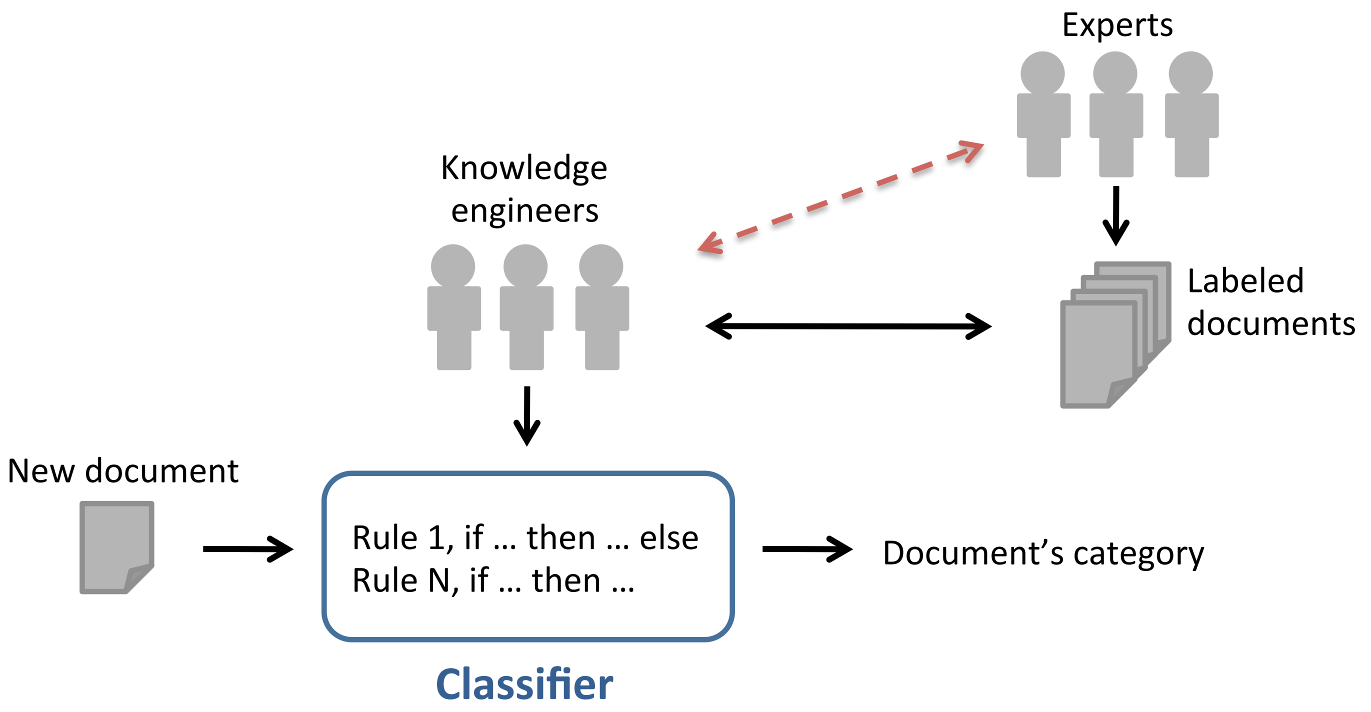
El jugador del Real Madrid recibió un fuerte golpe en el hombro izquierdo durante el duelo ante Estados Unidos, que le obligó a ser sustituido.

"Esperaremos a ver cómo evoluciona", dijo este domingo Néstor Lorenzo, entrenador asistente de la selección, en conferencia de prensa.

Sports?
Religion?
Music?

Did you need to read it?
Did you require to understand it?
So, how to automate this process?

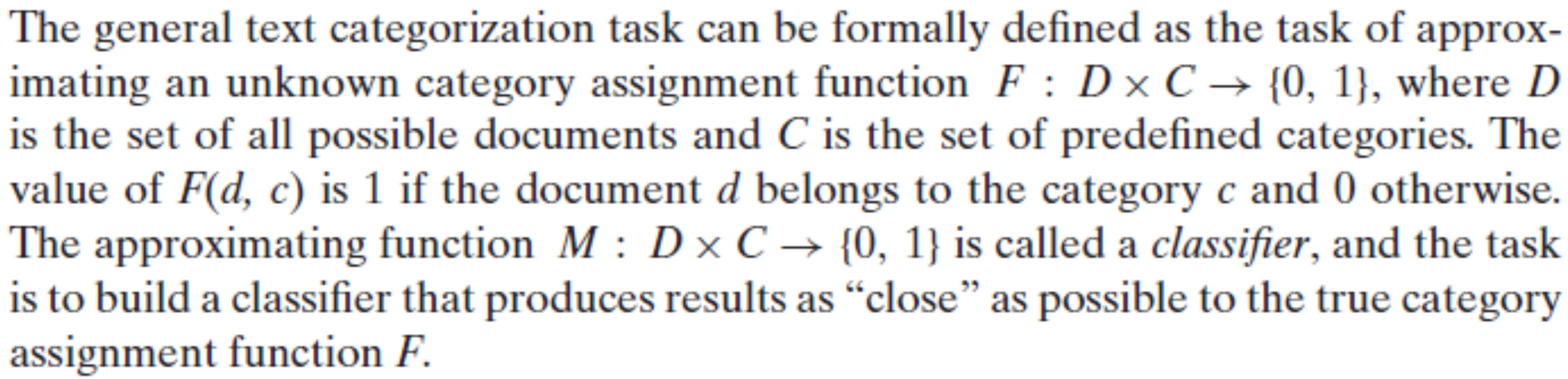
Hand-coded rule based systems



- Main approach in the 80s
- Disadvantage → knowledge acquisition bottleneck
  - too time consuming, too difficult, inconsistency issues

Machine learning approach (1)

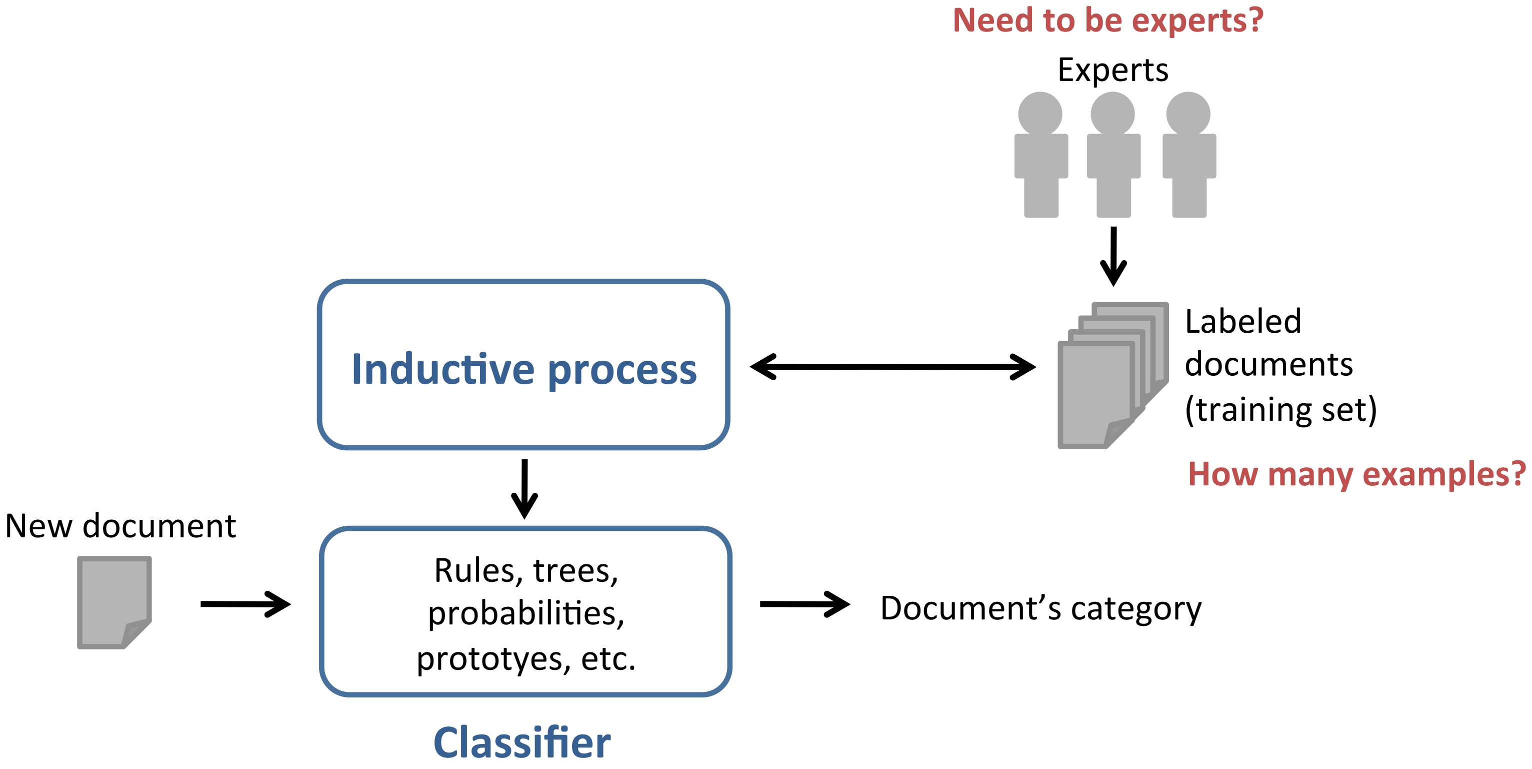
- A general inductive process builds a classifier by learning from a set of preclassified examples.
  - Determines the characteristics associated with each one of the topics.



The general text categorization task can be formally defined as the task of approximating an unknown category assignment function $F : D \times C \to \{0, 1\}$, where D is the set of all possible documents and C is the set of predefined categories. The value of $F(d, c)$ is 1 if the document d belongs to the category c and 0 otherwise. The approximating function $M : D \times C \to \{0, 1\}$ is called a *classifier*, and the task is to build a classifier that produces results as "close" as possible to the true category assignment function F.

Ronen Feldman and James Sanger, The Text Mining Handbook

Machine learning approach (2)



How to represent documents?

Representation of documents

- First step is to transform documents, which typically are strings of characters, into a representation suitable for the learning algorithm.
- The most common used document representation is the *bag of words*.
  - Documents are represent by the set of different words in all of the documents
  - Word order is not capture by this representation
  - There is no attempt for understanding their content

Representation of documents

Vocabulary from the collection
(*set of different words*)

	t_1	t_1	…	t_n
d_1				
d_2				
:		$w_{i,j}$		
d_m				

All documents
(*one vector per document*)

Weight indicating the contribution of word *j* in document *i*.

Which words are good features?
How to select/extract them?
How to compute their weights?

Preprocessing

- Eliminate information about style, such as html or xml tags.
  - For some applications this information may be useful. For instance, only index some document sections.
- Remove stop words
  - Functional words such as articles, prepositions, conjunctions are not useful (do not have an own meaning).
- Perform stemming or lemmatization
  - The goal is to reduce inflectional forms, and sometimes derivationally related forms.

am, are, is → be
car, cars, car's → car

Term weighting - two main ideas

- The importance of a term increases proportionally to the number of times it appears in the document.
  - It helps to *describe* document's content.
- The general importance of a term decreases proportionally to its occurrences in the entire collection.
  - Common terms are not good to *discriminate* between different classes

Term weighting – main approaches

- Binary weights:
  - $w_{i,j}$ = 1 iff document d_i contains term t_j , otherwise 0.
- Term frequency (tf):
  - $w_{i,j}$ = (no. of occurrences of t_j in d_i)
- tf x idf weighting scheme:
  - $w_{i,j} = \mathrm{tf}(t_j, d_i) \times \mathrm{idf}(t_j)$, where:
    - $\mathrm{tf}(t_j, d_i)$ indicates the ocurrences of t_j in document d_i
    - $\mathrm{idf}(t_j) = \log [N/\mathrm{df}(t_j)]$, where $\mathrm{df}(t_j)$ is the number of documets that contain the term $t_{j.}$

Need of normalization? How to do it?

Extended document representations

- BOW is simple and tend to produce good results, but it has important limitations
  - Does not capture word order neither semantic information
- New representations attempt to handle these limitations. Some examples are:
  - Distributional term representations
  - Locally weighted bag of words
  - Bag of concepts
  - Concise semantic analysis
  - Latent semantic indexing
  - Topic modeling
  - ..

We are going to talk about some of them

Dimensionality reduction

- A central problem in text classification is the high dimensionality of the feature space.
  - Exist one dimension for each unique word found in the collection → can reach hundreds of thousands
  - Processing is extremely costly in computational terms
  - Most of the words (features) are irrelevant to the categorization task

How to select/extract relevant features?

How to evaluate the relevance of the features?

Two main approaches

- Feature selection
  - Idea: removal of non-informative words according to corpus statistics
  - Output: subset of original features
  - Main techniques: document frequency, mutual information and information gain
- Re-parameterization
  - Idea: combine lower level features (words) into higher-level orthogonal dimensions
  - Output: a new set of features (not words)
  - Main techniques: word clustering and Latent semantic indexing (LSI)

Document frequency

- The document frequency for a word is the number of documents in which it occurs.
- This technique consists in the removal of words whose document frequency is less than some specified threshold
- The basic assumption is that *rare words* are either *non-informative* for category prediction or not influential in global performance.

Mutual information

- Measures the *mutual dependence* of the two variables
  - In TC, it measures the information that a word *t* and a class *c* share: how much knowing word *t* reduces our uncertainty about class *c*

$$I(t,c) = \log \frac{P_r(t \wedge c)}{P_r(t) \times P_r(c)}$$

The idea is to select words that are very related with one single class

$$I_{max}(t) = \max_{i=1}^{m}\{I(t,c_i)\}$$

Information gain (1)

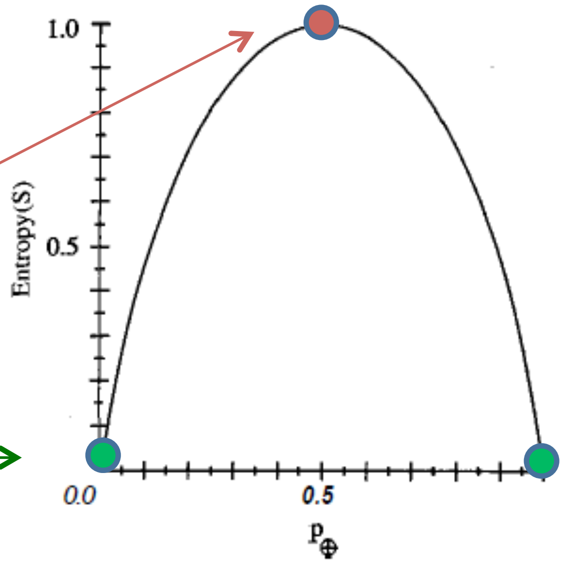
- Information gain (IG) measures how well an attribute separates the training examples according to their target classification
  - Is the attribute a good classifier?
- The idea is to select the set of attributes having the greatest IG values
  - Commonly, maintain attributes with IG > 0

How to measure the worth (IG) of an attribute?

Information gain (2)

- Information gain → *Entropy*
- Entropy characterizes the impurity of an arbitrary collection of examples.
  - It specifies the minimum number of bits of information needed to *encode the classification* of an arbitrary member of the dataset (*S*).
- For a binary problem:

$$Entropy(S) \equiv -p_{\oplus}\log_2 p_{\oplus} - p_{\ominus}\log_2 p_{\ominus}$$



Greatest uncertainty
1 bit to encode the class

No uncertainty
always positive/negative
not need to encode the class

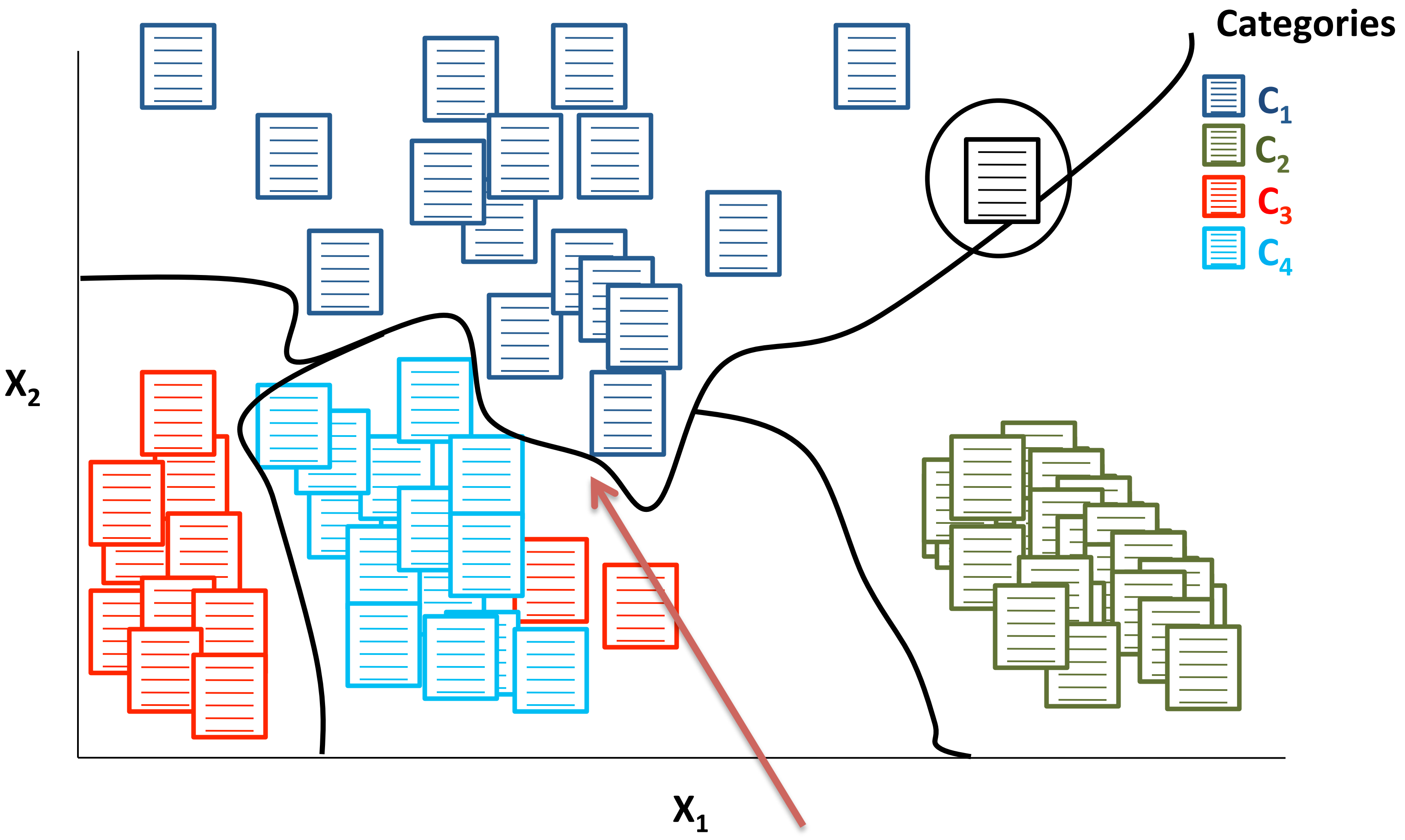
Information gain (3)

$$Entropy(S) \equiv \sum_{i=1}^{c} -p_i \log_2 p_i$$

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

- IG of an attribute measures the expected reduction in entropy caused by partitioning the examples according to this attribute.
  - The greatest the IG, the better the attribute for classification
  - IG < 0 indicates that we have a problem with greater uncertainty than the original
  - The maximum value is log *C*; *C* is the number of classes.

Learning the classification model



How to learn this?

Classification algorithms

- Popular classification algorithms for TC are:
  - Naïve Bayes
    - *Probabilistic* approach
  - K-Nearest Neighbors
    - *Example-based* approach
  - Centroid-based classification
    - *Prototype*-based approach
  - Support Vector Machines
    - *Kernel*-based approach

Naïve Bayes

- It is the simplest probabilistic classifier used to classify documents
  - Based on the application of the Bayes theorem
- Builds a *generative model* that approximates how data is produced
  - Uses *prior* probability of each category given no information about an item.
  - Categorization produces a *posterior* probability distribution over the possible categories given a description of an item.

Bayes’ Rule for text classification

- For a document *d* and a class c_j

$$P(c_j \mid d) = \frac{P(d \mid c_j)P(c_j)}{P(d)} \longrightarrow P(c_j \mid d) = P(c_j)\prod_{i=1}^{M} P(t_i \mid c_j)$$

- Estimation of probabilities

Smoothing to avoid zero-values

$$P(c_j) = \frac{N_j}{N} \qquad P(t_i \mid c_j) = \frac{1 + N_{ij}}{M + \sum_{k=1}^{M} N_{kj}}$$

Prior probability of class c_j

Probability of occurrence of word t_i in class c_j

Naïve Bayes classifier

- Assignment of the class:

$$class = \arg\max_{c_j \in C} P(c_j | d) = \arg\max_{c_j \in C} P(c_j) \prod_{i=1}^{M} P(t_i | c_j)$$

- Assignment using underflow prevention:
  - Multiplying lots of probabilities can result in floating-point underflow
  - Since log(*xy*) = log(*x*) + log(*y*), it is better to perform all computations by summing logs of probabilities rather than multiplying probabilities

$$class = \underset{c_j \in C}{\mathrm{argmax}} \left[\log P(c_j) + \sum_{i=1}^{M} \log P(t_i | c_j) \right]$$

Comments on NB classifier

- Very simple classifier which works very well on numerical and textual data.
- Very easy to implement and computationally cheap when compared to other classification algorithms.
- One of its major limitations is that it performs very poorly when features are highly correlated.
- Concerning text classification, it fails to consider the frequency of word occurrences in the feature vector.

KNN – initial ideas

- Do not build explicit declarative representations of categories.
  - This kind of methods are called lazy learners
- "Training" for such classifiers consists of simply storing the representations of the training documents together with their category labels.
- To decide whether a document *d* belongs to the category *c,* kNN checks whether the *k* training documents most similar to *d* belong to *c.*
  - Key element: a definition of "similarity" between docuemnts

KNN – the algorithm

- Given a new document *d*:
  1. Find the *k* most similar documents from the training set.
     - Common similarity measures are the cosine similarity and the Dice coefficient.
  2. Assign the class to *d* by considering the classes of its *k* nearest neighbors
     - Majority voting scheme
     - Weighted-sum voting scheme

Common similarity measures

- Dice coefficient

$$s\left(d_i, d_j\right) = \frac{2\sum_{k=1}^{n}\left(w_{ki} \times w_{kj}\right)}{\sum_{k=1}^{m} w_{ki}^{2} + \sum_{k=1}^{m} w_{kj}^{2}}$$

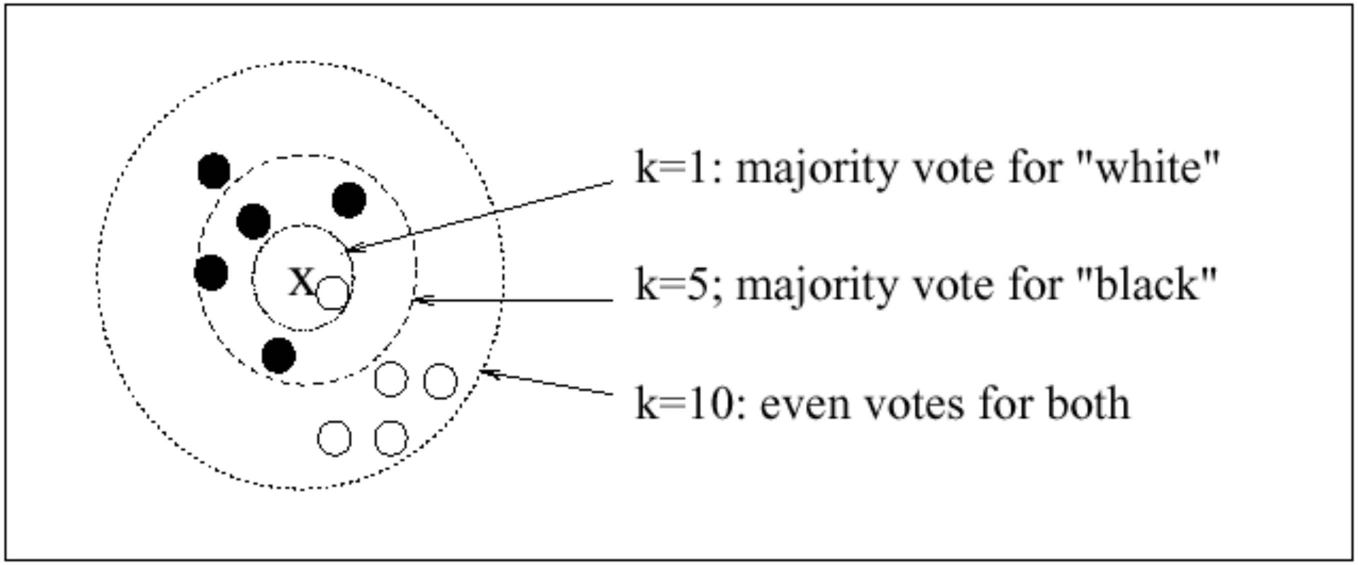
- Cosine measure

$$s\left(d_i, d_j\right) = \frac{\sum_{k=1}^{n}\left(w_{ki} \times w_{kj}\right)}{\sqrt{\sum_{k=1}^{m} w_{ki}^{2}} \times \sqrt{\sum_{k=1}^{m} w_{kj}^{2}}}$$

w_{ki} indicates the weight of word *k* in document *i*

Selection of *K*

K-Nearest Neighbor using a *majority* voting scheme



k=1: majority vote for "white"

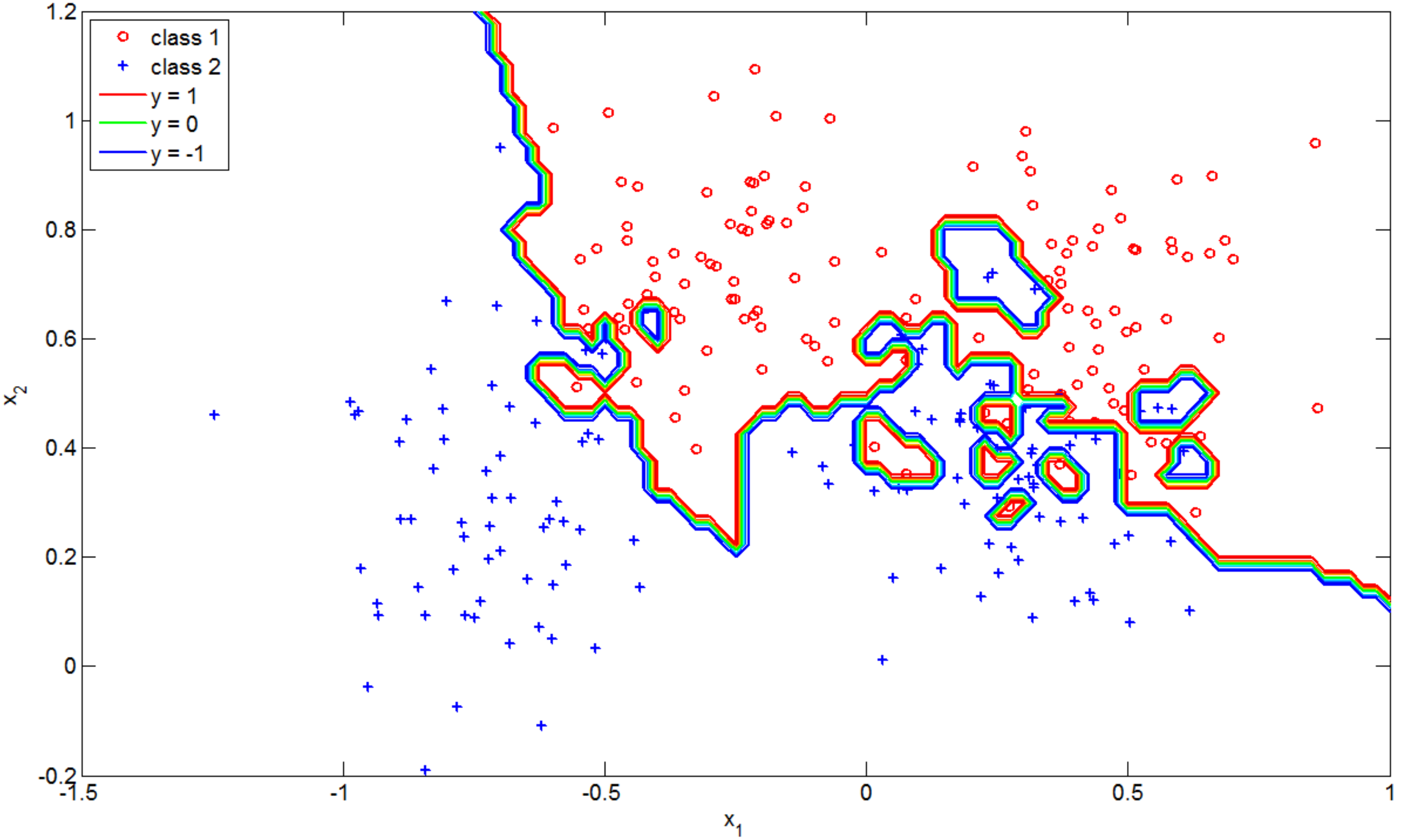
k=5; majority vote for "black"

k=10: even votes for both

How to select a good value for *K*?

Decision surface of KNN

http://clopinet.com/CLOP

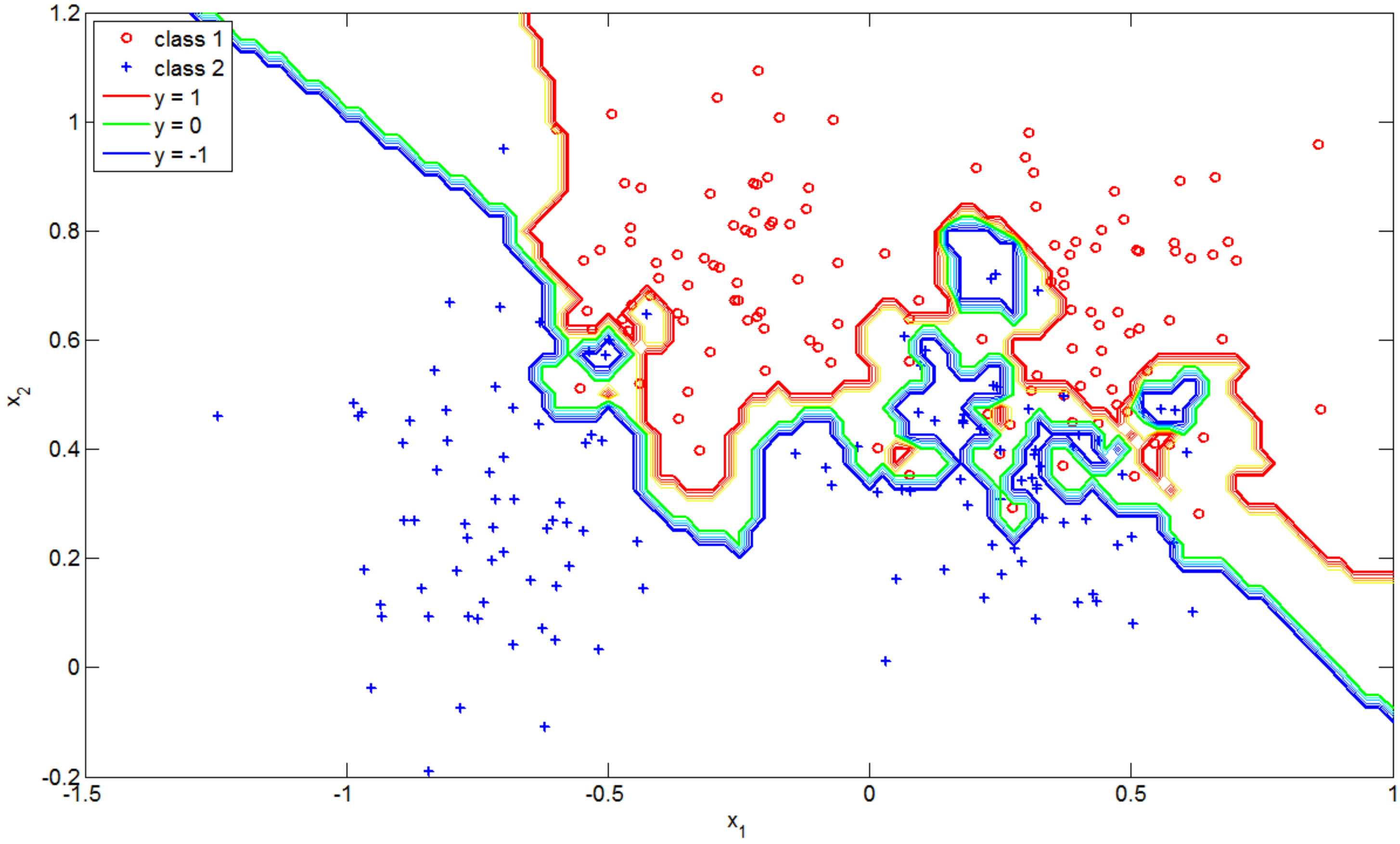


K=1

Laboratorio de
Tecnologías del Lenguaje
Ciencias Computacionales, INAOE

Decision surface of KNN

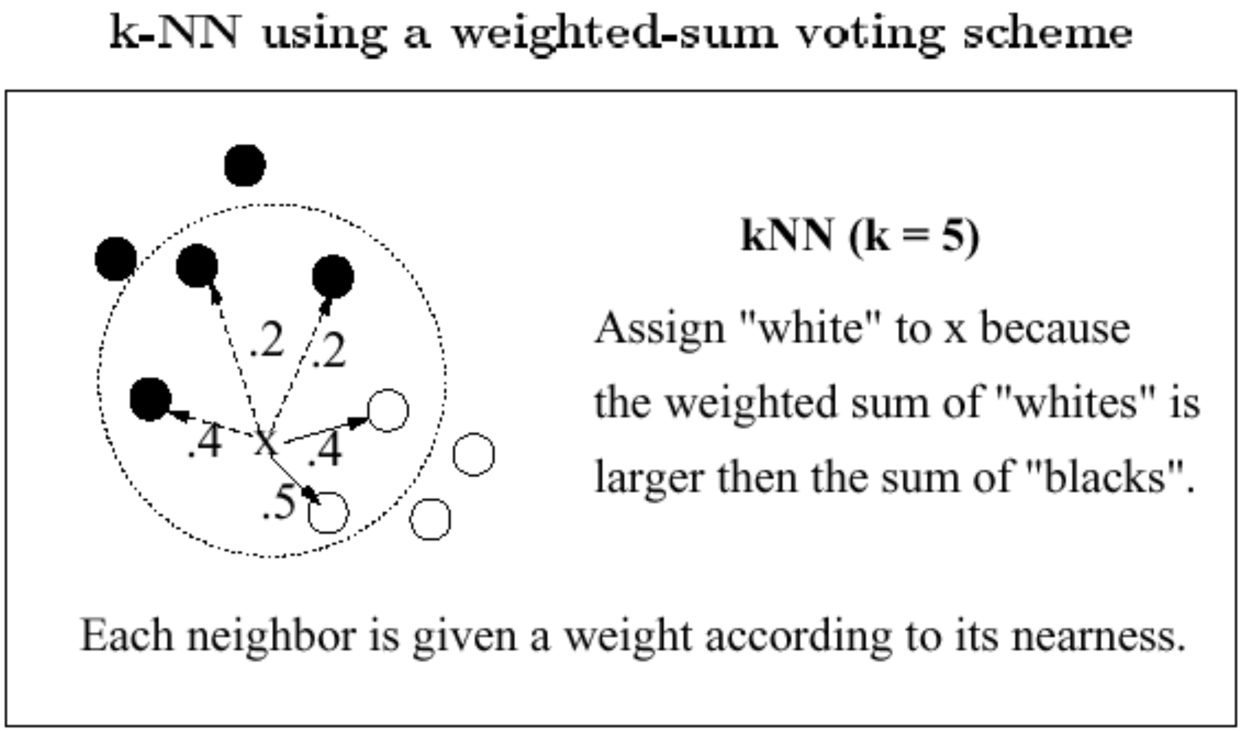
http://clopinet.com/CLOP



K=2

Laboratorio de
Tecnologías del Lenguaje
Ciencias Computacionales, INAOE

The weighted-sum voting scheme



k-NN using a weighted-sum voting scheme

kNN (k = 5)

Assign "white" to x because the weighted sum of "whites" is larger then the sum of "blacks".

Each neighbor is given a weight according to its nearness.

Other alternatives for computing the weights?

KNN - comments

- One of the best-performing text classifiers.
- It is robust in the sense of not requiring the categories to be linearly separated.
- The major drawback is the computational effort during classification.
- Other limitation is that its performance is primarily determined by the choice of *k* as well as the distance metric applied.

Centroid-based classification

- This method has two main phases:
  - Training phase: it considers the construction of one single representative instance, called prototype, for each class.
  - Test phase: each unlabeled document is compared against all prototypes and is assigned to the class having the greatest similarity score.
- Different from k-NN which represent each document in the training set individually.

How to compute the prototypes?

Calculating the centroids

- Centroid as average

$$\vec{c_j} = \frac{1}{|C_j|} \cdot \sum_{\vec{d_i} \in C_j} \vec{d_i}$$

- Centroid as sum

$$\vec{c_j} = \sum_{\vec{d_i} \in C_j} \vec{d_i}$$

- Centroid as normalized sum

$$\vec{c_j} = \frac{1}{\|\vec{c_j}\|} \cdot \sum_{\vec{d_i} \in C_j} \vec{d_i}$$

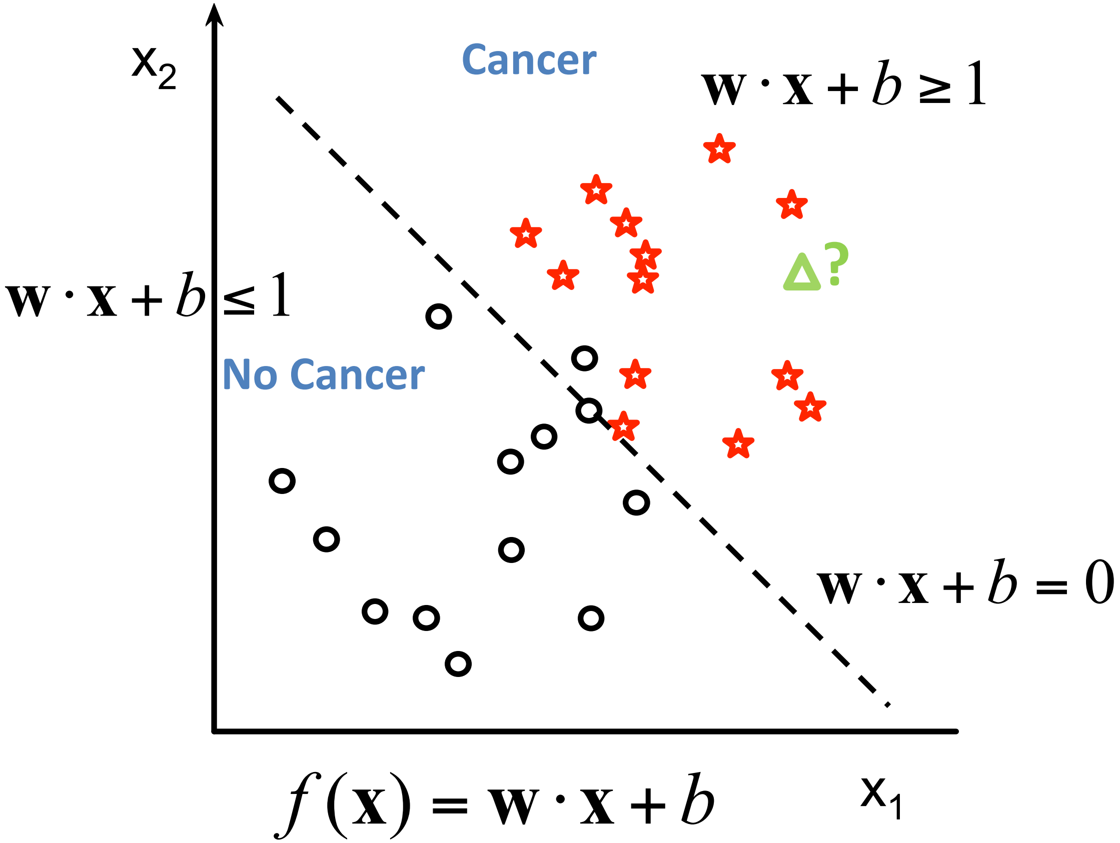
- Centroid computation using the Rocchio formula

$$\vec{c_j} = \beta \cdot \sum_{\vec{d_i} \in C_j} \vec{d_i} - \gamma \cdot \sum_{\vec{d_i} \notin C_j} \vec{d_i}$$

Comments on Centroid-Based Classification

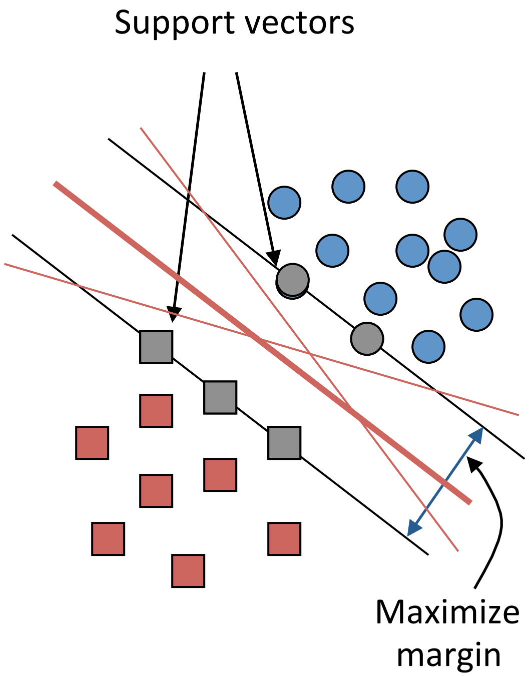
- Computationally simple and fast model
  - Short training and testing time
- Good results in text classification
- Amenable to changes in the training set
- Can handle imbalanced document sets
- Disadvantages:
  - Inadequate for non-linear classification problems
  - Problem of inductive bias or model misfit
    - Classifiers are tuned to the contingent characteristics of the training data rather than the constitutive characteristics of the categories

Linear models


Cancer
$\mathbf{w}\cdot\mathbf{x}+b \geq 1$
x_2
△?
$\mathbf{w}\cdot\mathbf{x}+b \leq 1$
No Cancer
$\mathbf{w}\cdot\mathbf{x}+b = 0$
$f(\mathbf{x}) = \mathbf{w}\cdot\mathbf{x}+b$
x_1

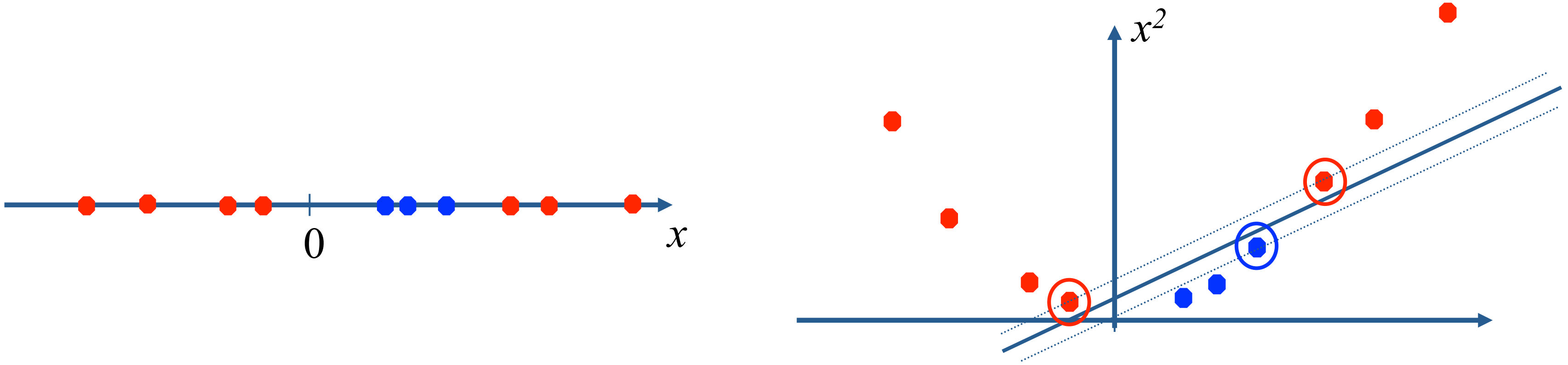
SVM

- A binary SVM classifier can be seen as a hyperplane in the feature space separating the points that represent the positive from negative instances.
  - SVMs selects the hyperplane that maximizes the margin around it.
  - Hyperplanes are fully determined by a small subset of the training instances, called the *support vectors.*



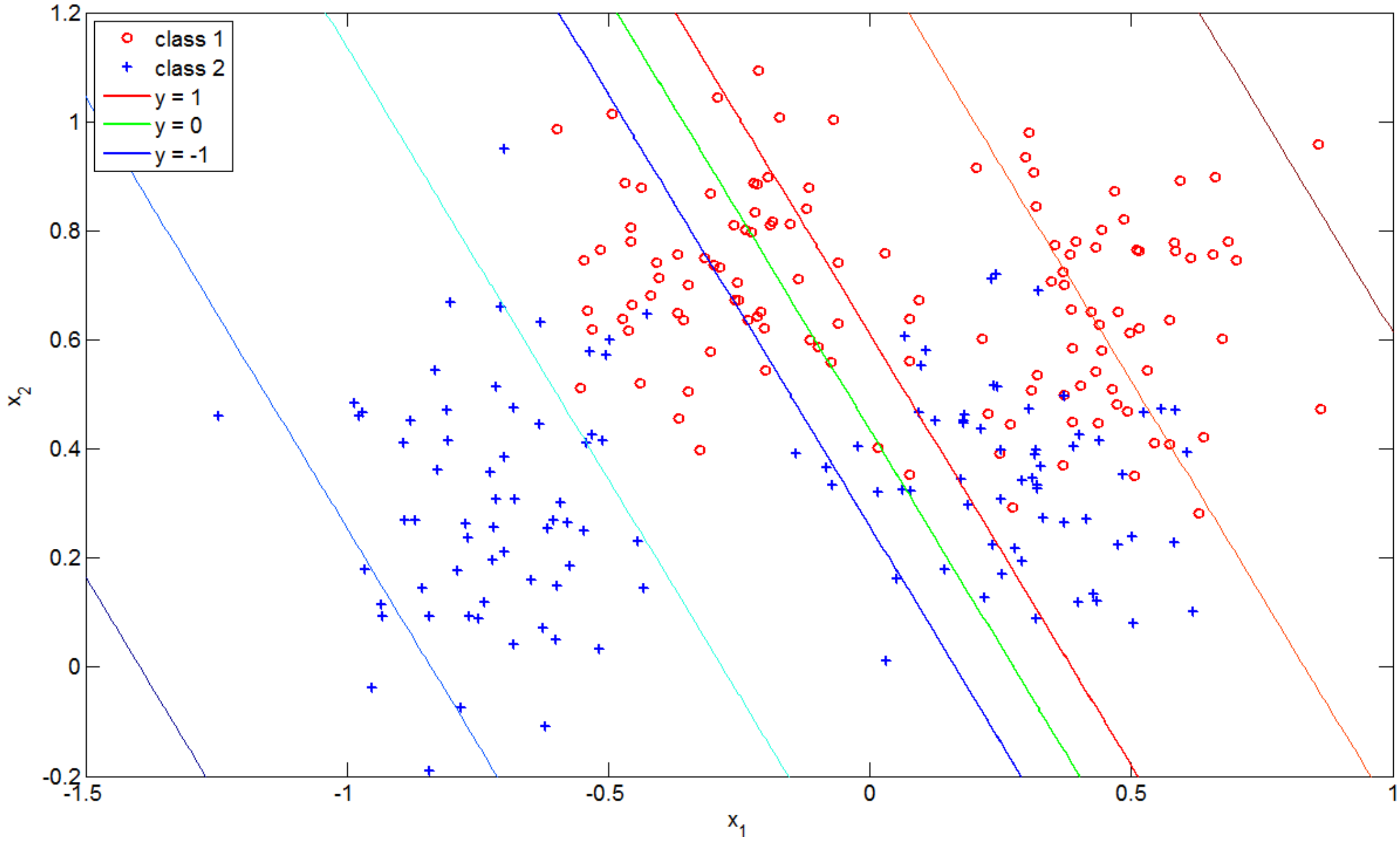
Non-linear SVM

- What about classes whose training instances are not linearly separable?
  - The original input space can always be *mapped* to some higher-dimensional feature space where the training set is separable.
    - A *kernel function* is some function that corresponds to an inner product in some expanded feature space.



Decision surface of SVMs

http://clopinet.com/CLOP

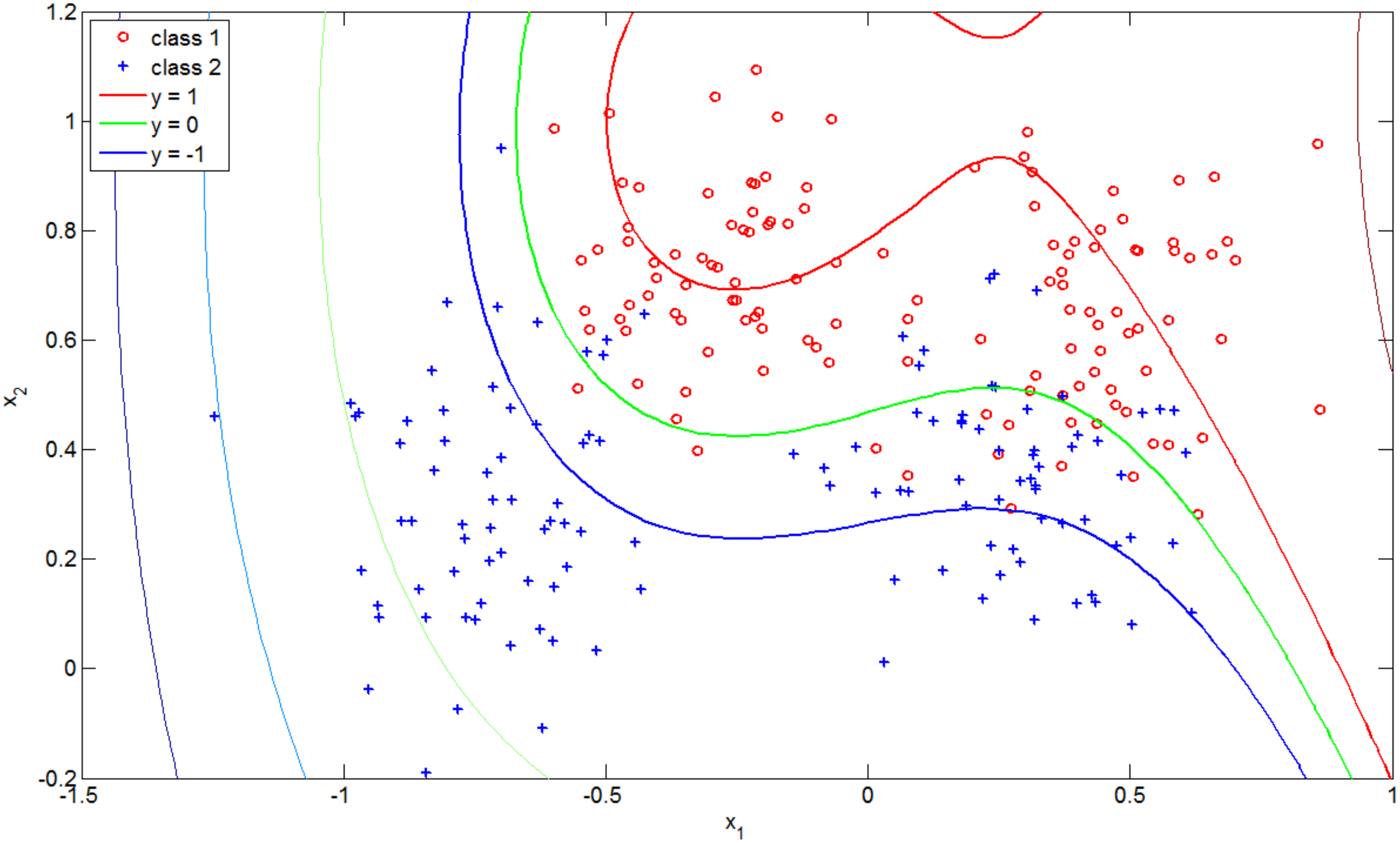


Linear support vector machine

Laboratorio de Tecnologías del Lenguaje
Ciencias Computacionales, INAOE

Decision surface of SVMs

http://clopinet.com/CLOP



Non-linear support vector machine

SVM – discussion

- The support vector machine (SVM) algorithm is very fast and effective for text classification problems.
  - Flexibility in choosing a similarity function
    - By means of a kernel function
  - Sparseness of solution when dealing with large data sets
    - Only support vectors are used to specify the separating hyperplane
  - Ability to handle large feature spaces
    - Complexity does not depend on the dimensionality of the feature space

Evaluation of text classification

- What to evaluate?
- How to carry out this evaluation?
  - Which elements (information) are required?
- How to know which is the best classifier for a given task?
  - Which things are important to perform a fair comparison?

Evaluation – general ideas

- Performance of classifiers is evaluated experimentally
- Requires a document set labeled with categories.
  - Divided into two parts: *training* and *test* sets
  - Usually, the test set is the smaller of the two
- A method to smooth out the variations in the corpus is the *n-fold cross-validation.*
  - The whole document collection is divided into *n* equal parts, and then the training-and-testing process is run *n* times, each time using a different part of the collection as the test set. Then the results for *n* folds are averaged.

Performance metrics

- Considering a binary problem

	Label YES	Label NO
Classifier YES	***a***	***b***
Classifier NO	***c***	***d***

$$\text{accuracy} = \frac{a+d}{a+b+c+d}$$

$$\text{recall (R)} = \frac{a}{a+c} \qquad \text{precision (P)} = \frac{a}{a+b} \longrightarrow F = \frac{2PR}{P+R}$$

- Recall for a category is defined as the percentage of correctly classified documents among all documents belonging to that category, and precision is the percentage of correctly classified documents among all documents that were assigned to the category by the classifier.

What happen if there are more than two classes?

Micro and macro averages

- *Macroaveraging*: Compute performance for each category, then average.
  - Gives equal weights to all categories
- *Microaveraging*: Compute totals of a, b, c and d for all categories, and then compute performance measures.
  - Gives equal weights to all documents

Is it important the selection of the averaging strategy?

What happen if we are very bad classifying the minority class?

Comparison of different classifiers

- Direct comparison
  - Compared by testing them on the same collection of documents and with the same background conditions.
  - This is the more reliable method
- Indirect comparison
  - Two classifiers may be compared when they have been tested on different collections and with possibly different background conditions if both were compared with a common baseline.

Project

- Authorship attribution consists in assign a text of *unknown authorship* to one candidate author, given a *set of candidate authors* for whom text samples are available.
- You will use a corpus texts from 5 Mexican Poets.
- The project consists in building TWO classifiers:
  - one based on the BoW representation
  - other based on character n-grams
- For both classifiers use: TFIDF, SVM, 10CFV; report macro F1 results.
- The purpose is two determine what is more important for discriminating among these authors: content or style.
  - Please include an analysis of the most relevant features.

Project (2)

- At the end of the course you need to send us a report (2 pages) on the experiments.
  - Introduction about the purpose of the experiment
  - Description of used representations
  - Description of experimental setup (dataset, classifier configuration, evaluation measures)
  - Description of results, their analysis and discussion.
- Details on the submission process will be given next class.