Special topics on text mining

[Part I: text classification]

Hugo Jair Escalante, Aurelio Lopez,

Manuel Montes and Luis Villaseñor

Laboratorio de
Tecnologías del Lenguaje
Ciencias Computacionales, INAOE

Representation and preprocessing of documents

Hugo Jair Escalante, Aurelio Lopez,
Manuel Montes and Luis Villaseñor

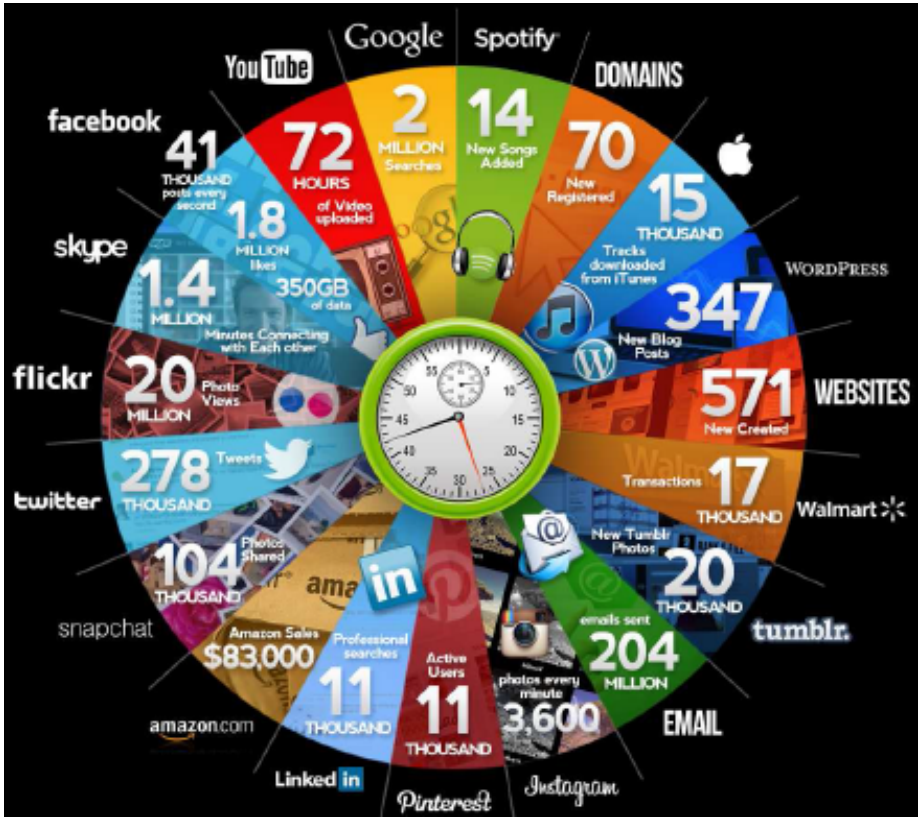
Laboratorio de
Tecnologías del Lenguaje
Ciencias Computacionales, INAOE

Agenda

- Recap: text classification
- Representation of documents
- Preprocessing
- Feature selection
- Discussion
- Assignments

Textual information

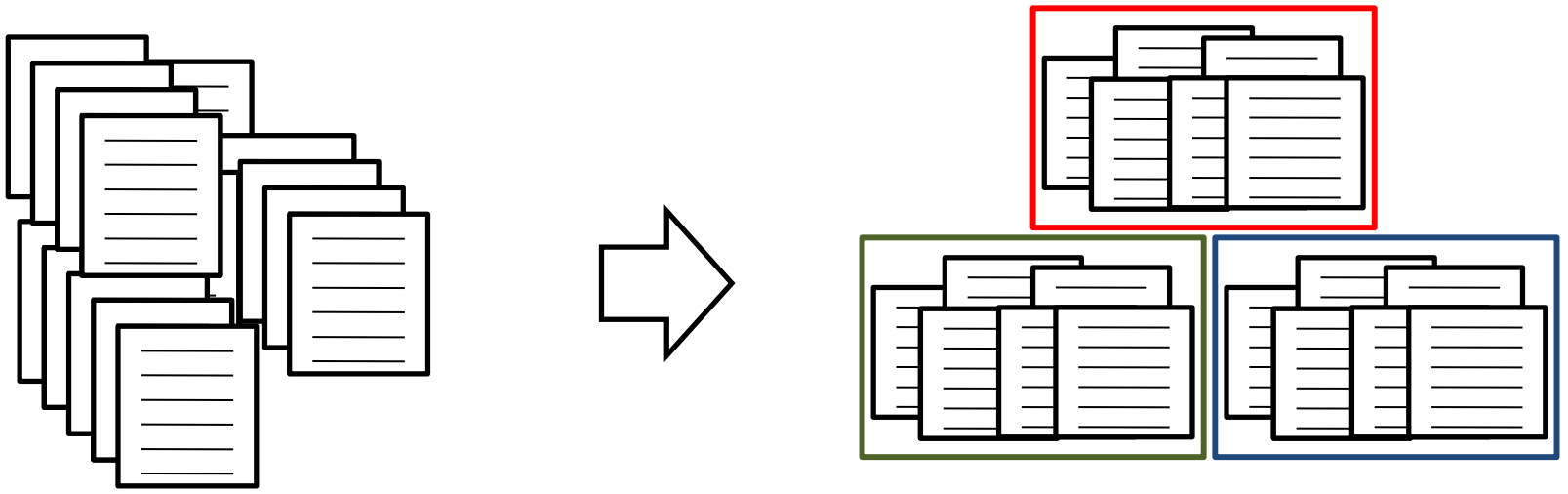
- Considerable growth of textual information in electronic format
- Inhability of human beings for processing such amounts of information in reasonable times
- Text mining: “The process of deriving high-quality information from text”



http://www.qmee.com

Text classification

- **Text classification** is the assignment of free-text documents to one or more predefined categories based on their content
- Categories depend on the object of interest



Documents (e.g., news articles)

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

Manual approach?

Text classification

- TC is probably the most studied topic within human language technology:
  - It can be considered a solved problem for certain scenarios: e.g., news classification and spam filtering**
- Nevertheless, many variants of TC are open problems; likewise, several tasks nowadays can be seen as TC and are far from being solved:
  - Cross-domain, multilingual, …
  - Author profiling, authorhip atribution, opinion mining, sarcasm/irony detection….

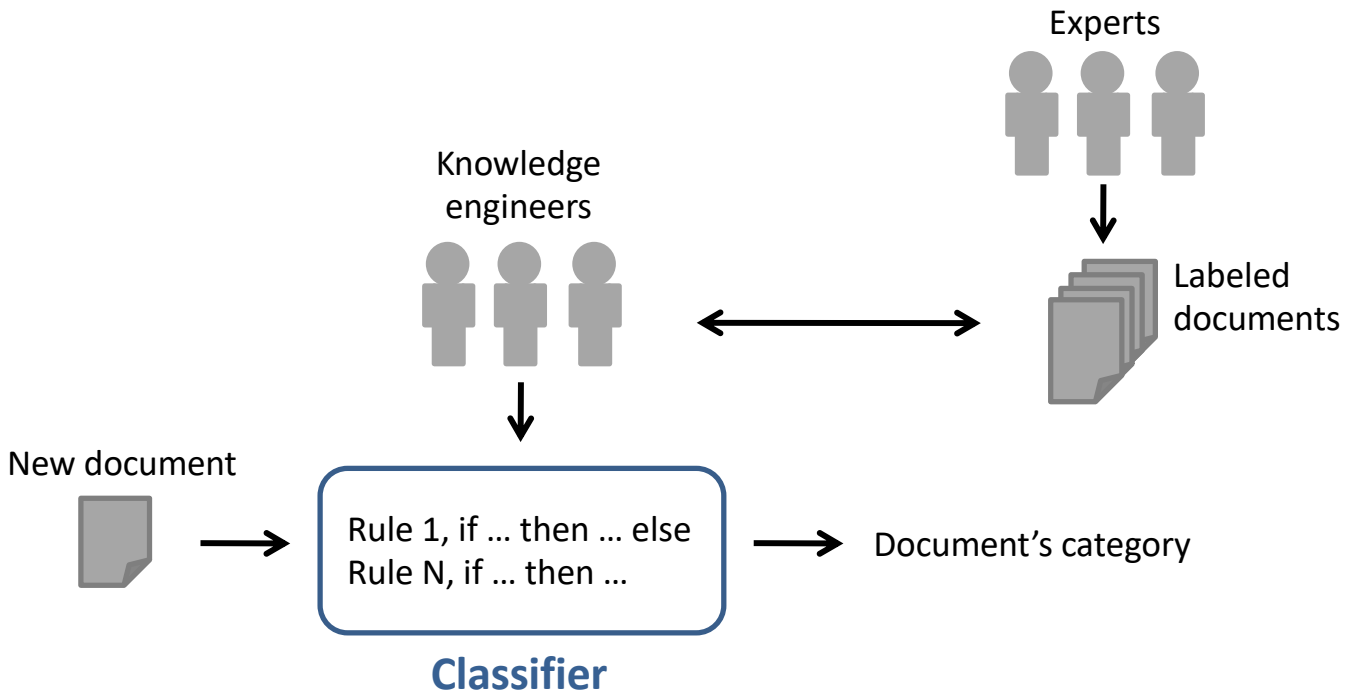
Manual classification

- **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 a classification function?

Hand-coded rule based systems



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

Example: filtering spam email

- **Rule-based classifier**

TABLE 1.1. *Average percentage of words or characters in an email message equal to the indicated word or character. We have chosen the words and characters showing the largest difference between* spam *and* email.

	george	you	your	hp	free	hpl	!	our	re	edu	remove
spam	0.00	2.26	1.38	0.02	0.52	0.01	0.51	0.51	0.13	0.01	0.28
email	1.27	1.27	0.44	0.90	0.07	0.43	0.11	0.18	0.42	0.29	0.01

Classifier 1

$$\text{if } (0.2 \cdot \%\texttt{you} - 0.3 \cdot \%\texttt{george}) > 0 \quad \text{then spam else email.}$$

Classifier 2

$$\text{if } (\%\texttt{george} < 0.6) \;\&\; (\%\texttt{you} > 1.5) \quad \text{then spam else email.}$$

Taken from Hastie et al. The Elements of Statistical Learning, 2007, Springer.

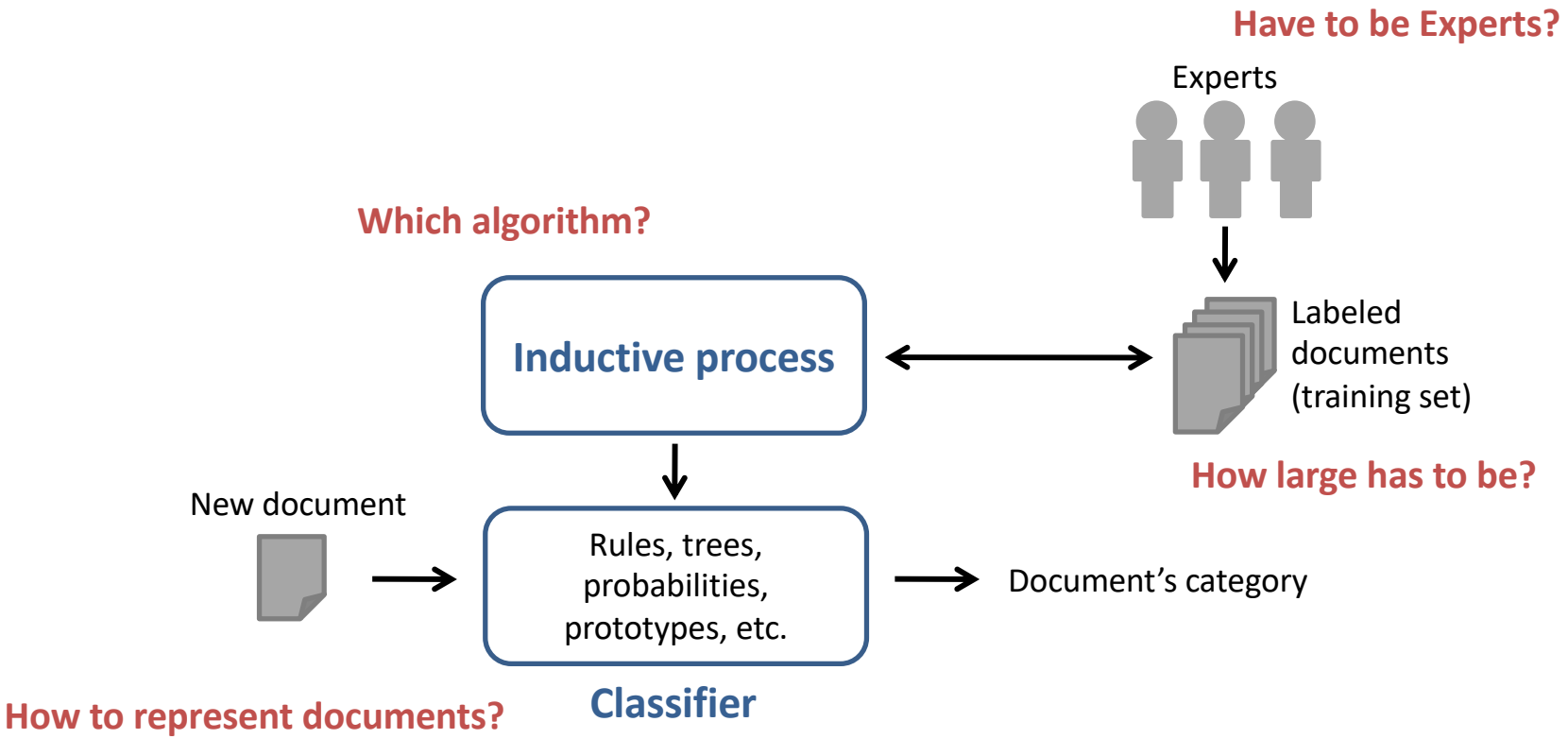
Machine learning approach (1)

- 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 $F : D \times C \rightarrow \{0, 1\}$, 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 $M : D \times C \rightarrow \{0, 1\}$ 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.

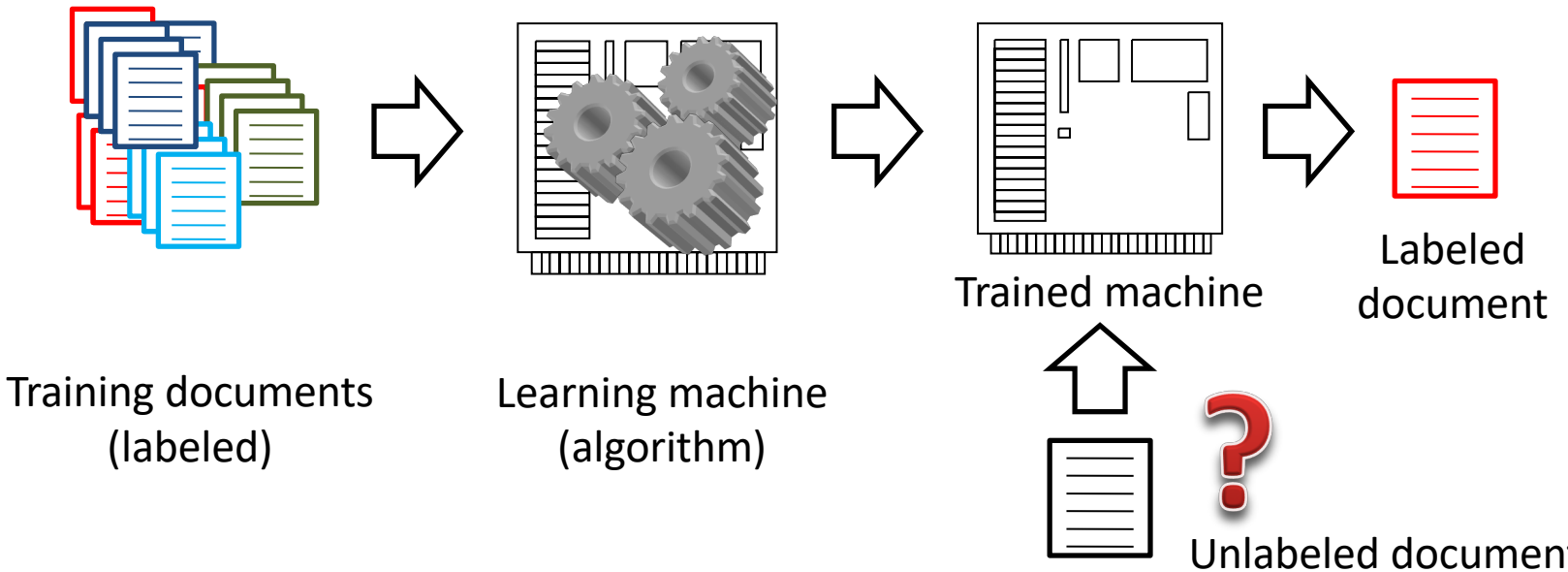
Ronen Feldman and James Sanger, The Text Mining Handbook

Machine learning approach (2)

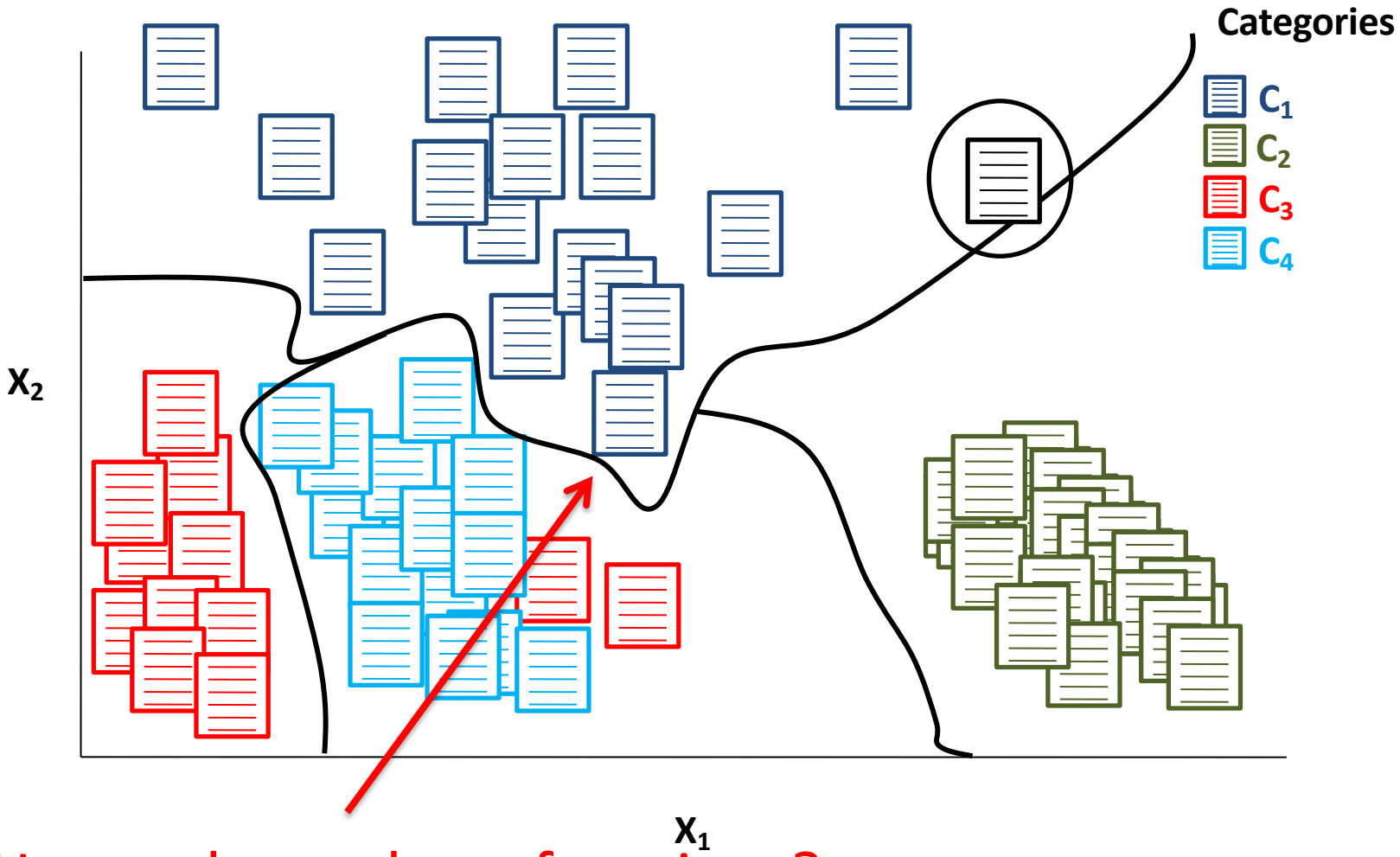


Machine learning: classification

- To learn a model able to make predictions regarding a variable of interest, using a set of other variables. Example: *text categorization*

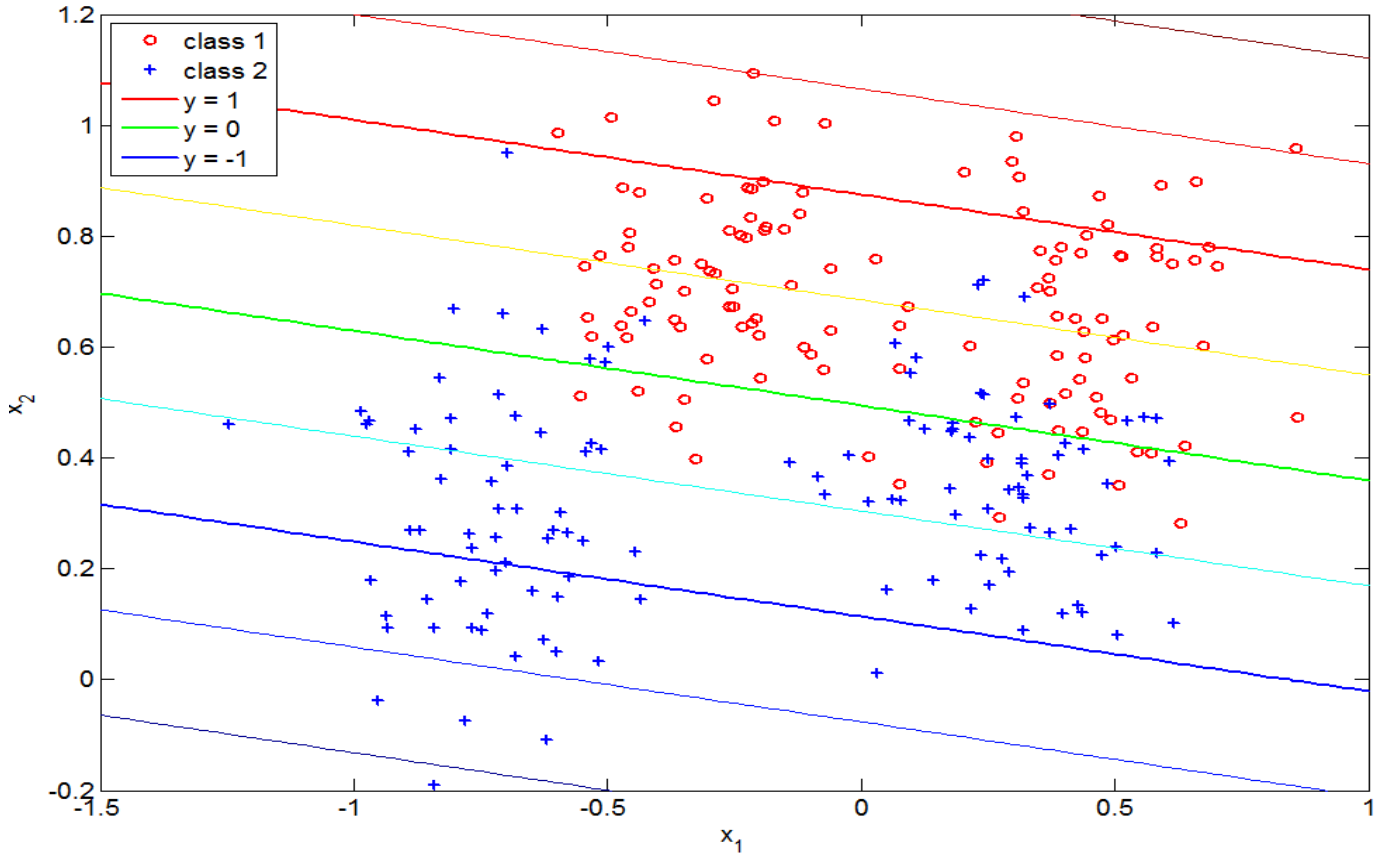


Machine learning: classification



How to learn these functions?

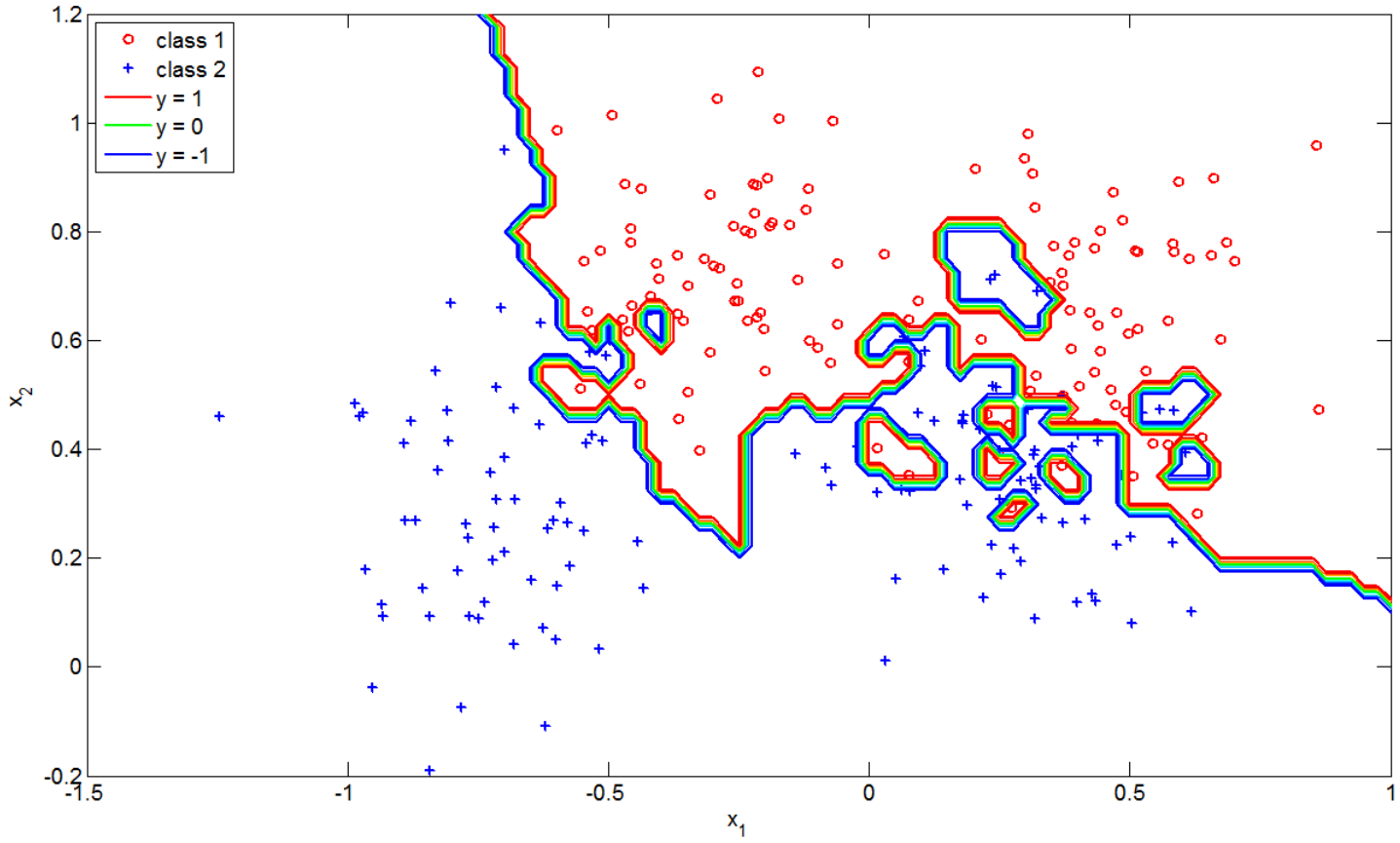
Machine learning: classification



Linear model

Machine learning: classification

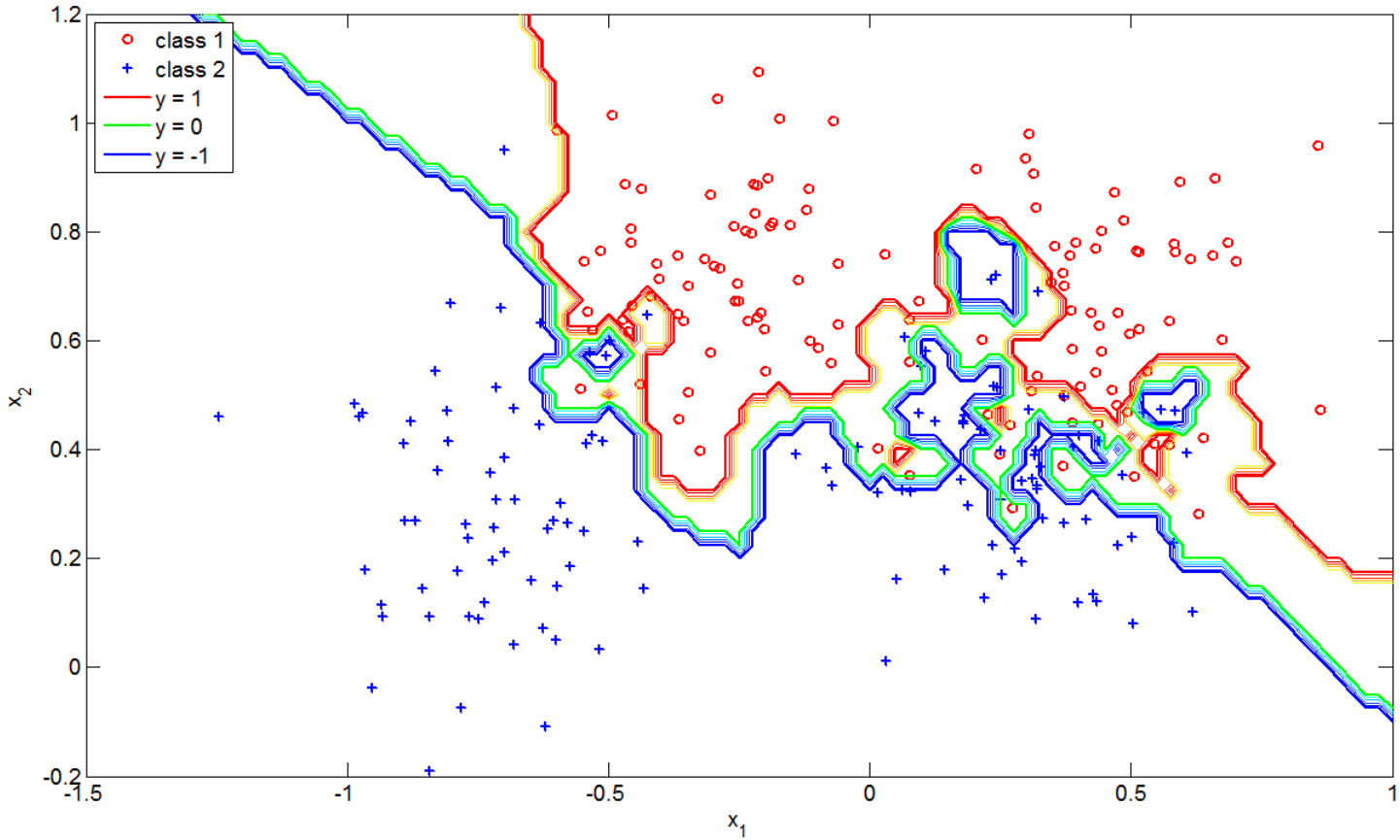
http://clopinet.com/CLOP



K=1

Machine learning: classification

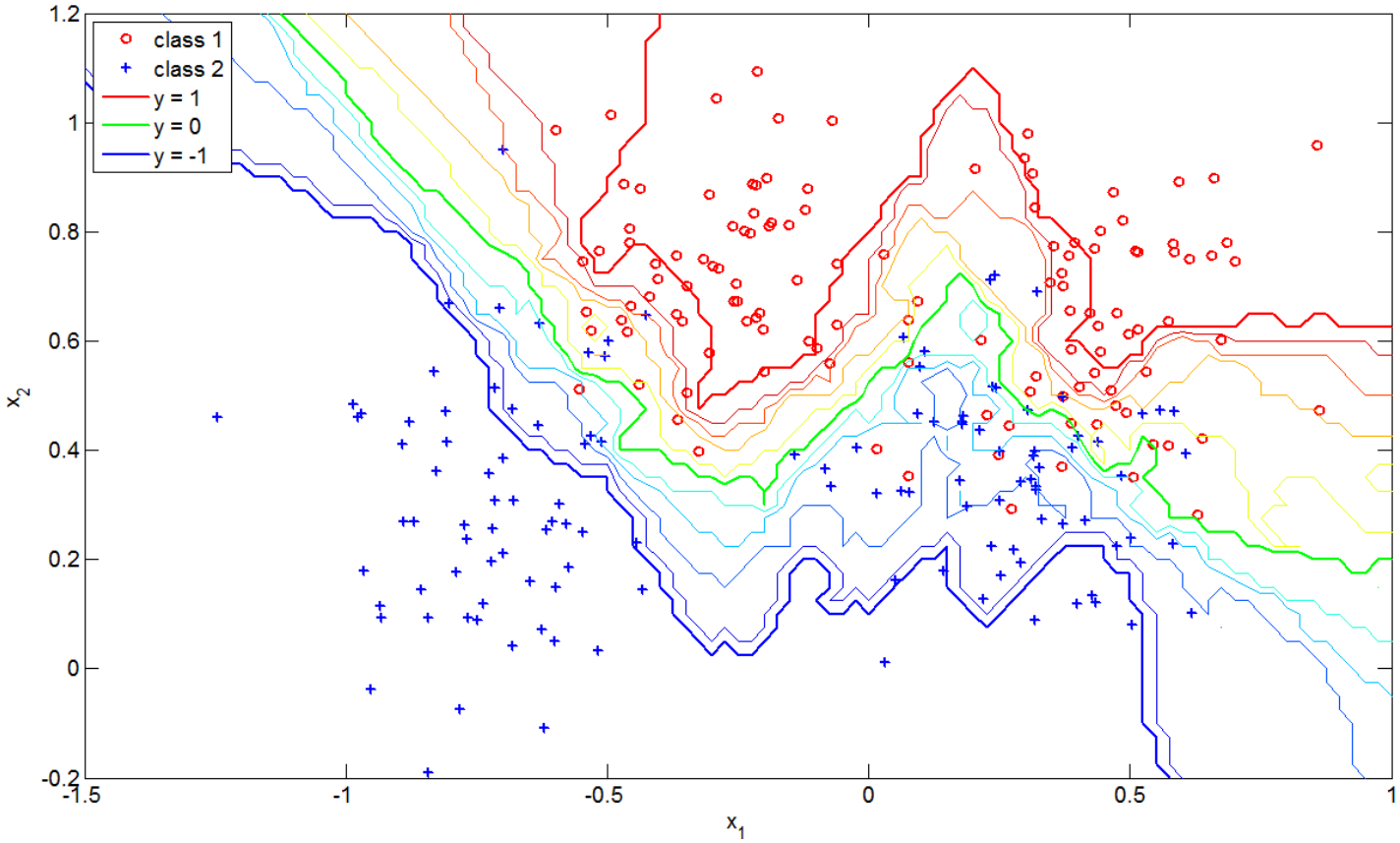
http://clopinet.com/CLOP



K=2

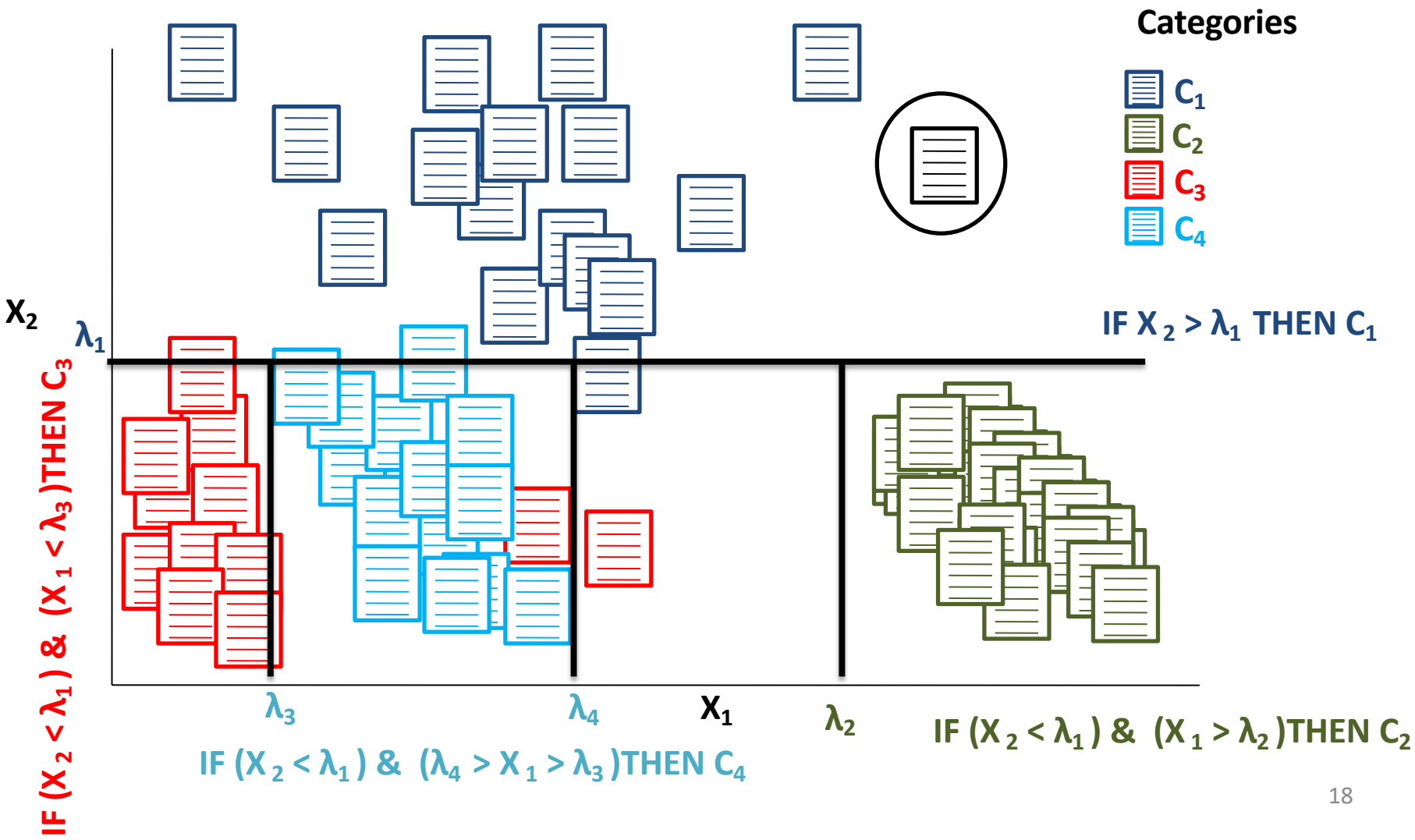
Machine learning: classification

http://clopinet.com/CLOP

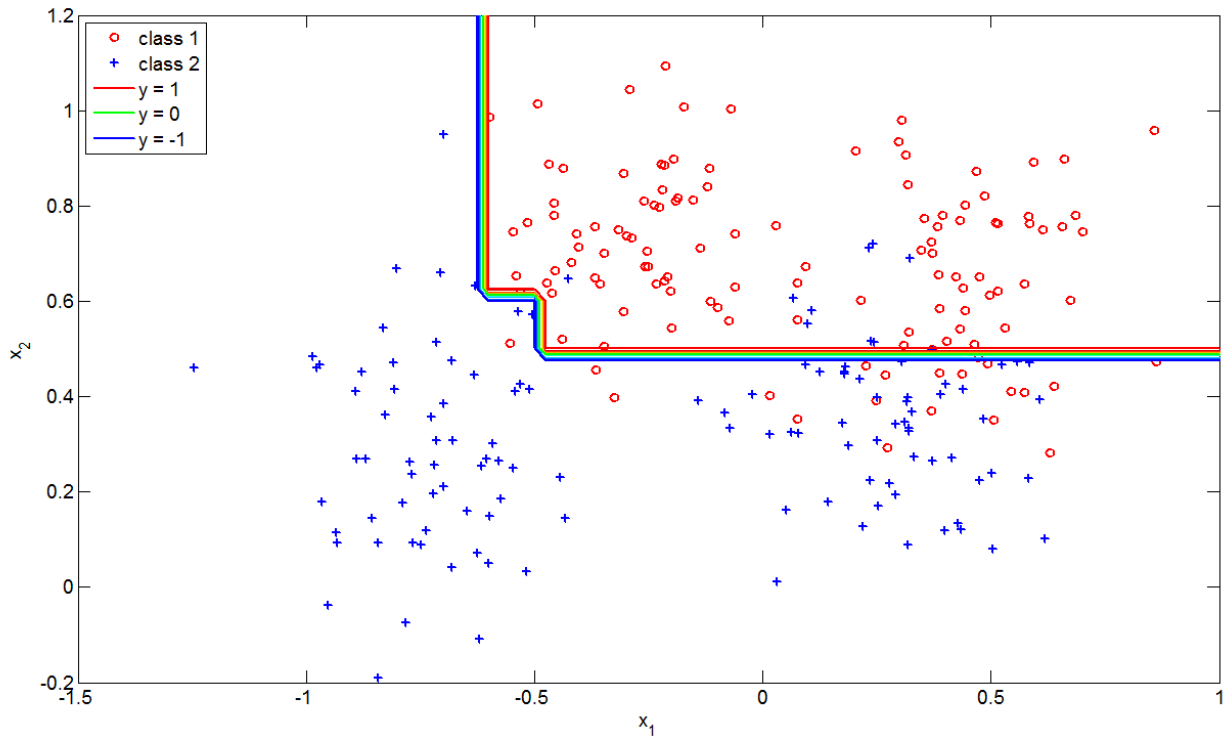


K=10

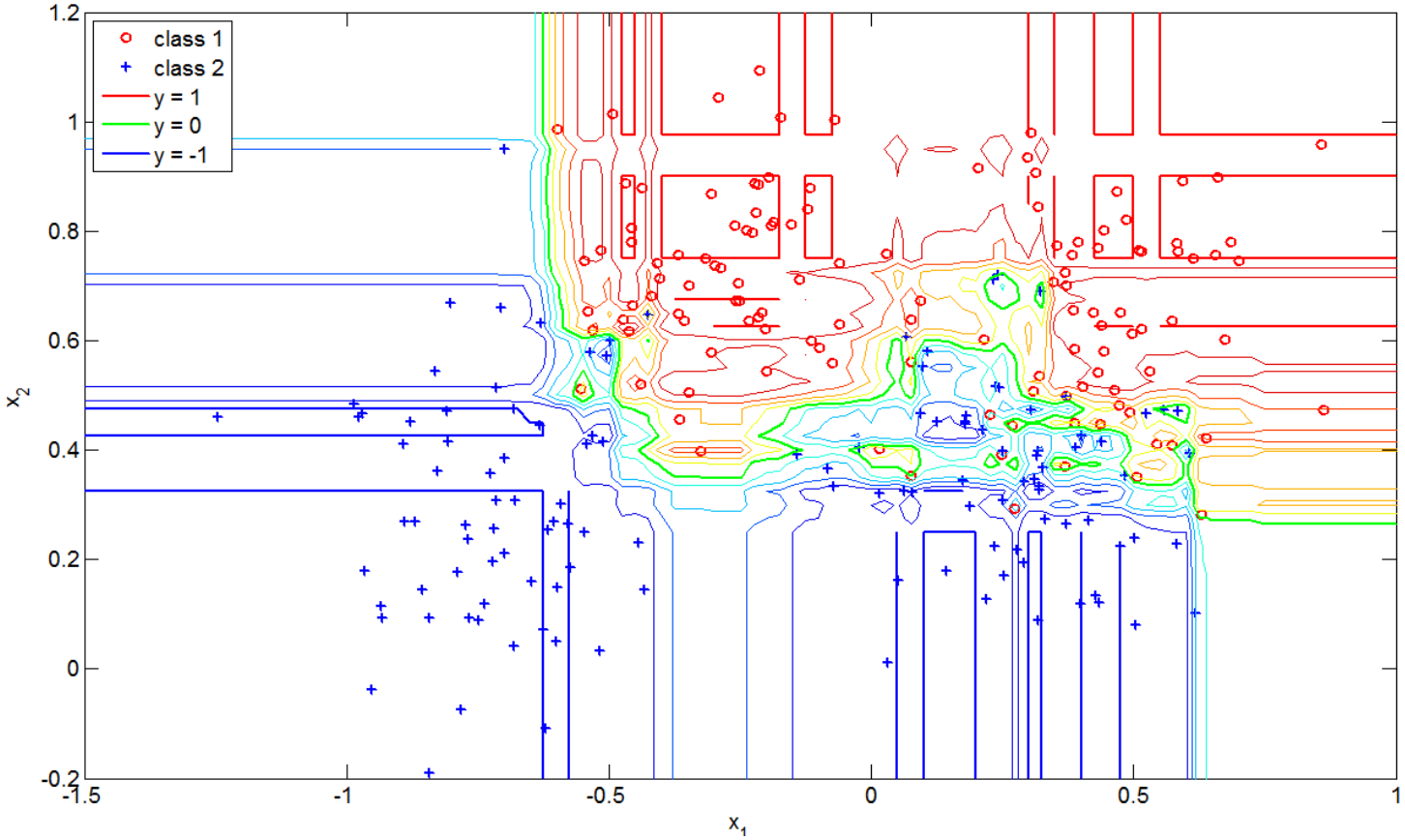
Machine learning: classification



Machine learning: classification DT

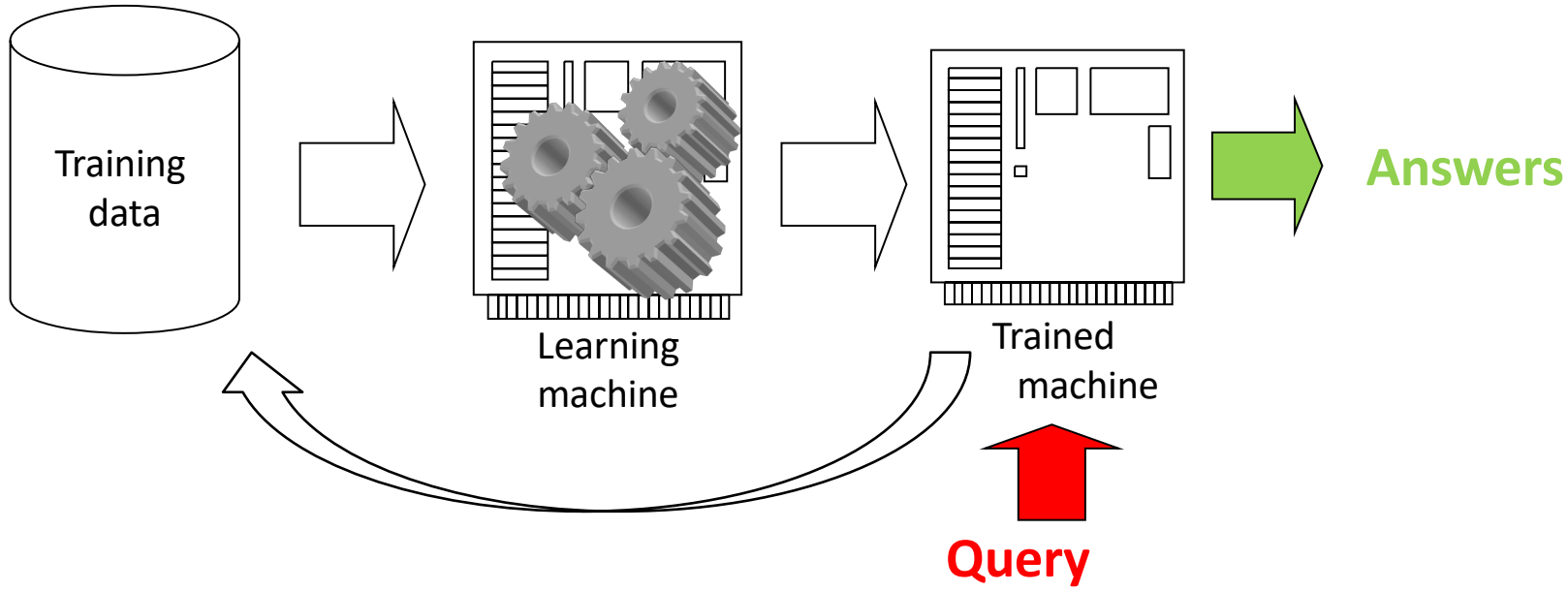


Machine learning: classification RF



Machine learning

Learning = representation + evaluation + optimization



Isabelle Guyon. **A Practical Guide to Model Selection.** In Jeremie Marie, editor, Machine Learning Summer School 2008, Springer Texts in Statistics, 2011. (slide from I.Guyon's)

Text classification

- **Machine learning approach to TC:**

Recipe

1. ~~Gather labeled documents~~
2. Construction of a classifier
   A. Document representation
   B. Preprocessing
   C. Dimensionality reduction
   D. Classification methods
3. Evaluation of a TC method

Assumption: a large enough training set of labeled documents is available

Later we will study methods that allow us to relax such assumption [semi-supervised and unsupervised learning]

Before representing documents: 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

Document representation

- Represent the content of *digital* documents in a way that they can be processed by a computer

Document representation

- Transform documents, which typically are strings of characters, into a representation suitable for the learning algorithm:
  - **Codify/represent/transform documents into a vector representation (vector space model)**

- The most common used document representation is the *bag of words (***BOW***)* approach
  - Documents are represented by the set of words that they contain
  - Word order is not captured by this representation
  - There is no attempt for understanding their content
  - The vocabulary of all of the different words in all of the documents is considered as the base for the vector representation

Document representation

Terms in the vocabulary
(*Basic units expressing document's content*)

	t_1	t_j	…	$t_{\|V\|}$
d_1				
d_2				
:		$w_{i,j}$		
d_m				

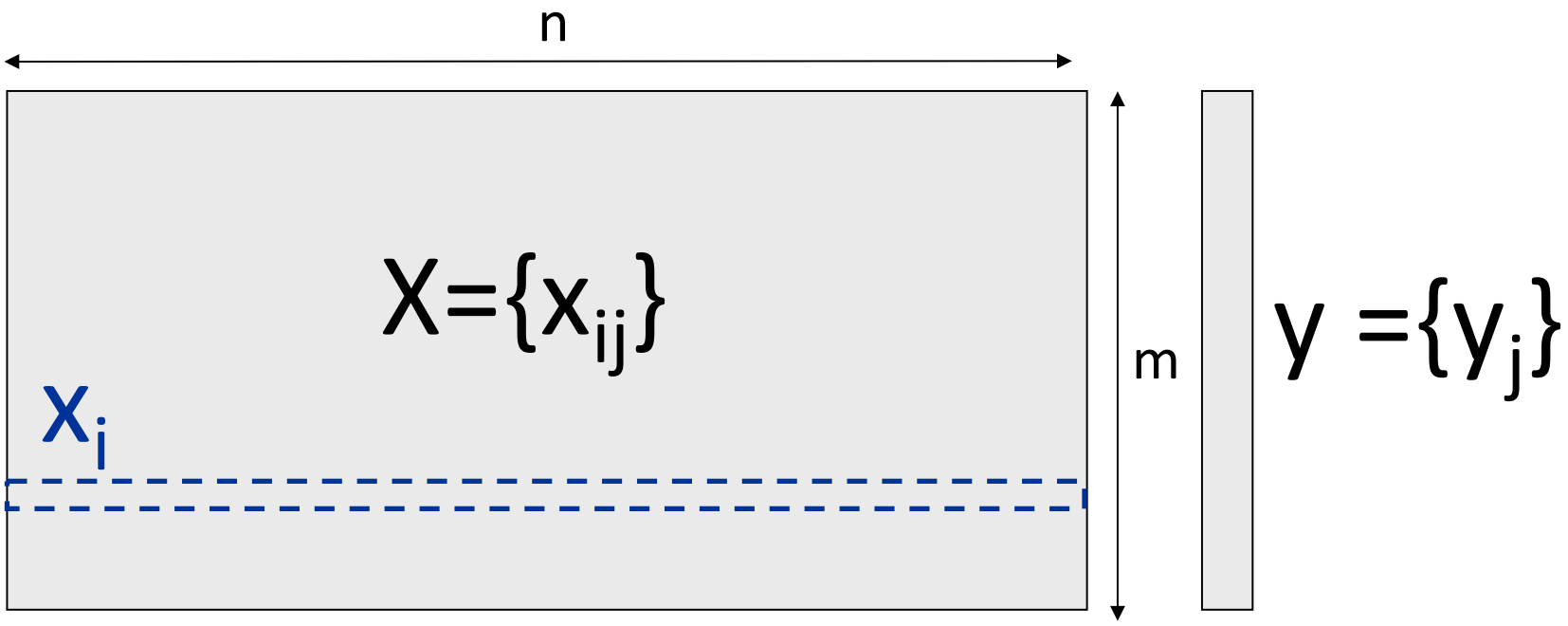
Documents in the corpus
(*one vector/row per document*)

V: Vocabulary from the collection *(i.e., et of all different words that occur in the corpus)*

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?

(ML Conventions)



Slide taken from I. Guyon. **Feature and Model Selection.** Machine Learning Summer School, Ile de Re, France, 2008.

Document representation

- **Simplest BOW-based representation:** Each document is represented by a binary vector whose entries indicate the presence/absence of terms from the vocabulary (**Boolean/binary weighting**)

Document	Content
Syllabus.txt	Advanced topics on text mining
Evaluation.txt	Homework, reports (text)
Students.txt	Graduate (Advanced)
Description.txt	Studying topics on text mining

Obtain the BOW representation with Boolean weighting for these documents

Term weighting
[extending the Boolean BOW]

- **Two main ideas:**

Local

  - The importance of a term increases proportionally to the number of times it appears in the document.
    - It helps to *describe* document's content.

Global

  - 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

Does the order of words matters?

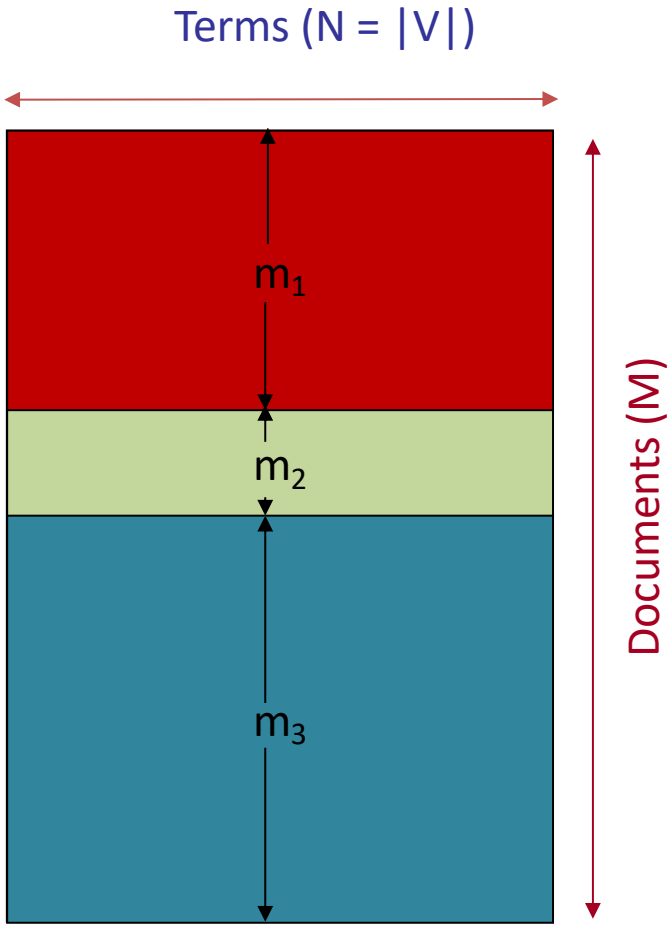
Term weighting – main approaches

- **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:**
  - $w_{i,j} = tf(t_j, d_i) \times idf(t_j)$, where:
    - $tf(t_j, d_i)$ indicates the ocurrences of t_j in document d_i
    - $idf(t_j) = \log [N/df(t_j)]$, where $df(t_j)$ is the number of documets that contain the term t_j.

These methods do not use the information of the classes, why?

(A brief note on evaluation in TC)

- The available data is divided into three subsets:
  - **Training (m1)**
    - used for the construction (learning) the classifier
  - **Validation (m2)**
    - Optimization of parameters of the TC method
  - **Test (m3)**
    - Used for the evaluation of the classifier



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)

$$w_{i,j} = \frac{w_{i,j}}{\sqrt{\sum_{k=1}^{M} (w_{i,k})^2}}$$

- **tf x idf weighting scheme:**
  - $w_{i,j}$ = tf(t_j, d_i) × idf(t_j), where:
    - tf(t_j, d_i) indicates the ocurrences of t_j in document d_i
    - idf(t_j) = log [N/df(t_j)], where df(t_j) is the number of documets that contain the term t_j.

Normalization?

Term weighting alternatives?

Term weighting alternatives

- Unsupervised weighting schemes: Traditional schemes, proposed for information retrieval, e.g., *tf, tf-idf*, *Booleano*, etc.

- Supervised schemes: Discriminative information is incorporated, designed for text classification problems, e.g., *tf-ig, tf-chi*^{2}, etc.

Acronym	Name	Formula	Description	Ref.
B	Boolean	$x_{i,j} = \mathbf{1}_{\{\#(t_i,d_j)>0\}}$	Indicates the prescense/abscense of terms	[2]
TF	Term-Frequency	$x_{i,j} = \#(t_i, d_j)$	Accounts for the frequency of occurrence of terms	[2]
TF-IDF	TF - Inverse Document Frequency	$x_{i,j} = \#(t_i, d_j) \times \log(\frac{N}{df(t_j)})$	An TF scheme that penalizes the frequency of terms across the collection	[2]
TF-IG	TF - Information Gain	$x_{i,j} = \#(t_i, d_j) \times IG(t_j)$	TF scheme that weights term occurrence by its information gain across the corpus.	[11]
TF-CHI	TF - Chi-square	$x_{i,j} = \#(t_i, d_j) \times CHI(t_j)$	TF scheme that weights term occurrence by its χ^2 statistic	[11]
TF-RF	TF - Relevance Frequency	$x_{i,j} = \#(t_i, d_j) \times \log(2 + \frac{TP}{\max(1,TN)})$	TF scheme that weights term occurrence by its χ^2 statistic	[10]

Homework?

Tarea 1

Enero 31, 2019,

Entrega: Febrero 5, 2019

1. Buscar y descargar al menos corpora asociados a alguna tarea de interés en minería de textos (clasificación)
    1. Las tareas deben ser diferentes entre todos los estudiantes (traslape máximo de 1)
    2. Al menos una de las colecciones debe estar asociada a más de 2 clases
2. Implementar la representación de BoW con pesados Booleano, TF, TFIDF
3. Describir el problema asociado de clasificación
4. Estimar lo siguiente
    1. Número de ceros en la matriz resultante (Booleano)
    2. Tamaño de vocabulario
    3. Longitud de palabra más larga/corta
    4. Número de palabras con frecuencia 1
    5. Número de clases
    6. Espacio en memoria que ocupa la matriz

Entrega: Febrero 5, 2019

1. Generar una gráfica con la frecuencia de las palabras
2. Gráficar los valores idf de todas las palabras del vocabulario

Scope of BoW?
(only text)

Bag of visual words

- Idea: to represent images as histograms that account for the frequency by which prototypical visual descriptors (visual words) occur

Learning a visual vocabulary

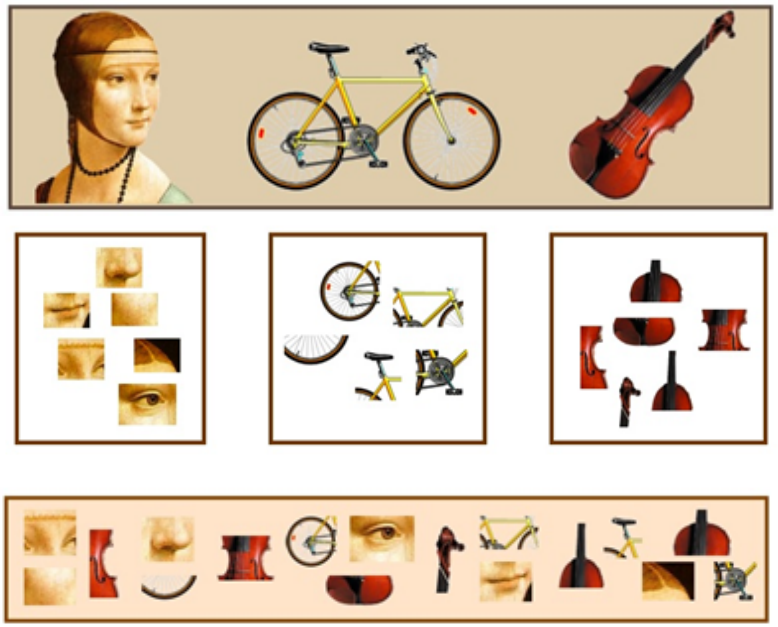
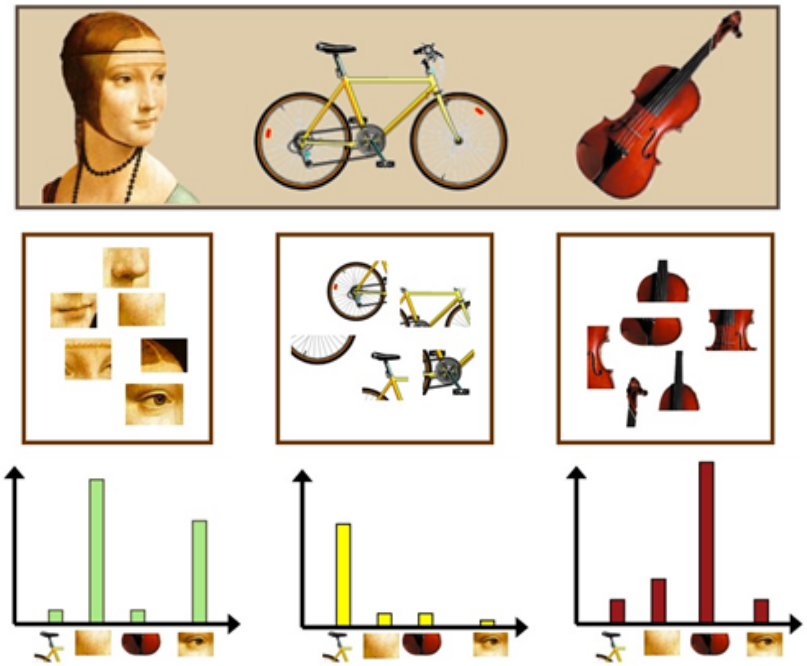
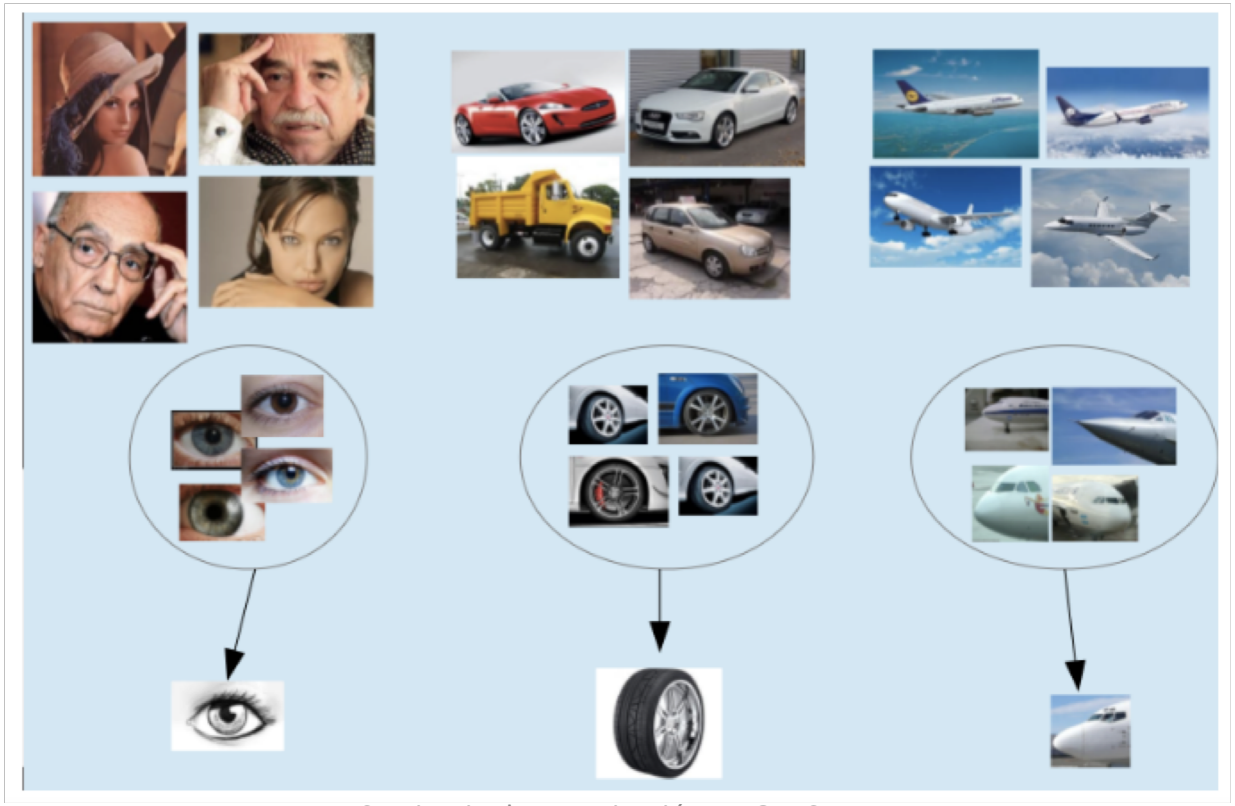


Image representation



Bag of Visual Words

- Define/learn a codebook (playing the role of the vocabulary in text)



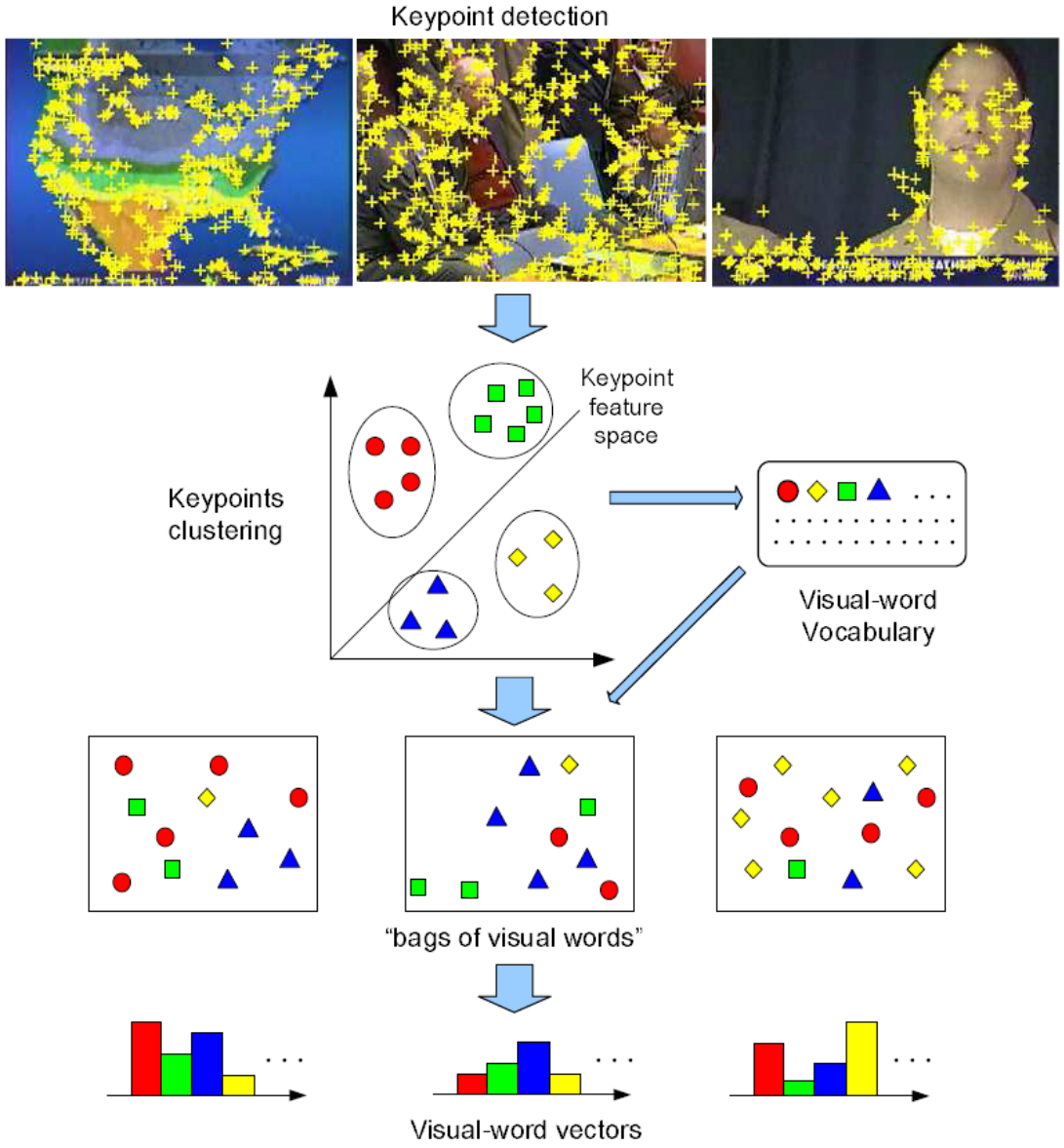
Bag of Visual Words

- Represent objects in a similar fashion as in BoW:

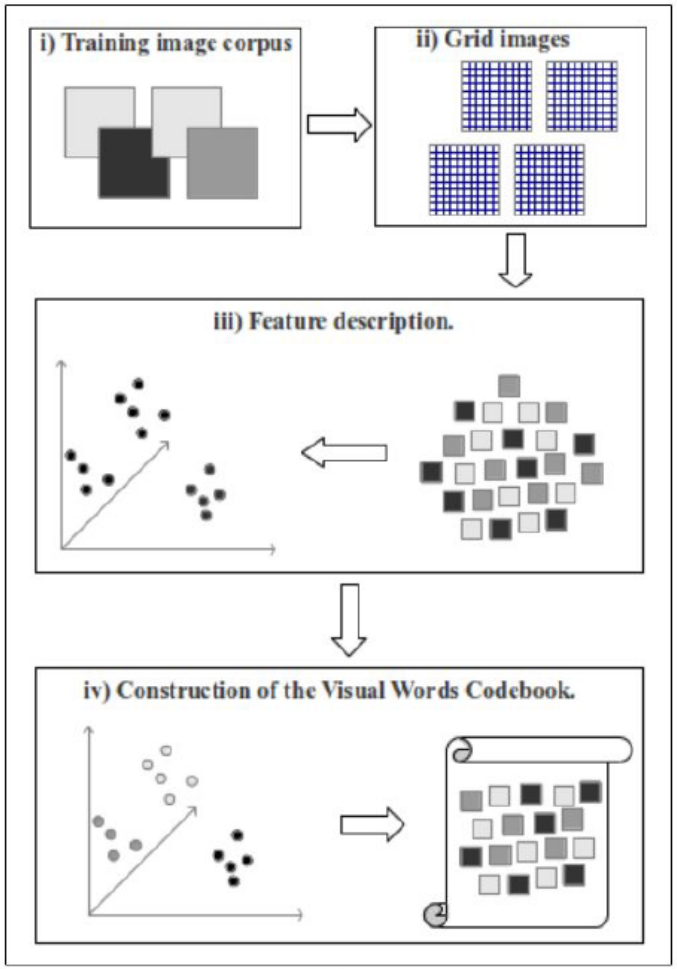
	t_1	t_j	…	$t_{\|V\|}$
d_1				
d_2				
:		$w_{i,j}$		
d_m				

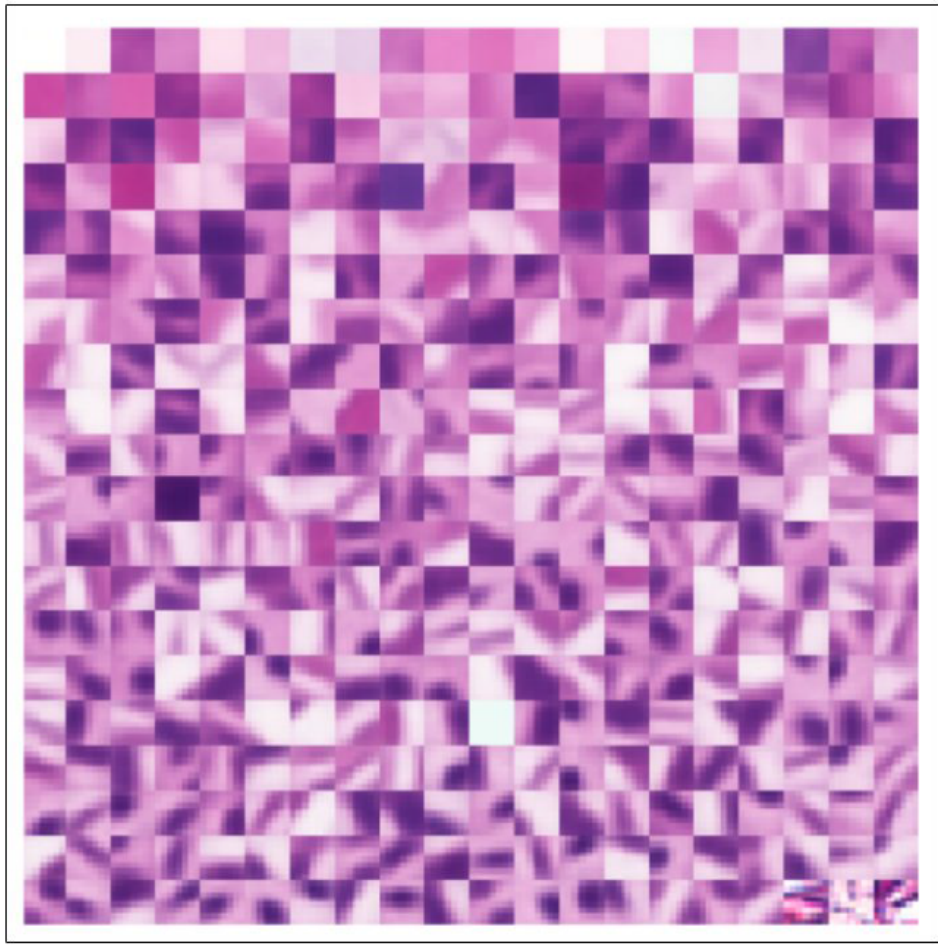
Bag of Visual Words

(Bag of visual words)

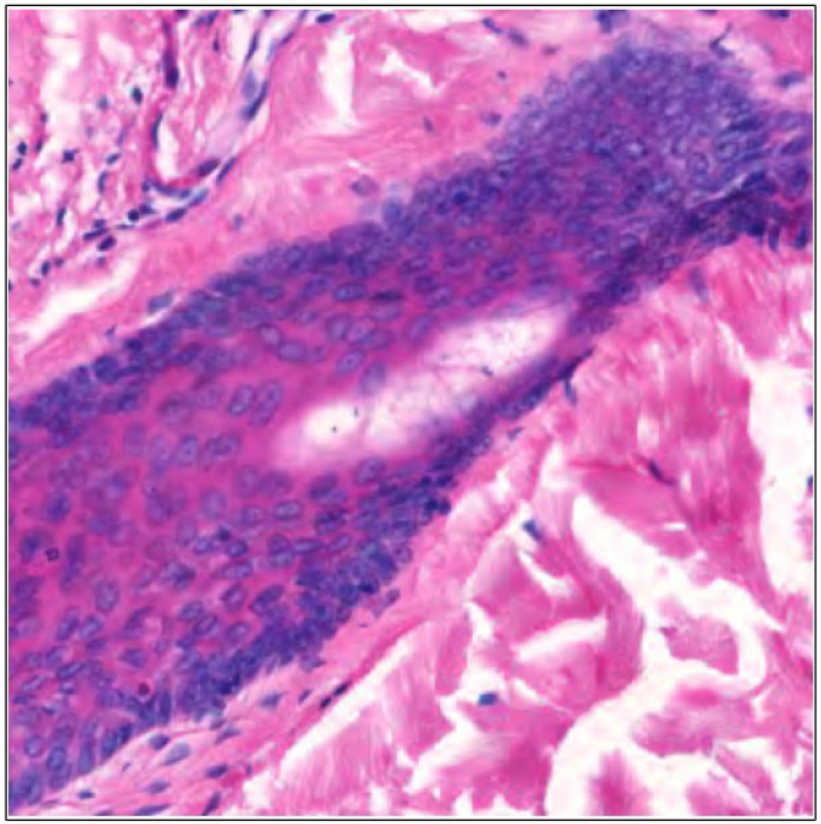


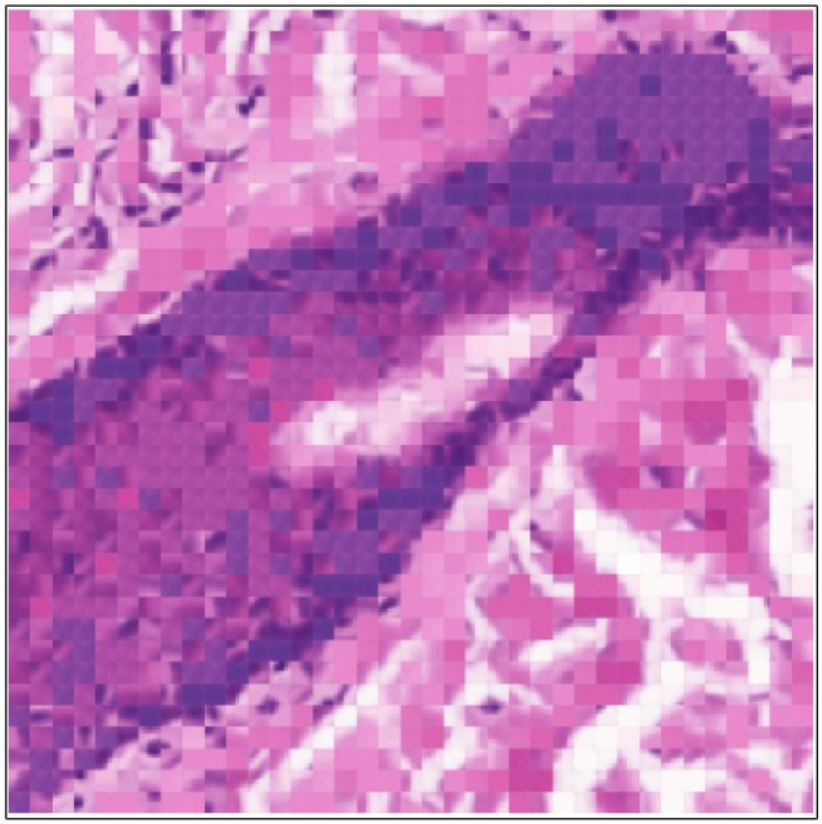
Bag of visual words





Bag of visual words





Bag of Visual Words

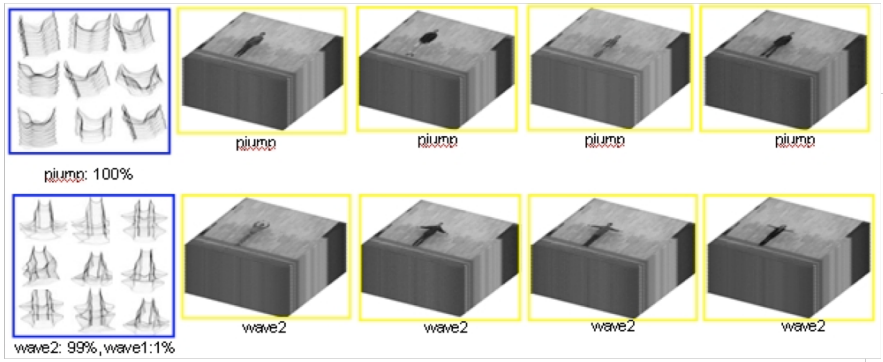
- Hence: we can use the stack of methods developed in NLP to deal / overcome / alleviate limitations / challenges in computer vision
  - BoVW is one of the most widely used representation in computer vision and related fields
  - It has reported outstanding performance in a wide variety of tasks

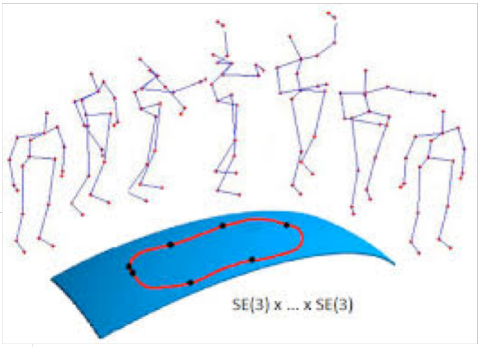
Scope of BoW?

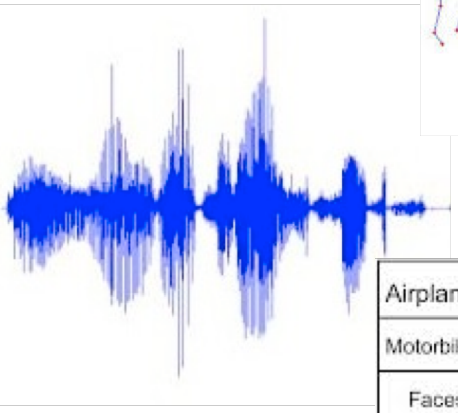
(only text and images)?

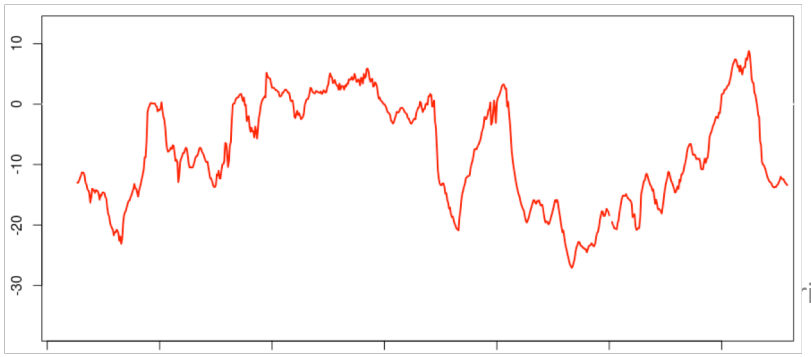
(Restricted to images?)

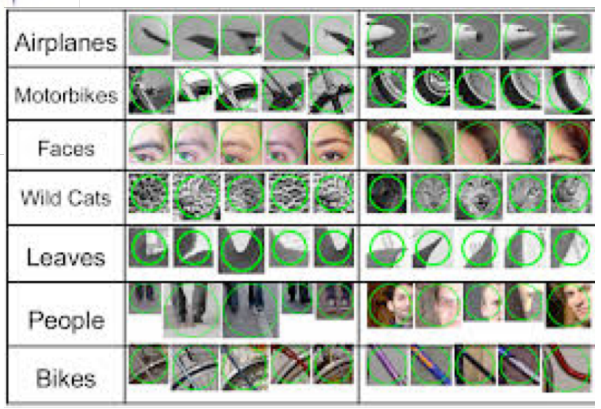
- No, we can process under the same scheme any kind of data that can be represented by codewords



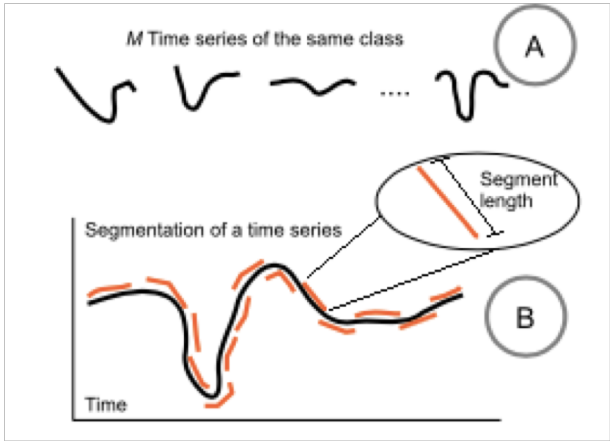


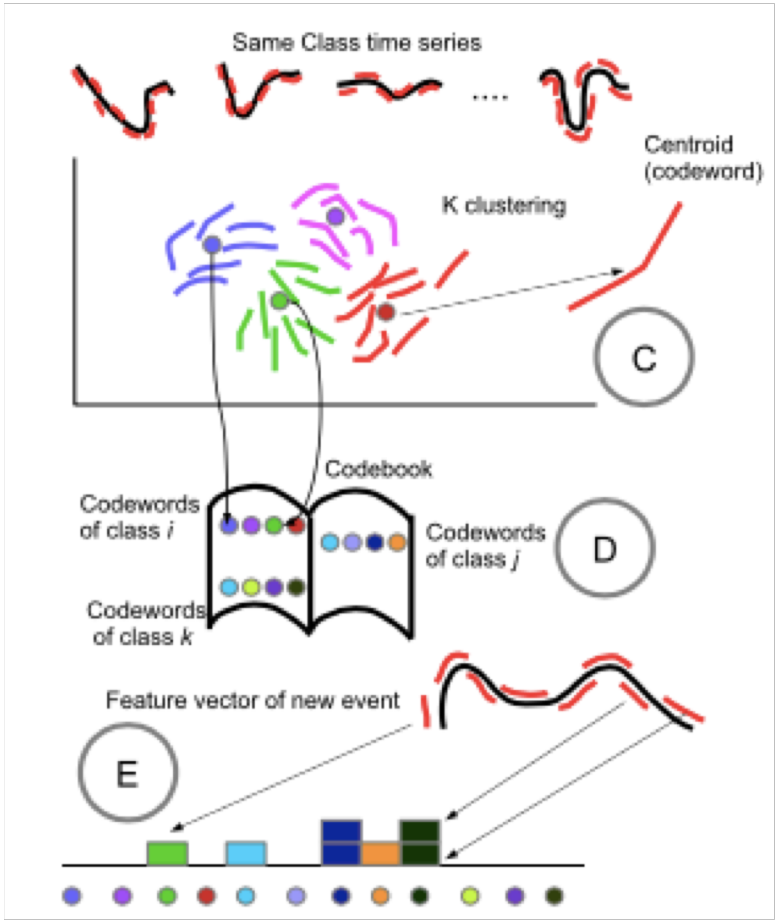






Same approach for modeling time series





L.C. Gonzalez Gurrola, R. Moreno, H. J. Escalante. **Learning Roadway Surface Disruption patterns using the Bag of Words representation**, IEEE Transactions on Intelligent Transportation Systems, doi: 10.1109/TITS.2017.2662483, 2017

Text classification

- **Machine learning approach to TC:**

Recipe

1. ~~Gather labeled documents~~
2. Construction of a classifier
   - ~~A. Document representation~~
   - ~~B. Preprocessing~~
   - C. Dimensionality reduction
   - D. Classification methods
3. Evaluation of a TC method

Dimensionality issues

- What do you think is the average size of the vocabulary in a small-scale text categorization problem (~1,000 - 10,000 documents)

- **It depends on the domain and type of the corpus, although usual vocabulary sizes in text classification range from a few thousands to millions of terms**

Dimensionality issues

- A central problem in text classification is the high dimensionality of the feature space.
  - There is 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 relevancy of the features?

The curse of dimensionality

- Dimensionality is a common issue in machine learning (in general)
- Number of positions scale exponentially with the dimensionality of the problem
- We need an exponential number of training examples to cover all positions

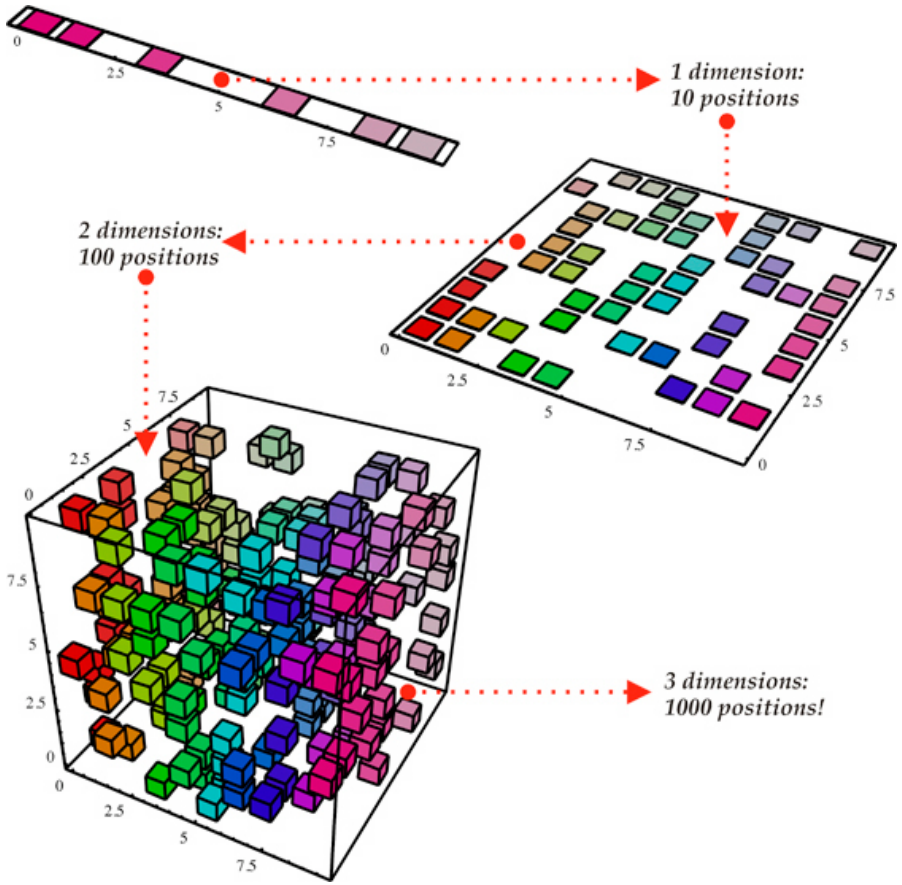


Image taken from: Samy Bengio and Yoshua Bengio, **Taking on the Curse of Dimensionality in Joint Distributions Using Neural Networks,** in: *IEEE Transaction on Neural Networks*, special issue on data mining and knowledge discovery, volume 11, number 3, pages 550-557, 2000.

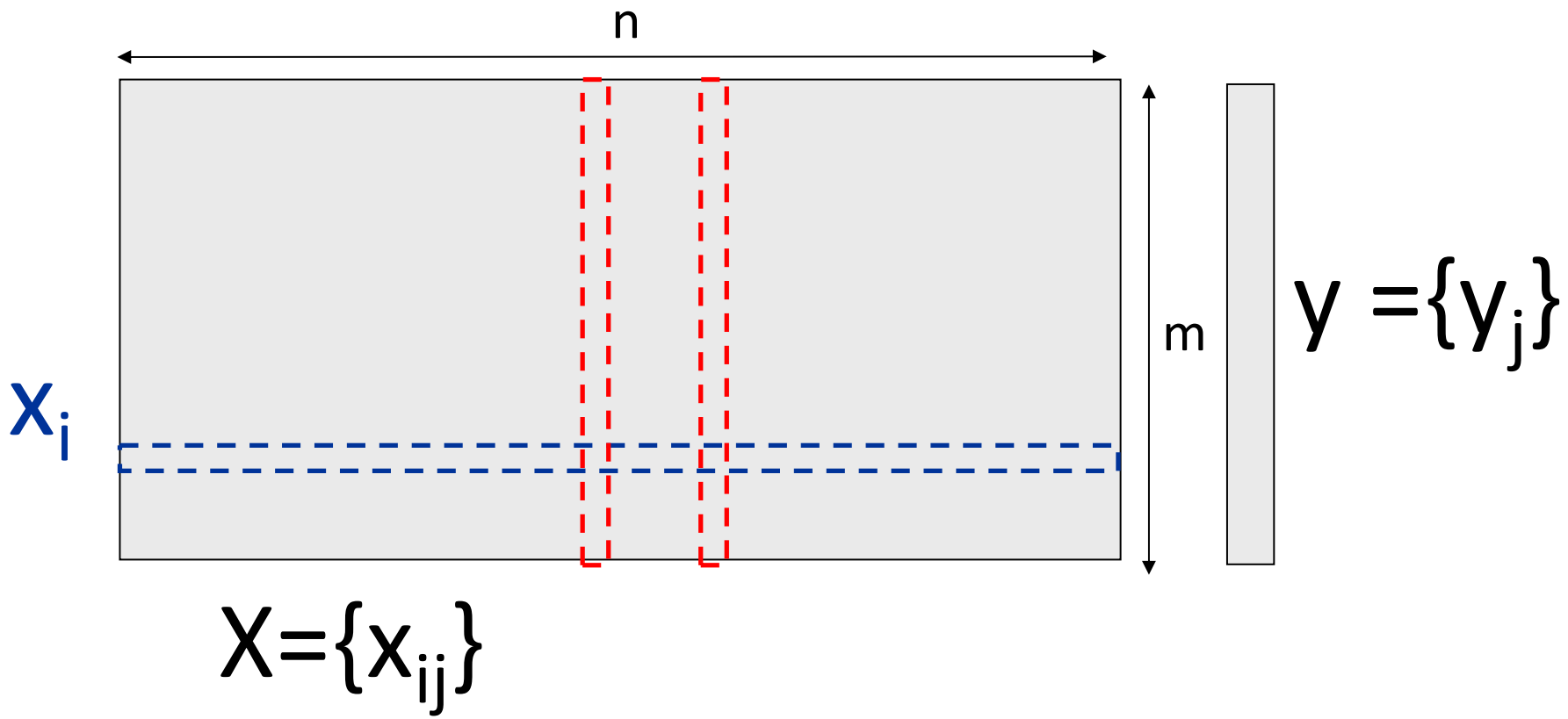
Dimensionality reduction: Two main approaches

- **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)

Feature selection in general

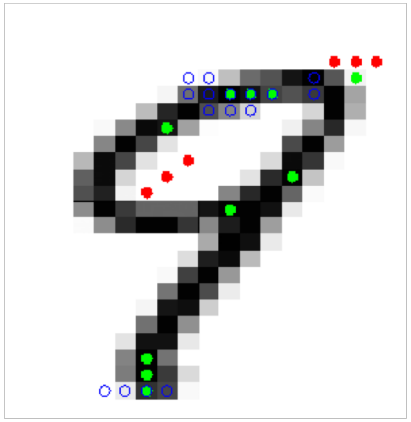
Select columns from this matrix

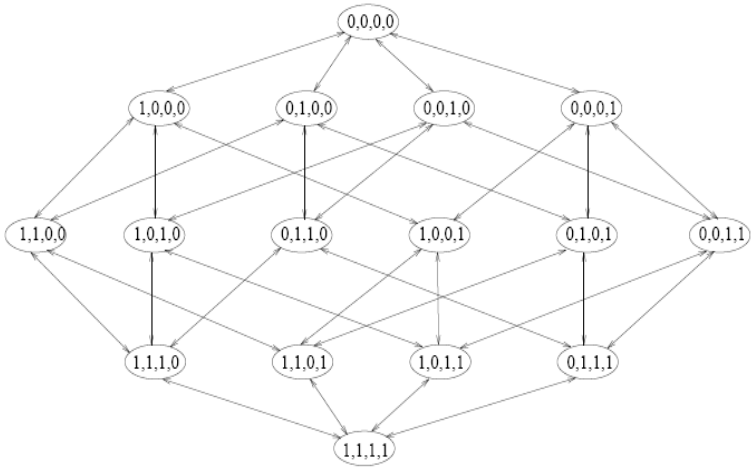


Slide taken from I. Guyon. **Feature and Model Selection.** Machine Learning Summer School, Ile de Re, France, 2008.

Feature selection: the ML perspective

- **Problem:** to find the subset of features that are more helpful for classification
  - Reduce the dimensionality of the data
  - Eliminate uninformative features
  - Find discriminate features

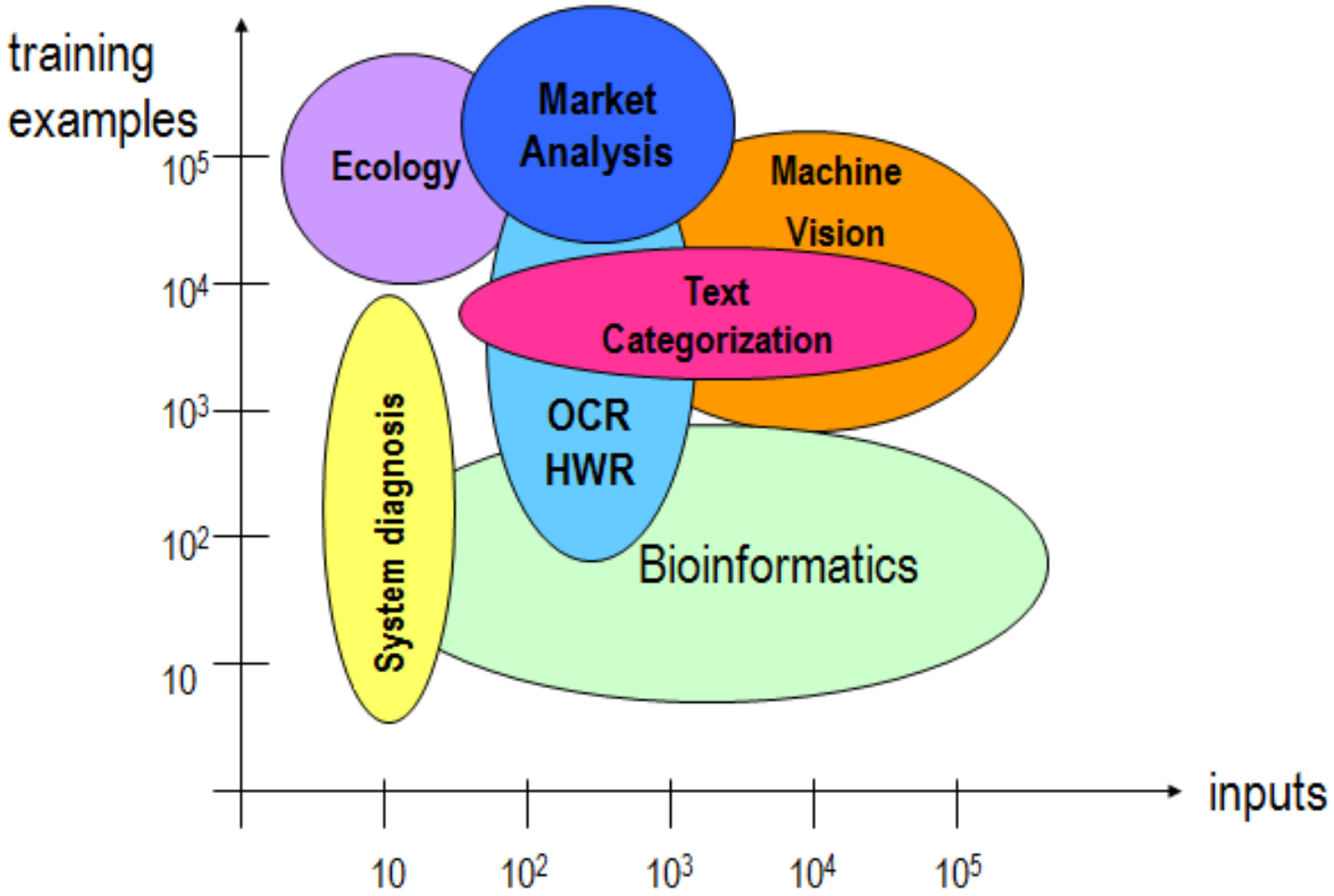




For a problem with n features there are 2^n different subsets of features

I. Guyon, et al**. Feature Extraction: Foundations and Applications,** Springer 2006.

Feature selection: the ML perspective



I. Guyon, et al**. Feature Extraction: Foundations and Applications,** Springer 2006.

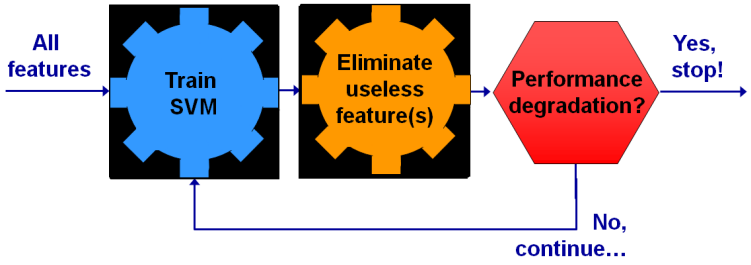
Feature selection: the ML perspective

- **Filters:** Evaluate the importance of features using methods that are independent of the classification model

$$\mathbf{w}_f^i = \frac{\mu_i^+ - \mu_i^-}{\sigma_i^+ + \sigma_i^-}$$

- **Wrappers:** Evaluate the importance of subsets of features using the classification model (a search strategy is adopted)

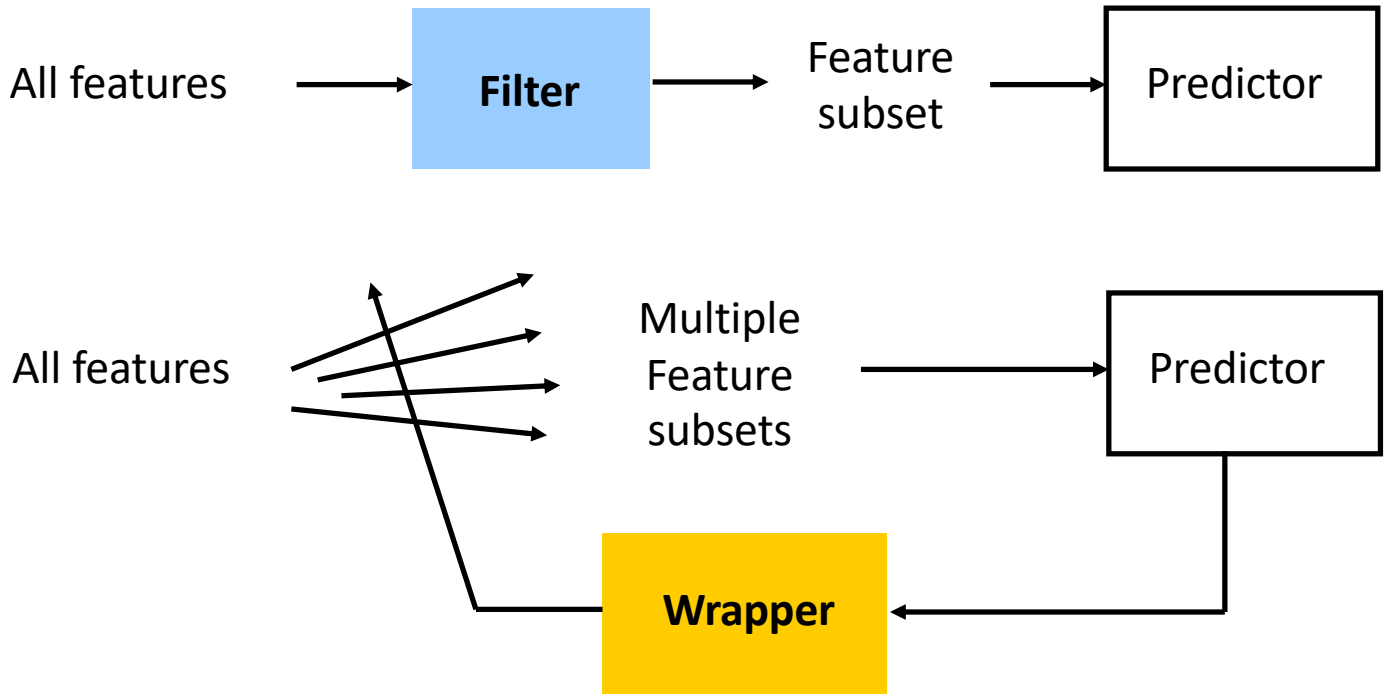
- **Embedded:** Take advantage of the nature of the classification model being considered



I. Guyon, et al**. Feature Extraction: Foundations and Applications,** Springer 2006.

Filters vs. Wrappers

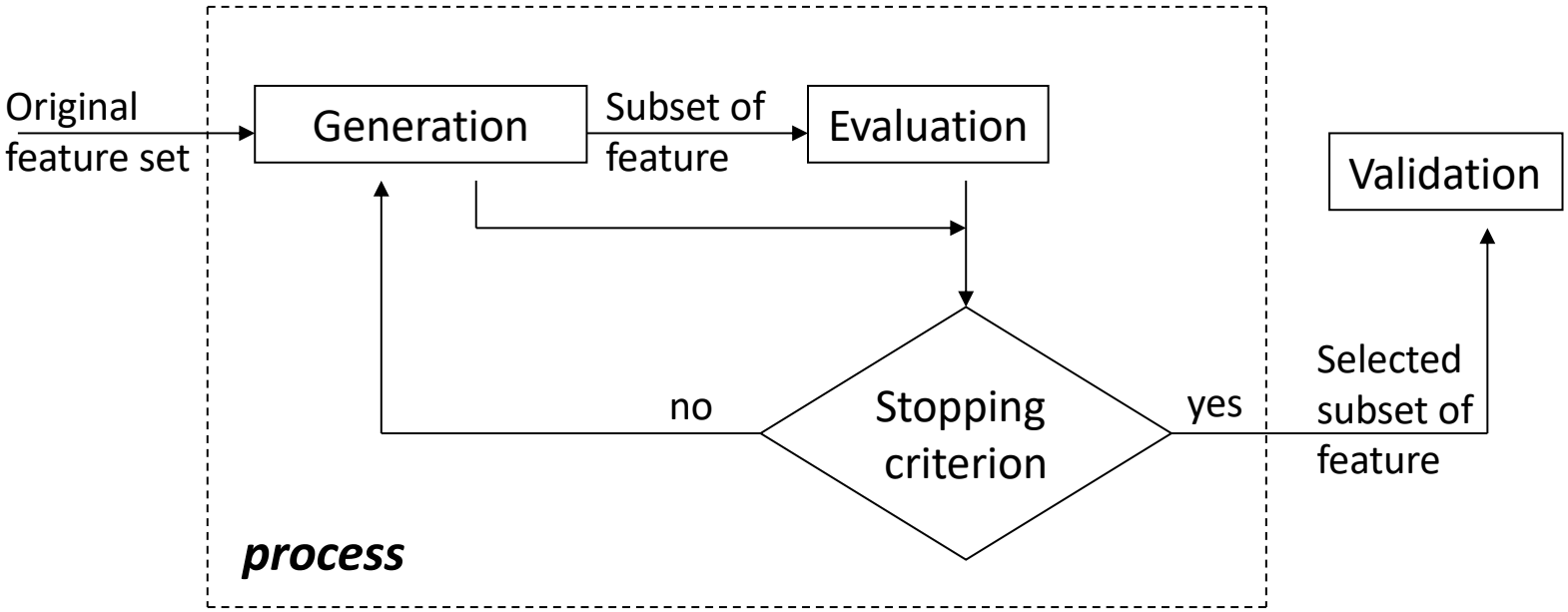
- **Main goal:** rank subsets of useful features.



- **Danger of over-fitting** with intensive search!

Feature selection: the ML perspective

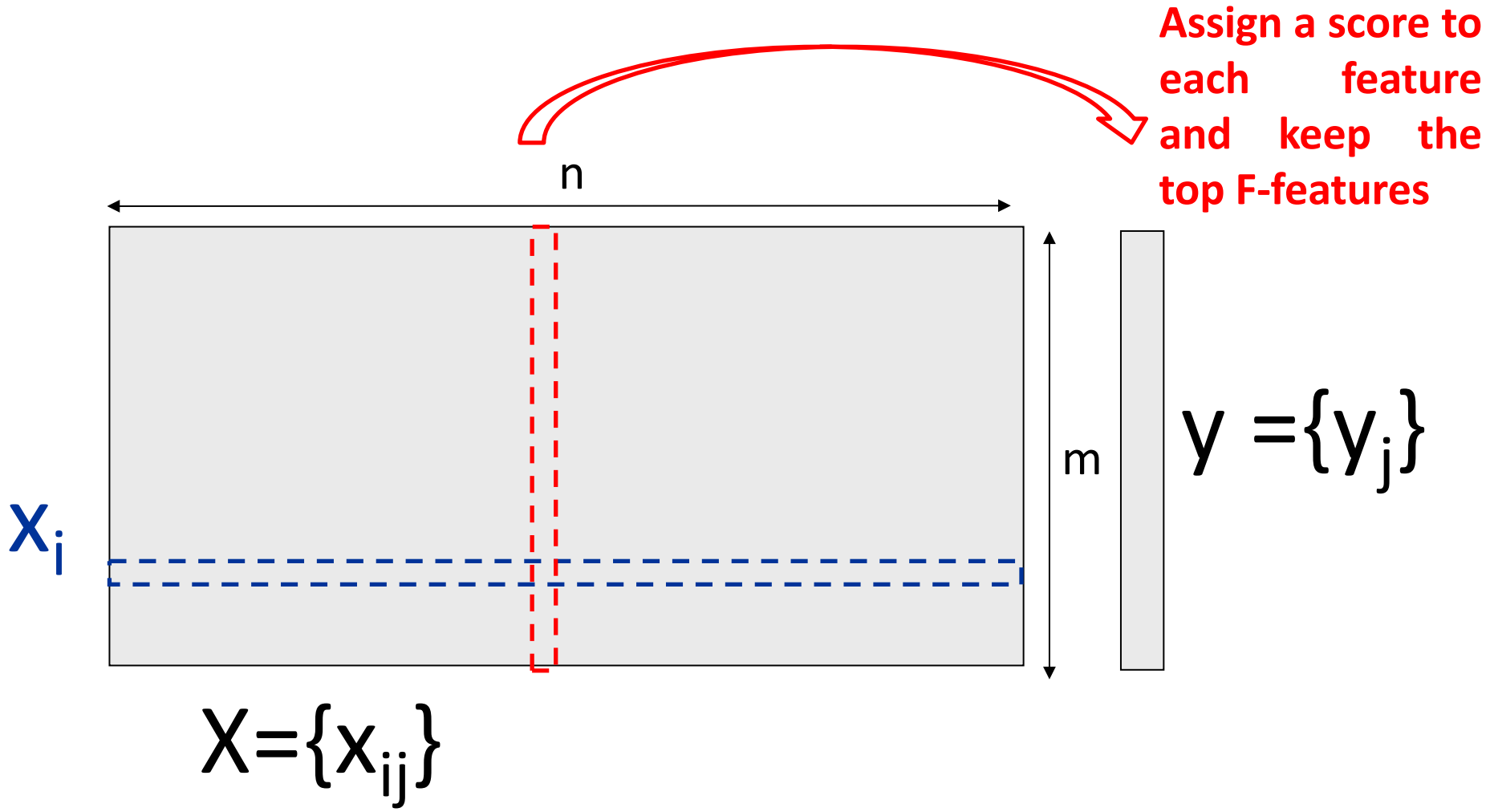
- General diagram of a wrapper feature selection method



Generation = select feature subset candidate.
Evaluation = compute relevancy value of the subset.
Stopping criterion = determine whether subset is relevant.
Validation = verify subset validity.

From M. Dash and H. Liu. http://www.comp.nus.edu.sg/~wongszec/group10.ppt

Feature selection in text mining

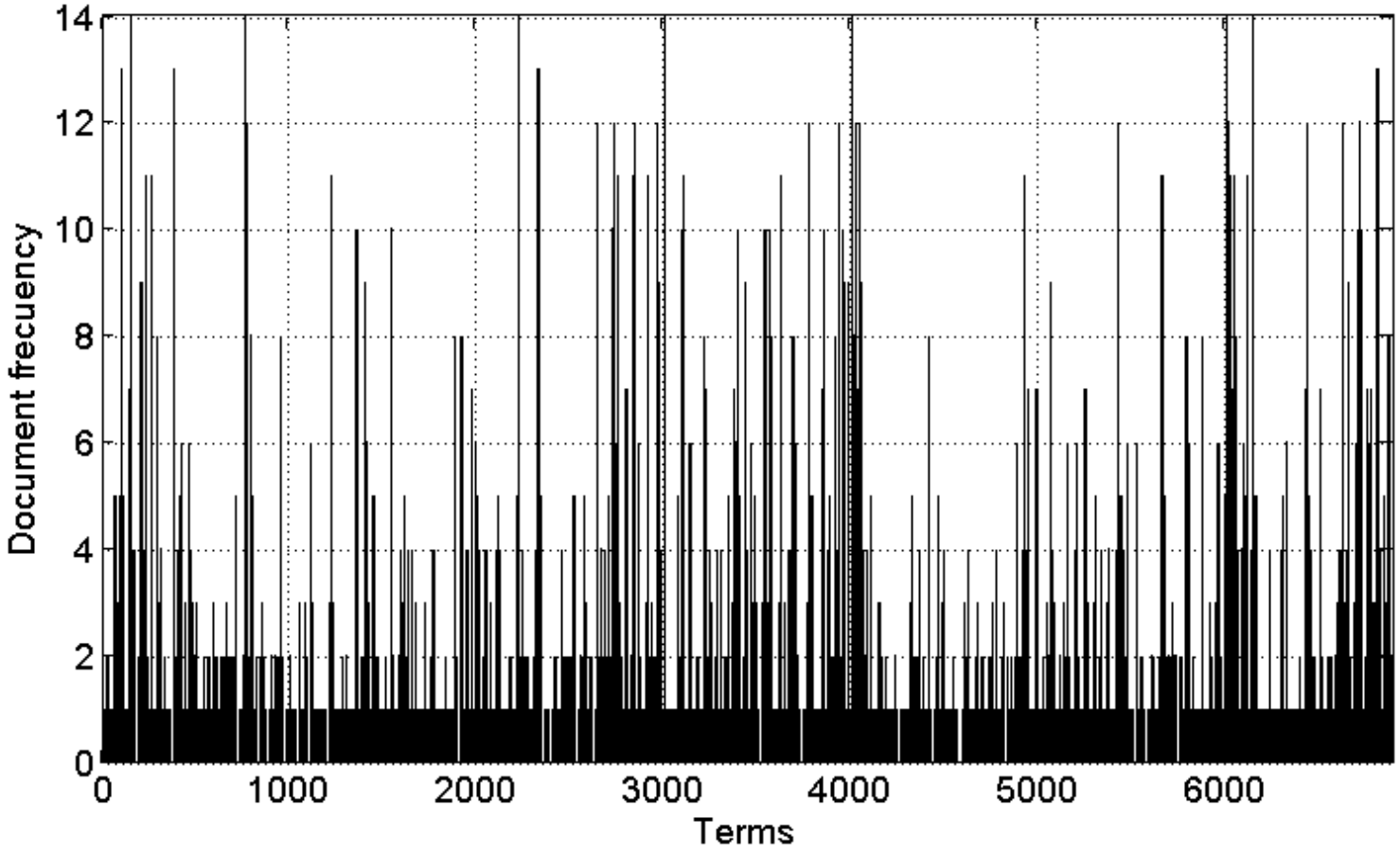


Slide taken from I. Guyon. **Feature and Model Selection.** Machine Learning Summer School, Ile de Re, France, 2008.

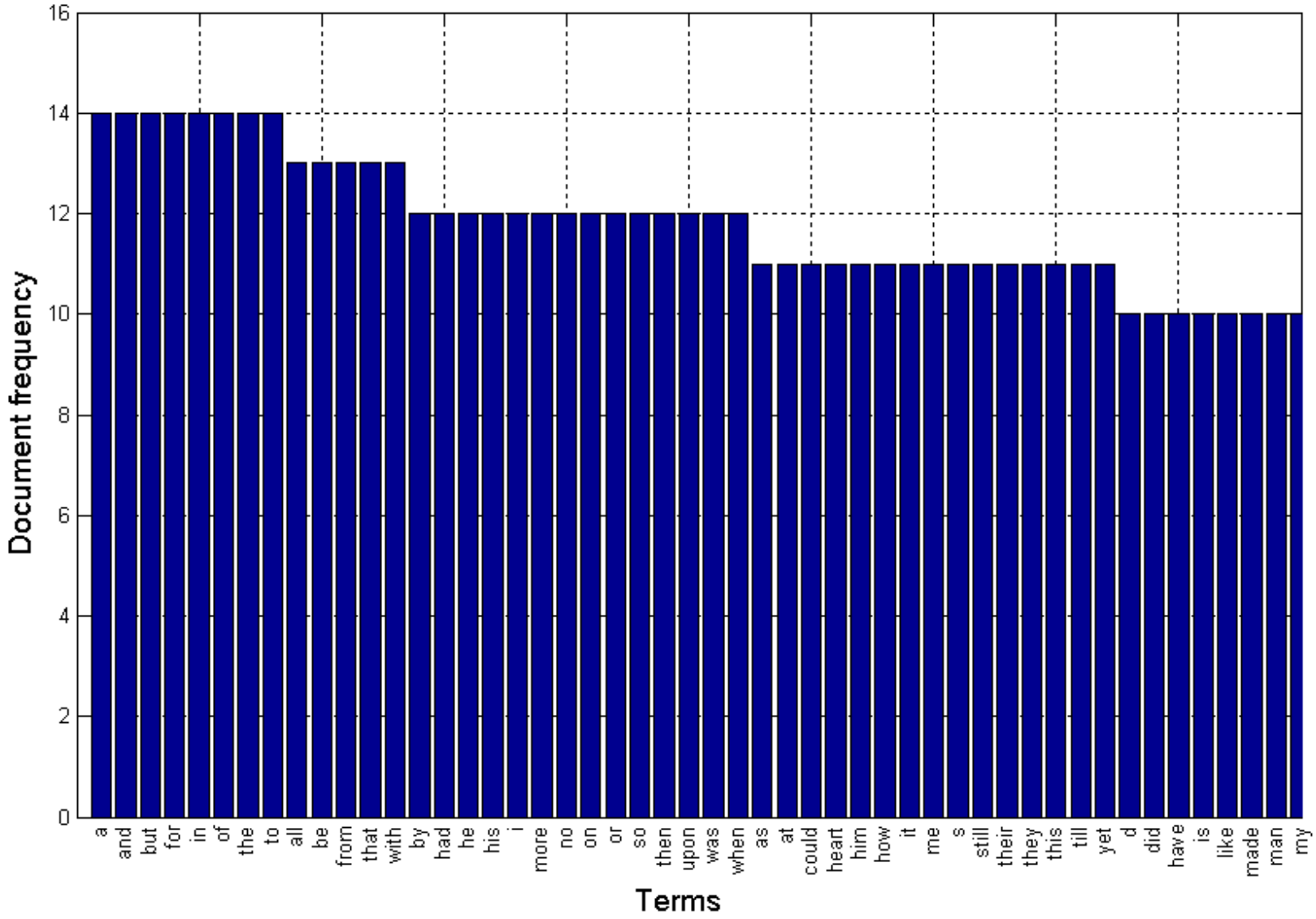
FS: Document frequency

- 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 a specified threshold
- The basic assumption is that *rare words* are either *non-informative* for category prediction or *not influential* in global performance.

FS: Document frequency



FS: Document frequency



Zipf's law

- The frequency of any word is inversely proportional to its rank in the frequency table. Thus the most frequent word will occur approximately twice as often as the second most frequent word, three times as often as the third most frequent word, etc.

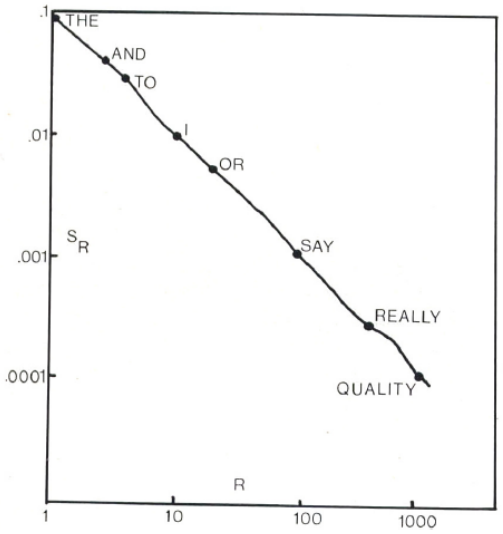


Figure 1 Frequency of word usage in English

$$f_q(w) = \frac{1}{rank(w)^p}$$

G. Kirby. **Zipf's law.** UK Journal of Naval Science Volume 10, No. 3 pp 180 – 185, 1985.

FS: 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 class

FS: Mutual information

- Let:

  A: # times t and c co-occur
  B: # times t occurs without c
  C: # times c occurs without t
  N: # documents

	T	¬ T
c	A	C
¬c	B	D

- Then:

$$I(t,c) \approx \log \frac{A \times N}{(A+C) \times (A+B)}$$

- To get the global MI for term *t*:

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

FS: 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?

FS: Information gain (2)

Entropy: Average information from a message that can take m values

- Information gain → *Entropy*

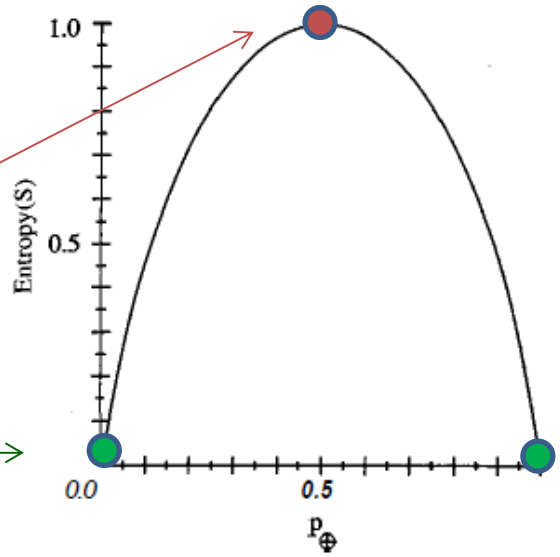
- Entropy characterizes the impurity of an arbitrary collection of examples.
  - It specifies the minimum number of bits of information needed to *encode the classification* of an arbitrary member of the dataset (*S*).

- 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



FS: Information gain (3)

$$G(t) = -\sum_{i=1}^{m} P_r(c_i) \log P_r(c_i) + P_r(t) \sum_{i=1}^{m} P_r(c_i|t) \log P_r(c_i|t) + P_r(\bar{t}) \sum_{i=1}^{m} P_r(c_i|\bar{t}) \log P_r(c_i|\bar{t})$$

- 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.

FS: Information gain (4)

Definition [edit]

If $H(Y|X=x)$ is the entropy of the variable Y conditioned on the variable X taking a certain value x, then $H(Y|X)$ is the result of averaging $H(Y|X=x)$ over all possible values x that X may take.

Given discrete random variables X with domain $\mathcal{X}$ and Y with domain $\mathcal{Y}$, the conditional entropy of Y given X is defined as:[1]

$$\begin{aligned} H(Y|X) &\equiv \sum_{x \in \mathcal{X}} p(x)\, H(Y|X=x) \\ &= -\sum_{x \in \mathcal{X}} p(x) \sum_{y \in \mathcal{Y}} p(y|x)\, \log\, p(y|x) \end{aligned}$$

Other FS methods for TC

Name	Description	Formula
Acc	Accuracy	$tp - fp$
Acc2	Accuracy balanced[†]	$\lvert tpr - fpr \rvert$
BNS	Bi-Normal Separation[†]	$\lvert F^{-1}(tpr) - F^{-1}(fpr) \rvert$ where F is the Normal c.d.f.
Chi	Chi-Squared[‡]	$t(tp, (tp+fp)P_{pos}) + t(fn, (fn+tn)P_{pos}) + t(fp, (tp+fp)P_{neg}) + t(tn, (fn+tn)P_{neg})$ where $t(count, expect) = (count - expect)^2 / expect$
DFreq	Document Frequency[†‡°]	$tp + fp$
F1	F_1-Measure	$\frac{2\,recall\,precision}{(recall + precision)} = \frac{2tp}{(pos + tp + fp)}$
IG	Information Gain[†‡]	$e(pos, neg) - [P_{word}\, e(tp, fp) + P_{\overline{word}}\, e(fn, tn)]$ where $e(x,y) = -\frac{x}{x+y}\log_2\frac{x}{x+y} - \frac{y}{x+y}\log_2\frac{y}{x+y}$
OddN	Odds Ratio Numerator	$tpr\,(1 - fpr)$
Odds	Odds Ratio[†]	$\frac{tpr(1-fpr)}{(1-tpr)\,fpr} = \frac{tp\ \ tn}{fp\ fn}$
Pow	Power	$(1 - fpr)^k - (1 - tpr)^k$ where k=5
PR	Probability Ratio	tpr / fpr
Rand	Random[‡°]	random()

[†] Acc2, BNS, DFreq, IG, and Odds select a substantial number of negative features.
[‡] Chi, IG, DFreq, and Rand also generalize for multi-class problems.
[°] DFreq and Rand do not require the class labels.

G. Forman. **An Extensive Empirical Study of Feature Selection Metrics for Text Classification.** JMLR, 3:1289—1305, 2003

Other FS methods for TC

Notation:

tp: true positives = number of positive cases containing word
fp: false positives = number of negative cases containing word
pos: number of positive cases = $tp + fn$
neg: number of negative cases = $fp + tn$
tpr: sample true positive rate = tp / pos
fpr: sample false positive rate = fp / neg
precision = $tp / (tp+fp)$

fn: false negatives
tn: true negatives
$P_{pos} = pos / all$
$P_{neg} = neg / all$
$P_{word} = (tp+fp) / all$
$P_{\overline{word}} = 1\text{-}P(word)$
recall = *tpr*

Note: Metrics such as BNS, Chi and IG are naturally symmetric with respect to negatively correlated features. For the metrics that devalue all negative features, we invert any negative feature, i.e. $tpr' = 1 - tpr$ and $fpr' = 1 - fpr$, without reversing the classes. Hence, without loss of generality, $tpr > fpr$.

G. Forman. **An Extensive Empirical Study of Feature Selection Metrics for Text Classification.** JMLR, 3:1289—1305, 2003