ECE8813
Statistical Natural Language Processing

Lectures 19-20: Text Categorization

Chin-Hui Lee

School of Electrical and Computer Engineering

Georgia Institute of Technology

Atlanta, GA 30332, USA

chl@ece.gatech.edu

What is Text Classification?

- We are given:
  - a fixed set of categories: $C=\{c_1, c_2, \ldots, c_n\}$
  - a document d_j □ *D*, where *D* is the domain of documents
- We want to:
  - assign a Boolean value to the pair $<d_j, c_i>$
  - if the value is T, the the d_j is classified under category c_i, otherwise it is not
- We essentially want to build categorization functions (classifiers) that assign these values

An example: Is this mail spam?

From: lotterias-espana@zwallet.com [mailto:lotterias-espana@zwallet.com]

Sent: Wednesday, June 30, 2004 12:26 PM

Subject: FINAL AWARD WINNING NOTIFICATION

FROM: The Desk of the Managing Director

International Promotion Prize Award Dept.

Ref:LP523275/2003/ES

BATCH:02033/1PD

RE: Final award winning notification.

We are pleased to inform you about the release today the 30th of june 2004 of sweepstake Loteria Primitiva de España held on the 24th may 2004, your name attached to ticket number: 524- 412-56-ES, with serial number 4253/03 drew the lucky number:75-23-58-46-51, which consequently won the lottery in the 3rd category. You have therefore been approved for a lump sum pay out of €500,000.00 euros (five hundred thousand euros) in cash credited to file:lp523275/2003/es. This form is from a total cash prize of €2 million euros share! among the four international lucky winners in this category. furthermore, your lucky winning number falls within our European booklet representative office in Madrid - Spain as indicated in your play coupon. in view of this, your €500,000.00 (five hundred thousand euros) would be released to you by our private security and trust company which had insured your winning in your name with their office in Madrid - Spain, congratulations!

An Example: Language Identification

Die Ausstellung zeigt den Einfluss der Freien Universität auf wissenschafts- und gesellschaftspolitische Entwicklungen im nationalen und internationalen Raum. Im Mittelpunkt stehen die Gründung der FU als Reaktion auf die Relegation, Verhaftung und Drangsalierung demokratisch orientierter Studenten im Jahre 1948, ihre Rolle bei den Studentenunruhen 1968, die Folgen des Mauerfalls 1989 sowie künftige Pläne für den Wissenschaftsstandort Dahlem. Weitere thematische Schwerpunkte sind die Architektur des Universitätsgeländes mit Bauten aus sechs Jahrzehnten, das breit gefächerte Spektrum der angebotenen Wissenschaften, das Leben auf dem Campus sowie Habitus und Ritual der akademischen Welt damals und heute.

Giorno della Memoria - La Casa dello Studente, uno dei luoghi più evocativi legati alle vicende dell'oppressione nazista a Genova e in Liguria e di alto significato morale per la storia della Liberazione, sarà aperto al pubblico per iniziativa dell'Università di Genova e dell'ERSU e permetterà la visita alle "celle" della sede del Comando delle S.S. (1943-1945) 31 gennaio 2005

- Here the decision may be more than binary: given a set of languages (English, French, Italian, German, Spanish, Portuguese, etc.) what language does a given text belong to?

Text Classification Examples

- Assign categories to web pages
  - e.g. sports:football, news:world:asia, finance, etc.
- Find the genre of a given web page
  - e.g. research page, news article, review page, etc.
- Categories may be binary
  - “spam”, non-spam”
  - “interesting-to-me”, “not-interesting-to-me”
  - “appropriate-for-kids”, “not-appropriate-for-kids”
  - etc.

Applications

- Document organisation
  - e.g. a newspaper that wants to “classified adds” put into categories such as “Car sales”, “Property Rental”, “Personals”, etc.
- Text filtering
  - classify a stream of incoming documents depending on their relevance to the information consumer
  - typically a binary case (relevant – not relevant)
  - common to have a profile for the information consumer
    - the profile can be updated depending on the consumer’s implicit or explicit relevance assessments on the provided information (adaptive filtering)

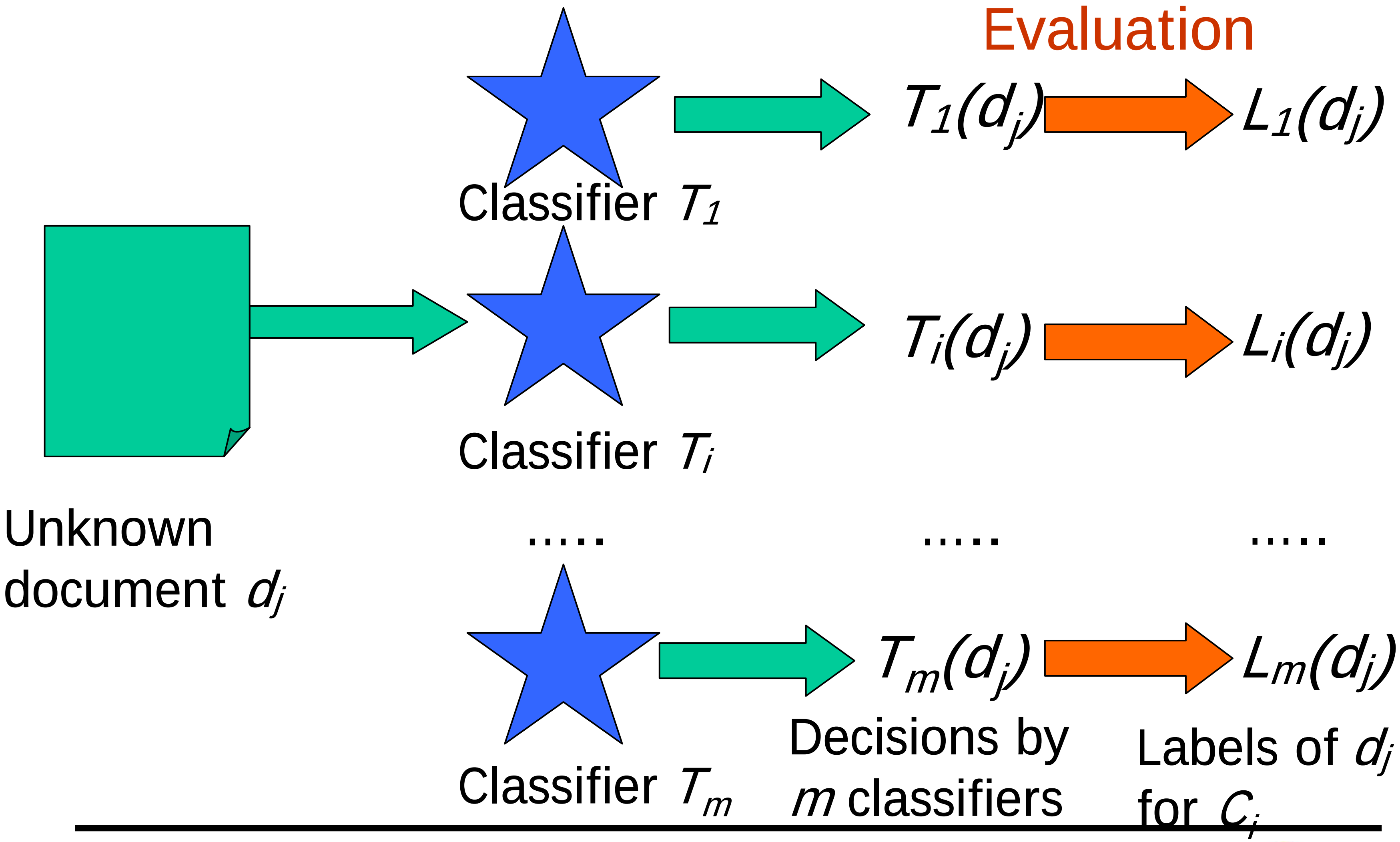
Applications (Cont.)

- Word sense disambiguation
  - e.g. “bank”: financial institution, or river bank?
  - we can view word occurrence contexts as documents, and word senses as categories
  - we have a number of “documents” put in the correct “categories”, and try to find the correct word sense for a new incoming word occurrence context
- Hierarchical categorisation of web pages
  - automatically classify pages under the hierarchical catalogue of e.g. Yahoo
  - searchers may find it easier to navigate in a hierarchy
  - the hypertextual nature of web pages is useful (one can take into advantage the links between pages)
  - the hierarchical structure of the categories is also useful
    - e.g. decompose the classification problem to a number of branching decisions at each internal node

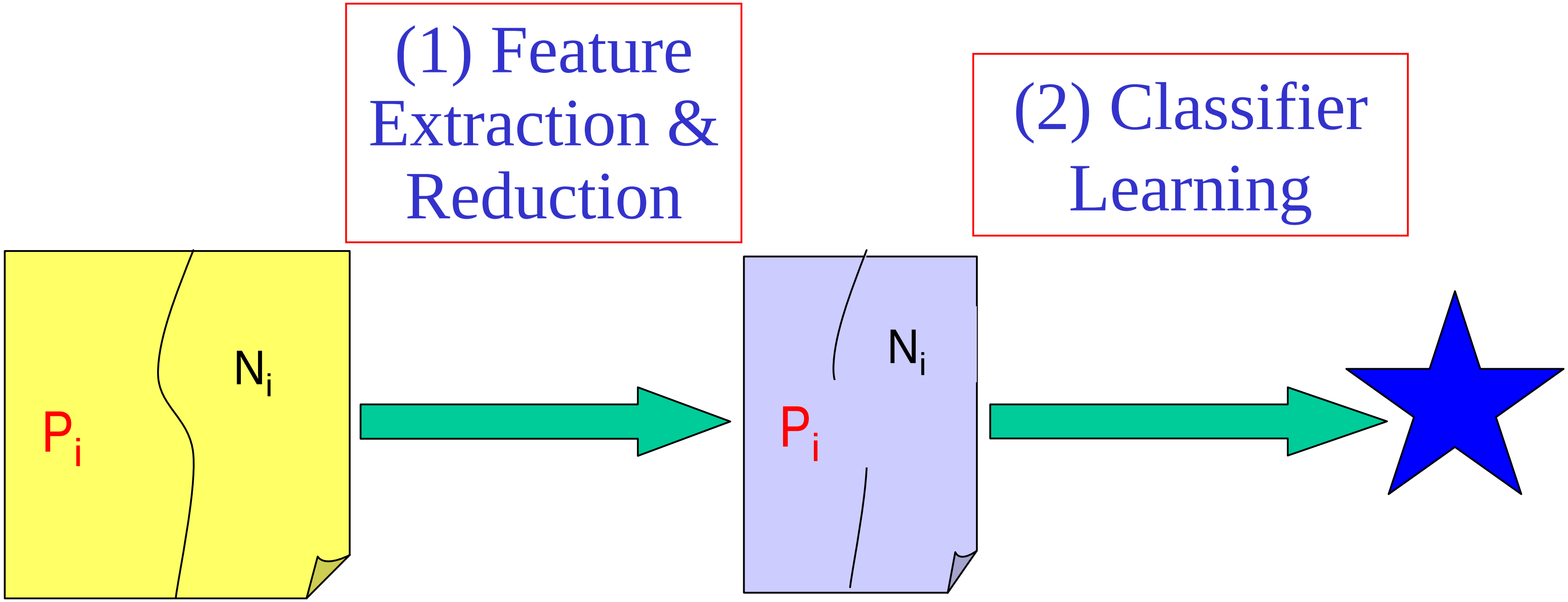
The Main Approach to Classification

- The machine learning approach
  - build a class for classifier for a category c_i by observing the properties of the set of documents manually classified under ci (learning)
  - from these properties, get the properties that an unseen document should have in order to be classified under c_i
  - this is a case of supervised learning
- The knowledge engineering approach
  - need a large set of rules *if* <> *then <category>*
  - rules manually constructed
  - major drawback: *knowledge acquisition bottleneck*, i.e. how do you deal with new categories, different domain, etc.

Text Categorization – Topic Identification



Text Categorization: Training Classifiers



(1) Feature Extraction & Reduction

(2) Classifier Learning

Training set for each category C_i , i= 1,…,m. (Positive +Negative)

Doc. in new feature space

Classifier T_i for category C_i

Multi-Class vs. Binary Decision Rule

- Multi-class (MC) classification

$$C(X) = \arg\max_{j} g_j(X;W), \quad 1 \le j \le m$$

if $$g_j(X;W) > g_{i \ne j}(X;W)$$

- Special case: Binary classifier with LDF *(C+: positive class, C-: negative class)*

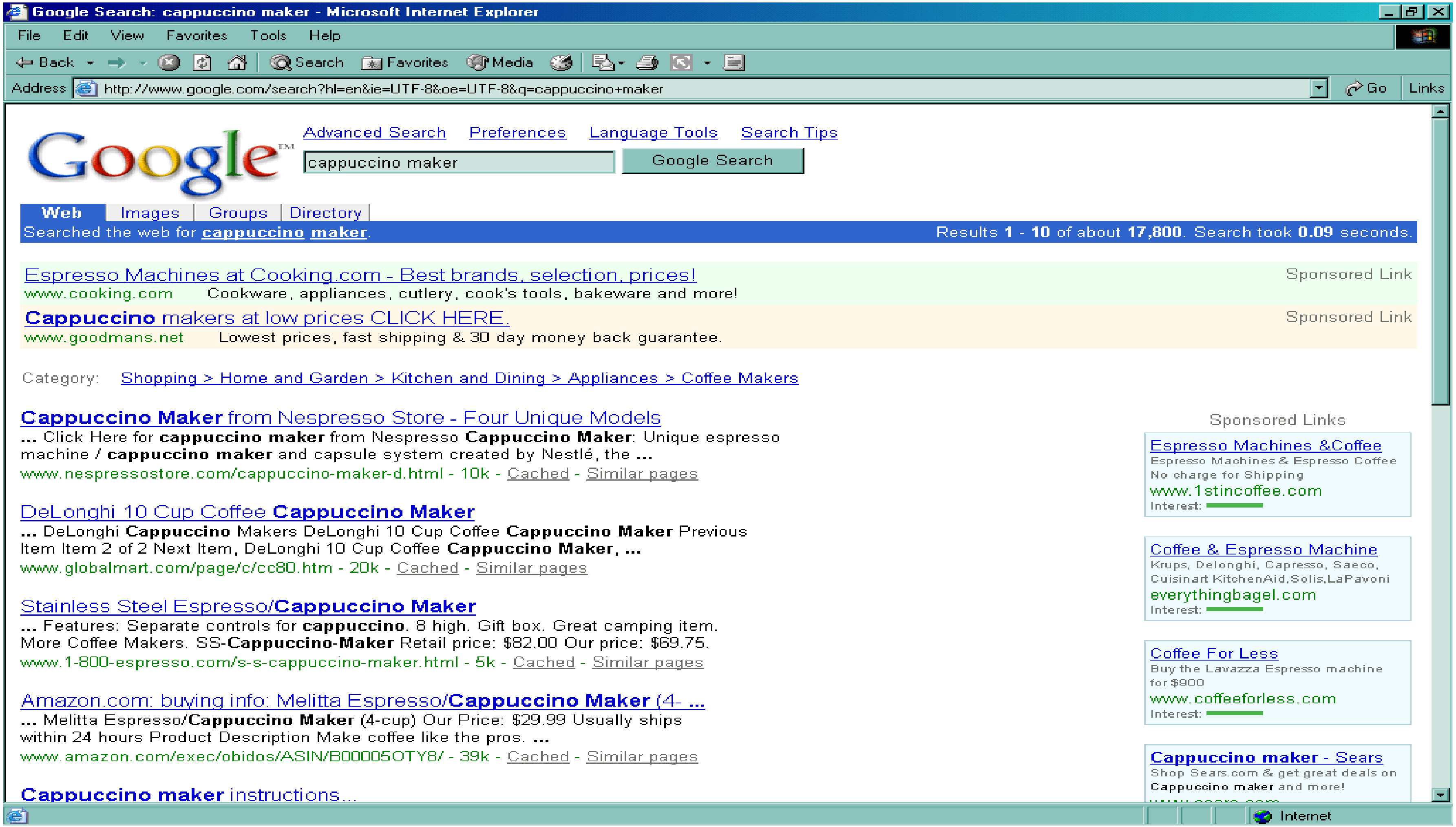
$$\begin{cases} f(W,X) \ge 0 & \text{label C+} \\ \text{Others} & \text{label C-} \end{cases}$$

➢ **Decision rule is a** ***discrete, non-differential function of the classifier parameters (need MFoM to optimize)***

A Text Categorization Scenario

- Suppose you want to buy a cappuccino maker as a gift on the web
  - try Google for “cappuccino maker”
  - try “Yahoo! Shopping” for “cappuccino maker”

Google Search Results



Searched the web for **cappuccino maker**. Results **1** - **10** of about **17,800**. Search took **0.09** seconds.

Espresso Machines at Cooking.com - Best brands, selection, prices! — Sponsored Link
www.cooking.com Cookware, appliances, cutlery, cook's tools, bakeware and more!

Cappuccino makers at low prices CLICK HERE. — Sponsored Link
www.goodmans.net Lowest prices, fast shipping & 30 day money back guarantee.

Category: Shopping > Home and Garden > Kitchen and Dining > Appliances > Coffee Makers

Cappuccino Maker from Nespresso Store - Four Unique Models
... Click Here for **cappuccino maker** from Nespresso **Cappuccino Maker**: Unique espresso machine / **cappuccino maker** and capsule system created by Nestlé, the **...**
www.nespressostore.com/cappuccino-maker-d.html - 10k - Cached - Similar pages

DeLonghi 10 Cup Coffee **Cappuccino Maker**
... DeLonghi **Cappuccino** Makers DeLonghi 10 Cup Coffee **Cappuccino Maker** Previous Item Item 2 of 2 Next Item, DeLonghi 10 Cup Coffee **Cappuccino Maker**, **...**
www.globalmart.com/page/c/cc80.htm - 20k - Cached - Similar pages

Stainless Steel Espresso/**Cappuccino Maker**
... Features: Separate controls for **cappuccino**. 8 high. Gift box. Great camping item. More Coffee Makers. SS-**Cappuccino-Maker** Retail price: $82.00 Our price: $69.75.
www.1-800-espresso.com/s-s-cappuccino-maker.html - 5k - Cached - Similar pages

Amazon.com: buying info: Melitta Espresso/**Cappuccino Maker** (4- **...**
... Melitta Espresso/**Cappuccino Maker** (4-cup) Our Price: $29.99 Usually ships within 24 hours Product Description Make coffee like the pros. **...**
www.amazon.com/exec/obidos/ASIN/B00005OTY8/ - 39k - Cached - Similar pages

Cappuccino maker instructions...

Sponsored Links

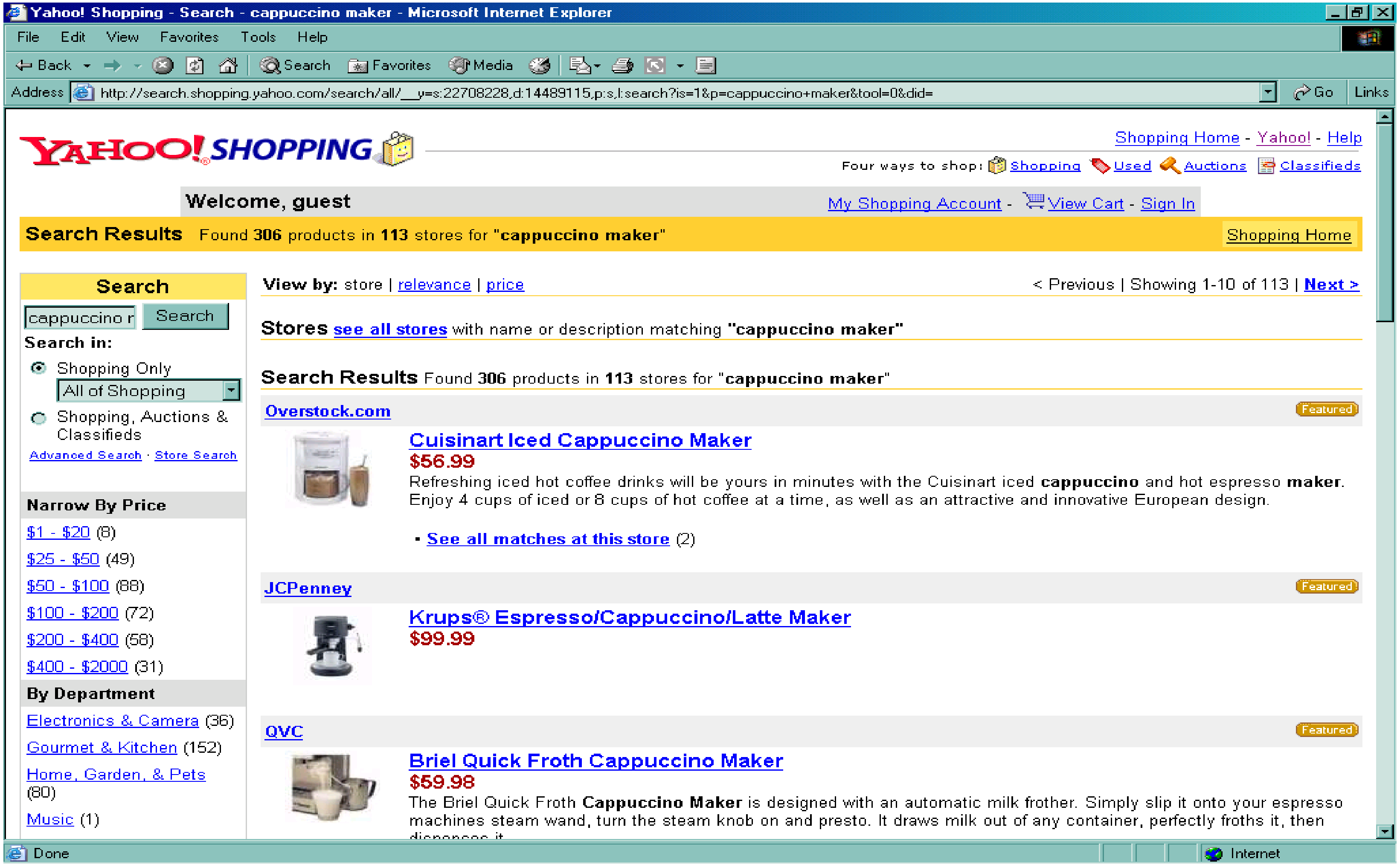
Espresso Machines &Coffee
Espresso Machines & Espresso Coffee
No charge for Shipping
www.1stincoffee.com
Interest:

Coffee & Espresso Machine
Krups, Delonghi, Capresso, Saeco, Cuisinart KitchenAid,Solis,LaPavoni
everythingbagel.com
Interest:

Coffee For Less
Buy the Lavazza Espresso machine for $900
www.coffeeforless.com
Interest:

Cappuccino maker - Sears
Shop Sears.com & get great deals on Cappuccino maker and more!

Yahoo Search Results



Yahoo! Shopping - Search - cappuccino maker - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://search.shopping.yahoo.com/search/all/__y=s:22708228,d:14489115,p:s,l:search?is=1&p=cappuccino+maker&tool=0&did=
Go Links

YAHOO! SHOPPING

Shopping Home - Yahoo! - Help

Four ways to shop: Shopping Used Auctions Classifieds

Welcome, guest

My Shopping Account - View Cart - Sign In

Search Results Found **306** products in **113** stores for "**cappuccino maker**"

Shopping Home

Search

cappuccino r Search

Search in:

● Shopping Only

All of Shopping

○ Shopping, Auctions & Classifieds

Advanced Search · Store Search

Narrow By Price

- $1 - $20 (8)
- $25 - $50 (49)
- $50 - $100 (88)
- $100 - $200 (72)
- $200 - $400 (58)
- $400 - $2000 (31)

By Department

- Electronics & Camera (36)
- Gourmet & Kitchen (152)
- Home, Garden, & Pets (80)
- Music (1)

View by: store | relevance | price

< Previous | Showing 1-10 of 113 | **Next >**

Stores **see all stores** with name or description matching "**cappuccino maker**"

Search Results Found **306** products in **113** stores for "**cappuccino maker**"

Overstock.com Featured

Cuisinart Iced Cappuccino Maker

$56.99

Refreshing iced hot coffee drinks will be yours in minutes with the Cuisinart iced **cappuccino** and hot espresso **maker**. Enjoy 4 cups of iced or 8 cups of hot coffee at a time, as well as an attractive and innovative European design.

- **See all matches at this store** (2)

JCPenney Featured

Krups® Espresso/Cappuccino/Latte Maker

$99.99

QVC Featured

Briel Quick Froth Cappuccino Maker

$59.98

The Briel Quick Froth **Cappuccino Maker** is designed with an automatic milk frother. Simply slip it onto your espresso machines steam wand, turn the steam knob on and presto. It draws milk out of any container, perfectly froths it, then dispenses it

Observations

- Broad indexing & speedy search alone are not enough
- Organizational view of data is critical for effective retrieval
- Categorized data are easy for user to browse
- Category taxonomies become most central in well-known web sites (Yahoo!, Lycos, ...)

Categorization/Classification

Given:

A description of an instance, $x \in X$, where X is the *instance language* or *instance space*

Issue: how to represent text documents?

Example: A fixed set of categories:

$C = \{c_1, c_2, \ldots, c_n\}$

Determine:

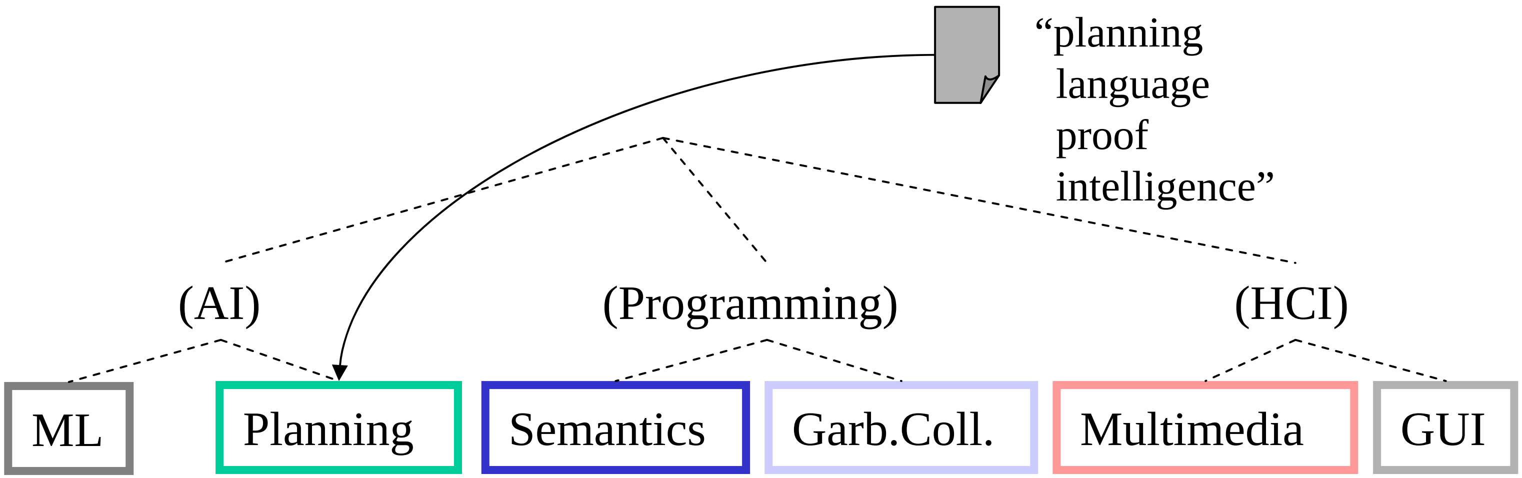
The category of x: $c(x) \in C$, where $c(x)$ is a *categorization function* whose domain is X and whose range is C

We want to know how to build categorization functions (“classifiers”), and often involve computing a score, or a goodness-of-fit function for each x and each $c(x) \in C$

Document Classification (Topic ID)

Test Data:

Classes:



Training Data:	learning intelligence algorithm reinforcement network...	planning temporal reasoning plan language...	programming semantics language proof...	garbage collection memory optimization region...	...	...

(Note: in real life there is often a hierarchy, not present in the above problem statement; and you get papers on ML approaches to Garb. Coll.)

Text Categorization Examples

- Assign labels to each document or web-page:
- Labels are most often topics such as Yahoo-categories
  - *e.g., "finance," "sports," "news>world>asia>business"*
- Labels may be genres
  - *e.g., "editorials" "movie-reviews" "news“*
- Labels may be opinion
  - *e.g., “like”, “hate”, “neutral”*
- Labels may be domain-specific binary
  - *e.g., "interesting-to-me" : "not-interesting-to-me”*
  - *e.g., “spam” : “not-spam”*
  - *e.g., “contains adult language” :“doesn’t”*

Text Categorization Applications

- Web pages organized into category hierarchies
- Journal articles indexed by subject categories (e.g., the Library of Congress, MEDLINE, etc.)
- Responses to Census Bureau occupations
- Patents archived using *International Patent Classification*
- Patient records coded using international insurance categories
- E-mail message filtering
- News events tracked and filtered by topics

Cost of Manual Text Categorization

- Yahoo!
  - 200 (?) people for manual labeling of Web pages
  - using a hierarchy of 500,000 categories
- MEDLINE (National Library of Medicine)
  - $2 million/year for manual indexing of journal articles
  - using MEdical Subject Headings (18,000 categories)
- Mayo Clinic
  - $1.4 million annually for coding patient-record events
  - using the International Classification of Diseases (ICD) for billing insurance companies
- US Census Bureau decennial census (1990: 22 million responses)
  - 232 industry categories and 504 occupation categories
  - $15 million if fully done by hand

Fast Entry is a Must to Compete

- Suppose you were starting a web search company, what would it take to compete with established engines?
  - You need to be able to establish a competing hierarchy *fast*
  - You will need a relatively *cheap* solution. (Unless you have investors that want to pay millions of dollars just to get off the ground)

Semi-Automatic Labeling

- Humans can encode knowledge of what constitutes membership in a category
- This encoding can then be automatically applied by a machine to categorize new examples
- For example...Text in a Web Page

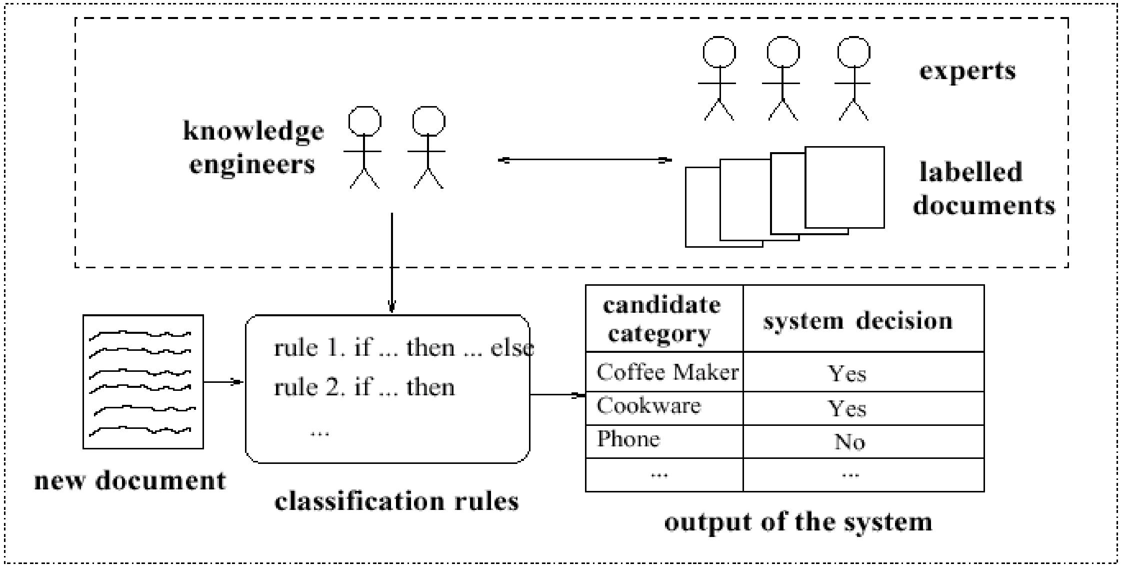
“Saeco revolutionized *espresso* brewing a decade ago by introducing Saeco SuperAutomatic *machines*, which go from bean to *coffee* at the touch of a button. The all-new Saeco Vienna Super-Automatic home coffee and *cappucino machine* combines top quality with low price!”

Rule-based Approach to TC

- Rules
  - Rule 1:
    (*espresso* **or** *coffee* **or** *cappucino*) **and** *machine** ⇨ *Coffee Maker*
  - Rule 2:
    *automat** **and** *answering* **and** *machine** ⇨ *Phone*
  - Rule ...
- Experience has shown that defining rules by hands is
  - too time consuming
  - too difficult
  - inconsistency issues (as the rule set gets large)

Expert System for TC (Late 1980s)

Expert system for text categorization (late 1980s)



knowledge engineers

experts

labelled documents

new document

rule 1. if ... then ... else
rule 2. if ... then
...

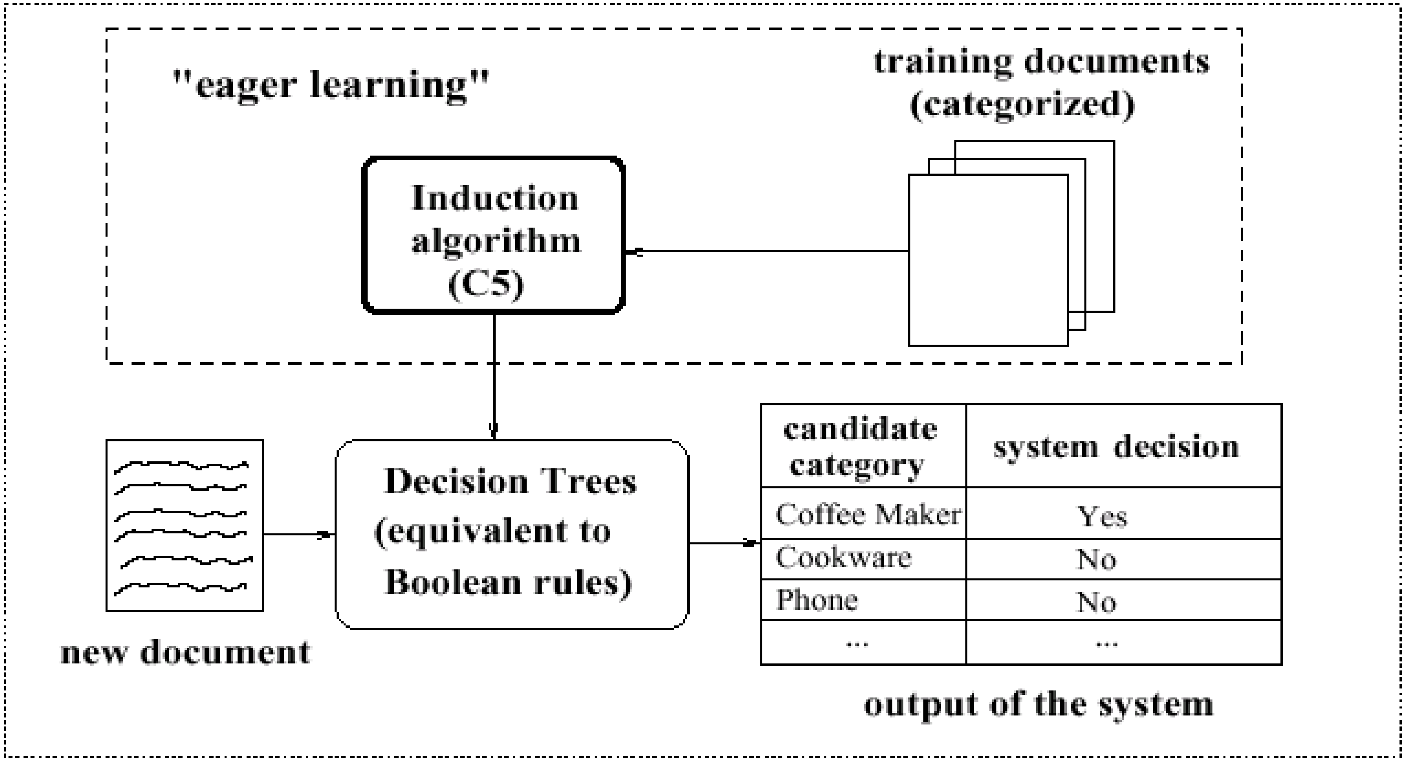
classification rules

candidate category	system decision
Coffee Maker	Yes
Cookware	Yes
Phone	No
...	...

output of the system

From Knowledge Engineering to Statistical Learner

DTree induction for text categorization (since 1994)


"eager learning"
training documents (categorized)
Induction algorithm (C5)
Decision Trees (equivalent to Boolean rules)
new document
output of the system

candidate category	system decision
Coffee Maker	Yes
Cookware	No
Phone	No
...	...

A Comparison: Another Familiar Story

- For US Census Bureau Decennial Census 1990
  - 232 industry categories and 504 occupation categories
  - $15 million if fully done by hand
- Define classification rules manually:
  - Expert System AIOCS
  - Development time: 192 person-months (2 people, 8 years)
  - Accuracy = 47%
- Learn classification function
  - Nearest Neighbor classification (Creecy ’92: 1-NN)
  - Development time: 4 person-months (Thinking Machine)
  - Accuracy = 60%

An Example: Predicting Topics of News Stories

- Given: Collection of example news stories already labeled with a category (topic)

- Task: Predict category for news stories not yet labeled

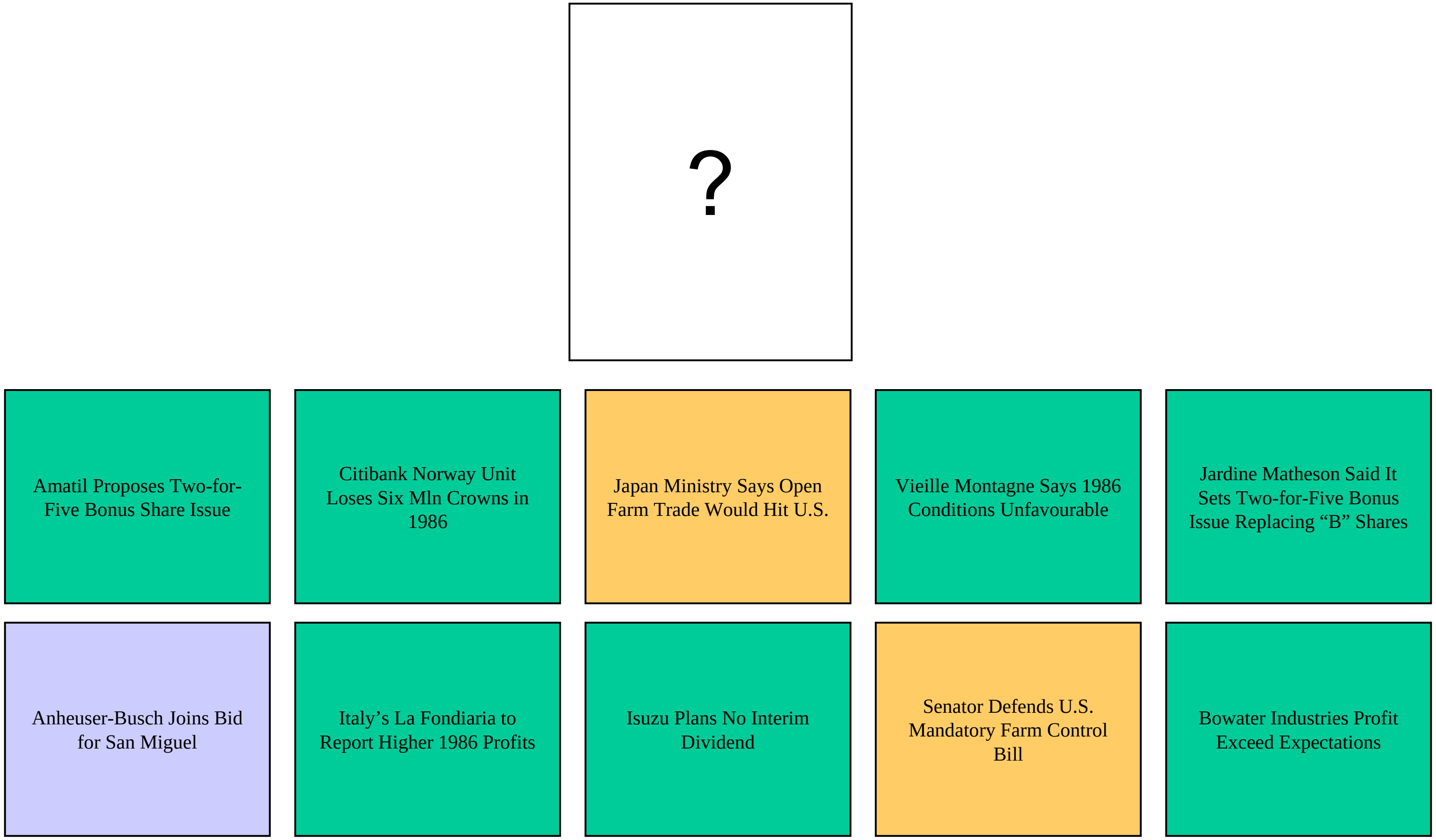
- For our example, we'll only get to see the headline of the news story

- We'll represent categories using colors (All examples with the same color belong to the same category)

Our Labeled Examples

🟩 Amatil Proposes Two-for-Five Bonus Share Issue	🟩 Citibank Norway Unit Loses Six Mln Crowns in 1986	🟨 Japan Ministry Says Open Farm Trade Would Hit U.S.	🟩 Vieille Montagne Says 1986 Conditions Unfavourable	🟩 Jardine Matheson Said It Sets Two-for-Five Bonus Issue Replacing "B" Shares
🟪 Anheuser-Busch Joins Bid for San Miguel	🟩 Italy's La Fondiaria to Report Higher 1986 Profits	🟩 Isuzu Plans No Interim Dividend	🟨 Senator Defends U.S. Mandatory Farm Control Bill	🟩 Bowater Industries Profit Exceed Expectations

Topic Prediction



?

Amatil Proposes Two-for-Five Bonus Share Issue

Citibank Norway Unit Loses Six Mln Crowns in 1986

Japan Ministry Says Open Farm Trade Would Hit U.S.

Vieille Montagne Says 1986 Conditions Unfavourable

Jardine Matheson Said It Sets Two-for-Five Bonus Issue Replacing “B” Shares

Anheuser-Busch Joins Bid for San Miguel

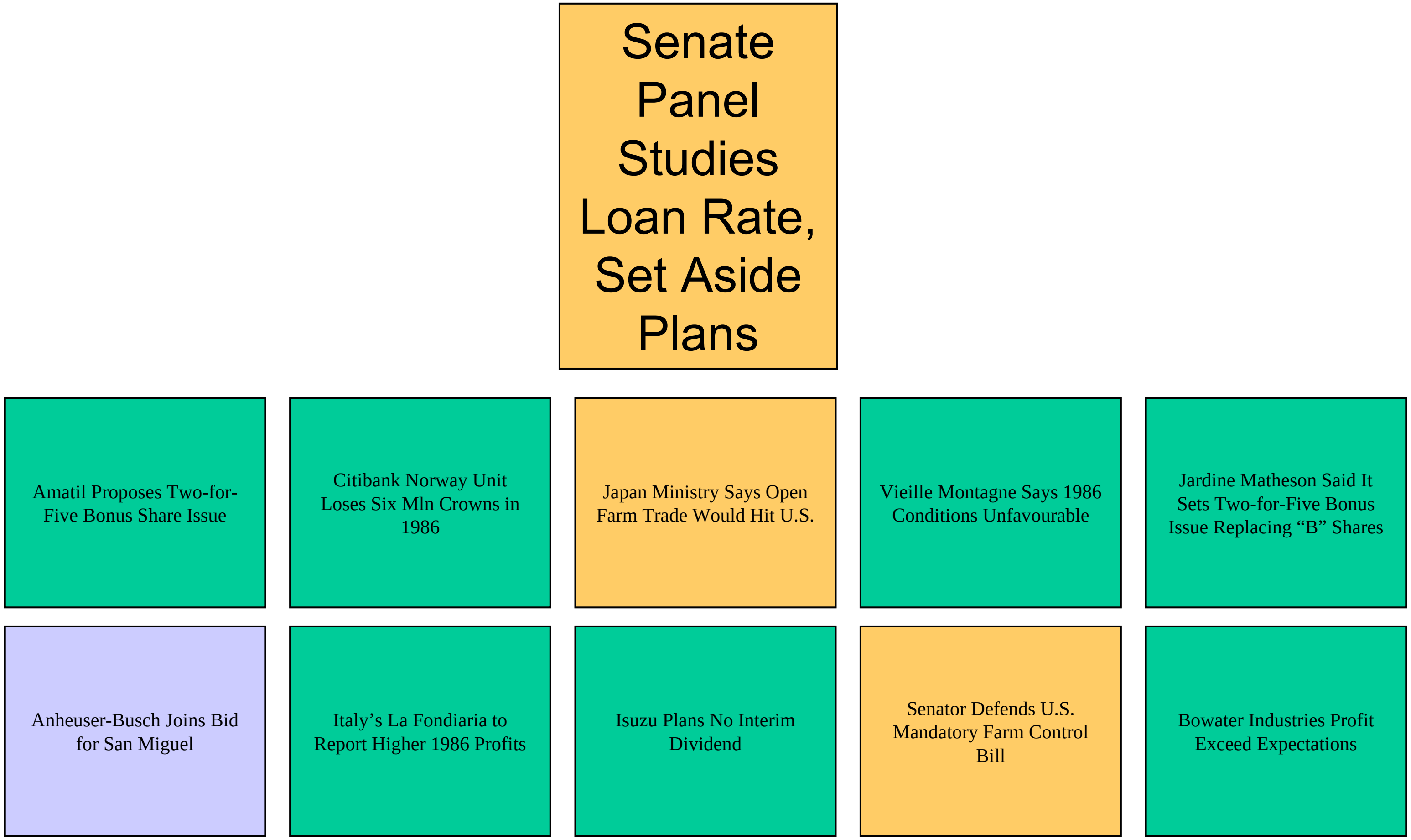
Italy’s La Fondiaria to Report Higher 1986 Profits

Isuzu Plans No Interim Dividend

Senator Defends U.S. Mandatory Farm Control Bill

Bowater Industries Profit Exceed Expectations

Topic Prediction with Evidence



Senate Panel Studies Loan Rate, Set Aside Plans

Amatil Proposes Two-for-Five Bonus Share Issue

Citibank Norway Unit Loses Six Mln Crowns in 1986

Japan Ministry Says Open Farm Trade Would Hit U.S.

Vieille Montagne Says 1986 Conditions Unfavourable

Jardine Matheson Said It Sets Two-for-Five Bonus Issue Replacing "B" Shares

Anheuser-Busch Joins Bid for San Miguel

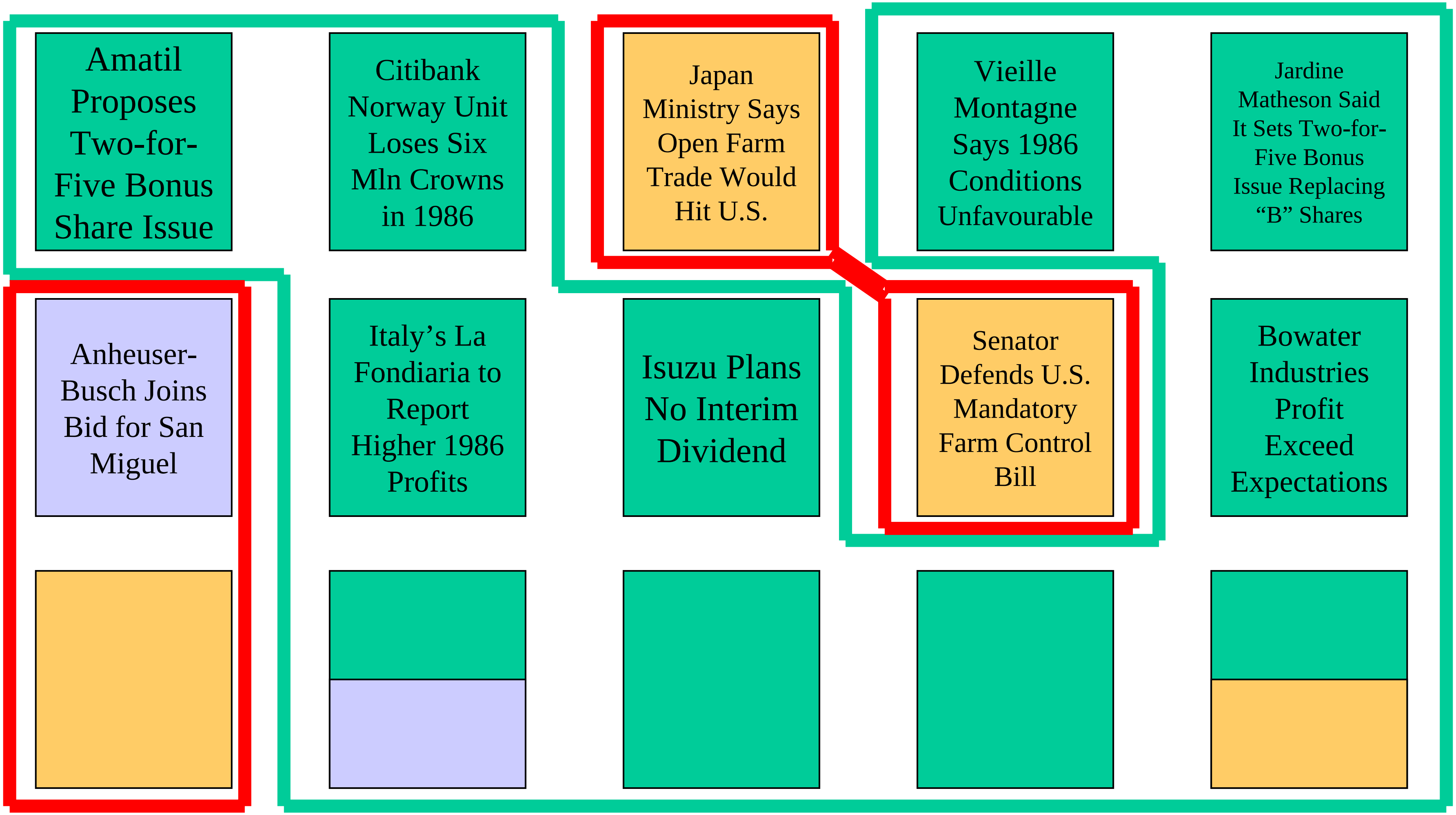
Italy's La Fondiaria to Report Higher 1986 Profits

Isuzu Plans No Interim Dividend

Senator Defends U.S. Mandatory Farm Control Bill

Bowater Industries Profit Exceed Expectations

Handling Documents with Multiple Classes



Amatil Proposes Two-for-Five Bonus Share Issue

Citibank Norway Unit Loses Six Mln Crowns in 1986

Japan Ministry Says Open Farm Trade Would Hit U.S.

Vieille Montagne Says 1986 Conditions Unfavourable

Jardine Matheson Said It Sets Two-for-Five Bonus Issue Replacing “B” Shares

Anheuser-Busch Joins Bid for San Miguel

Italy’s La Fondiaria to Report Higher 1986 Profits

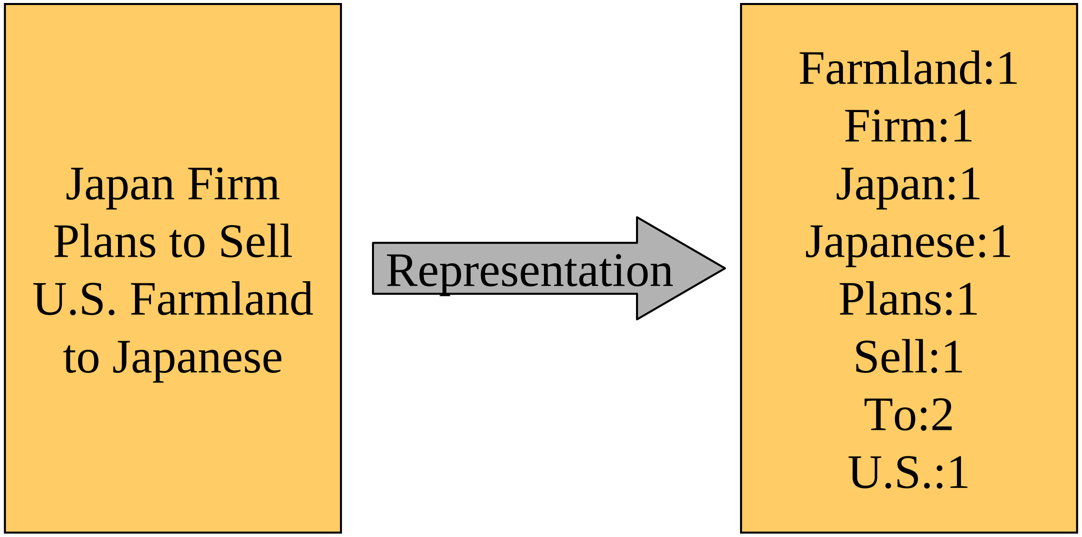
Isuzu Plans No Interim Dividend

Senator Defends U.S. Mandatory Farm Control Bill

Bowater Industries Profit Exceed Expectations

Document Representation

- Usually, an example is represented as a series of feature-value pairs. The features can be arbitrarily abstract (as long as they are easily computable) or very simple.
- For example, the features could be the set of all words and the values, their number of occurrences in a particular document.



Japan Firm Plans to Sell U.S. Farmland to Japanese

Representation

Farmland:1
Firm:1
Japan:1
Japanese:1
Plans:1
Sell:1
To:2
U.S.:1

Performance Evaluation

- Suppose we have a set *D* of labeled documents that we use as our training set for 1-NN. We need an idea of how well this system will perform in the future. So, we go through *D* and make predictions for each document
  - What will our accuracy be?
  - Is this a fair assessment of its performance? (i.e. is it likely that the performance will be within a small tolerance of what we've estimated)

Classification Performance Measures

- Given *n* test documents and *m* classes in consideration, a classifier makes *n* × *m* binary decisions. A two-by-two contingency table can be computed for each class

	truly YES	truly NO
system YES	a	b
system NO	c	d

Classification Performance Measures

- Recall = a/(a+c) where a + c > 0 (o.w. undefined).
  - Did we find all of those that belonged in the class?
- Precision = a/(a+b) where a+b>0 (o.w. undefined).
  - Of the times we predicted it was “in class”, how often are we correct?
- Accuracy = (a + d) / n
  - When one classes is overwhelmingly in the majority, this may not paint an accurate picture.
- Others: miss, false alarm (fallout), error, F-measure, area under PR ROC curve, break-even point, ...

Global Performance Measures

- Global Performance Measures

Category set $C = \{C_1,\ldots,C_m\}$		Manual Labels	
		C+	C-
Classifier Judgments	C+	$TP = \sum_{i=1}^{m} \mathrm{TP_i}$	$FP = \sum_{i=1}^{m} \mathrm{FP_i}$
	C-	$FN = \sum_{i=1}^{m} \mathrm{FN_i}$	$TN = \sum_{i=1}^{m} \mathrm{TN_i}$

Local Performance Measures in TC

- Local Performance Measures for Category C_i

Category C_i		Manual Labels	
		C+	C-
Classifier Judgments	C+	TP_i	FP_i
	C-	FN_i	TN_i

$$\Pr_i = \frac{TP_i}{TP_i + FP_i}$$

$$\mathrm{Re}_i = \frac{TP_i}{TP_i + FN_i}$$

$$F_{1i} = \frac{2\,\mathrm{Re}_i \Pr_i}{\mathrm{Re}_i + \Pr_i}$$

- *Precision*, *Recall* and *F1*

Summary Performance Measures

- Micro-averaging

$$\Pr^{u} = \frac{TP}{TP + FP}, \ \mathrm{Re}^{u} = \frac{TP}{TP + FN}, \quad F_1^{\mu} = \frac{2TP}{FP + FN + 2TP}.$$

- Macro-averaging

$$\Pr^{M} = \frac{\sum_{i=1}^{m} \Pr_i}{m}, \qquad \mathrm{Re}^{M} = \frac{\sum_{i=1}^{m} \mathrm{Re}_i}{m},$$

$$F_1^{M} = \frac{2\,\mathrm{Re}^{M}\,\Pr^{M}}{\mathrm{Re}^{M} + \Pr^{M}} = \frac{2\sum_{i=1}^{m} \mathrm{Re}_i \sum_{i=1}^{m} \Pr_i}{m(\sum_{i=1}^{m} \Pr_i + \sum_{i=1}^{m} \mathrm{Re}_i)}.$$

Summary of Performance Measure

$$\left.\begin{aligned} P &= \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} (TP_i + FP_i)} \\[1em] R &= \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} (TP_i + FN_i)} \end{aligned}\right\} \text{microaveraging} \qquad \left.\begin{aligned} P &= \frac{\sum_{i=1}^{|C|} P_i}{|C|} \\[1em] R &= \frac{\sum_{i=1}^{|C|} R_i}{|C|} \end{aligned}\right\} \text{macroaveraging}$$

- These two methods can give different results
- It is essential to make clear which method one uses when reporting P and R values for classification

Hold-out Sets (Validation Data)

- Estimating our performance on data we used in training is likely to give us a very skewed estimate of the final system's performance. As a result, if we have a set of labeled data, *D*, we typically split it into a training set,D_{train}, and a *hold-out* set, D_{test}

- D_{train} is the only data given to the classifier for training. D_{test} can then be used to estimate performance independently. Once performance estimates are used to choose the best classifier, the final classifier is usually trained over all of *D* before deployment (more data generally means better performance – so our estimate was pessimistic)

Empirically Tuning Parameters

- When parameters need to be empirically tuned as a part of training (e.g. choosing *k*), the performance of each possible choice needs to be estimated. For the same reasons as above, the classifier cannot simply check the performance on D_{train} to estimate future performance. Therefore D_{train} is usually subdivided into a portion used to train and another portion used for picking optimal parameters (usually referred to as the *validation* set)

- After setting the parameters, the classifier trains over all of D_{train} before returning to the function that will evaluate its performance over D_{test}

Approaches to Automated Text Categorization

- Regression based on Least Squares Fit (1991)
- Nearest Neighbor Classification (1992)
- Bayesian Probabilistic Models (1992)
- Symbolic Rule Induction (1994)
- Neural Networks (1995)
- Rocchio approach (traditional IR, 1996)
- Support Vector Machines (1997)
- Boosting or Bagging (1997)
- Hierarchical Language Modeling (1998)
- First-Order-Logic Rule Induction (1999)
- Maximum Entropy (1999)
- Hidden Markov Models (1999)
- Error-Correcting Output Coding (1999)
- Maximal Figure-of-Merit Learning (2003)

…

Classification Types

- Single label vs. multi-label
  - Exactly 1 category assigned to each document vs. 0 to |C|
- Binary vs. multi-way classification
  - Binary: a special case of single label, $d_j \in D$ is assigned either to c_i or to its complement (e.g.spam – non spam)
- Document-pivoted (DPC) vs. category pivoted (CPC)
  - Given a $d_j \in D$, we want to find all the $c_i \in C$ under which it should be classified (document-pivoted)
    - DPC is suitable when documents become available at different moments in time, e.g. filtering e-mail
  - Given a $c_i \in C$, we want to find all the $d_j \in D$ under that should be classified under it (category-pivoted)
    - CPC is suitable when new categories are likely to be be added to *C*

Related Work on Classifier Design

- Decision Tree: available tools, C4.5, CART, ID3

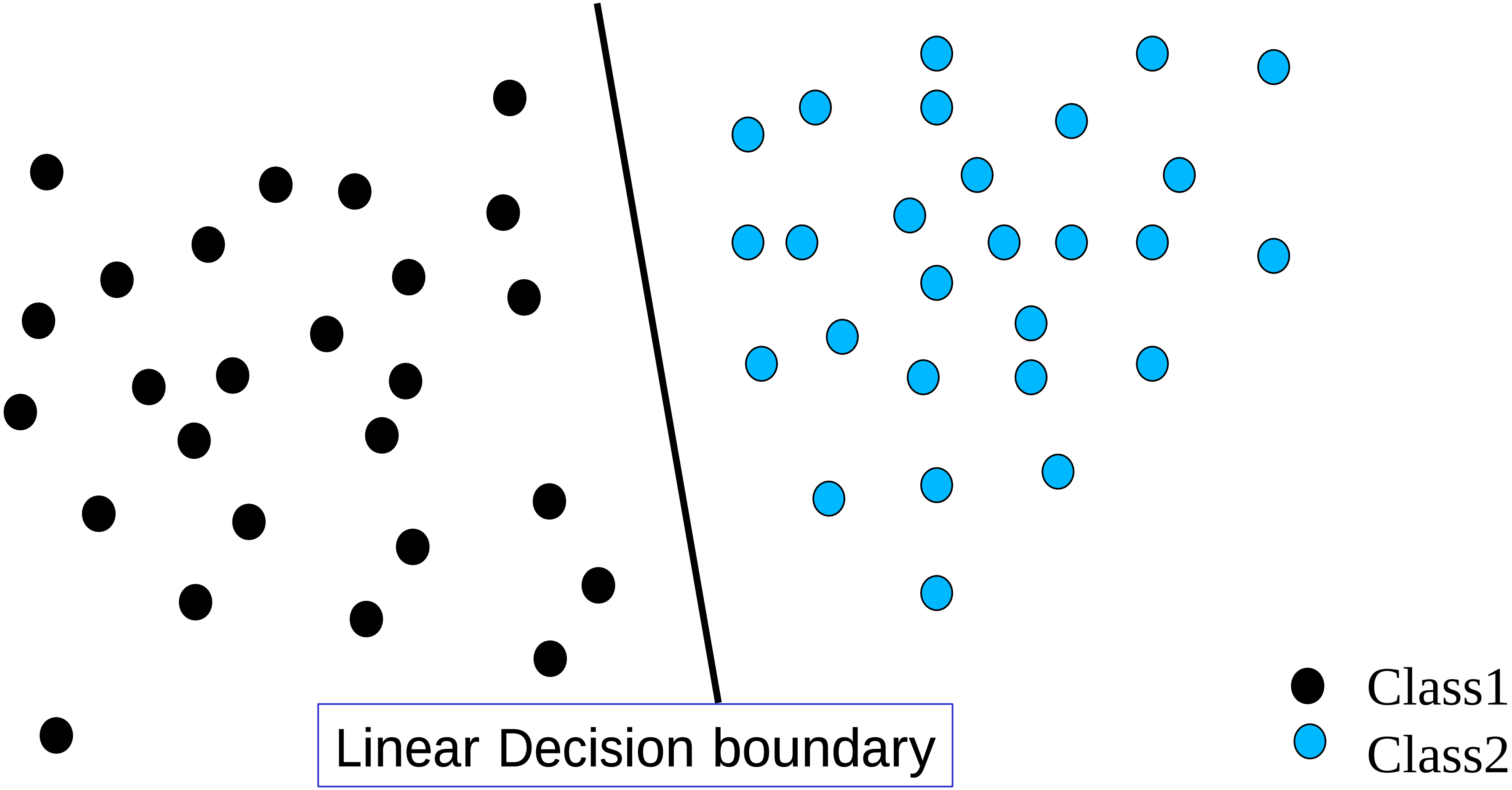
Linear discriminative function: $$f\left(X,W\right)=\sum_{i=1}^{D}w_ix_i-w_0$$

- *K*-Nearest Neighbor (*k*NN)
- Naïve Bayes: simple distributions for each class
- Support Vector Machine (SVM)
- Linear Discriminative Function (LDF)
- Artificial Neural Networks (ANN)
- Tree Classifiers (CART)
- Semantic Perceptron Net (SPN)
- Others: HMM, kernels, Discriminative Training

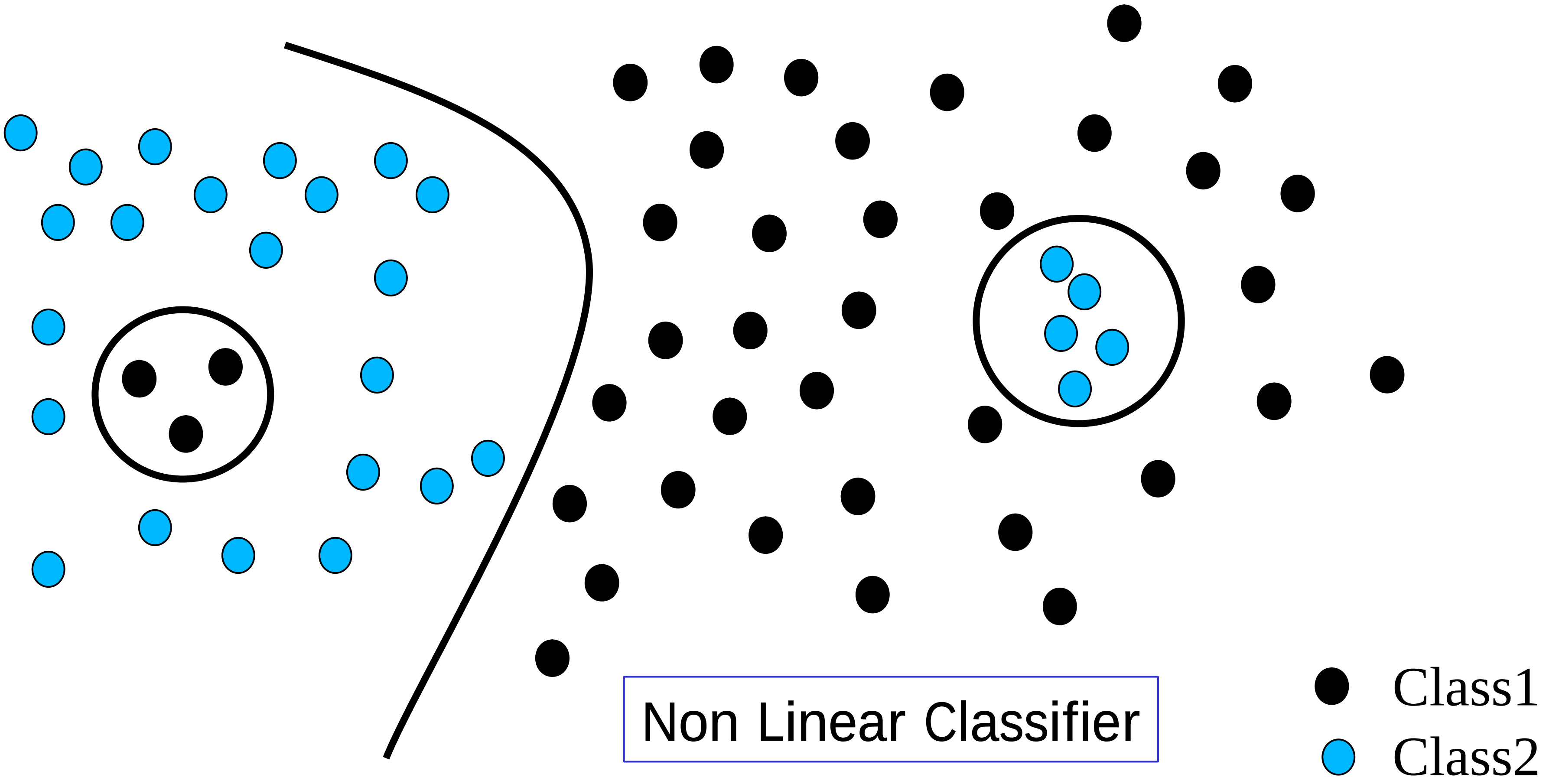
Linear vs. Nonlinear Classifiers

- Linear classifiers if
  - all data points can be correctly classified by a linear decision boundary
  - simpler, less parameters
- Non-linear otherwise
  - more accurate
  - more complicated, more parameters

Linear Case – An Example

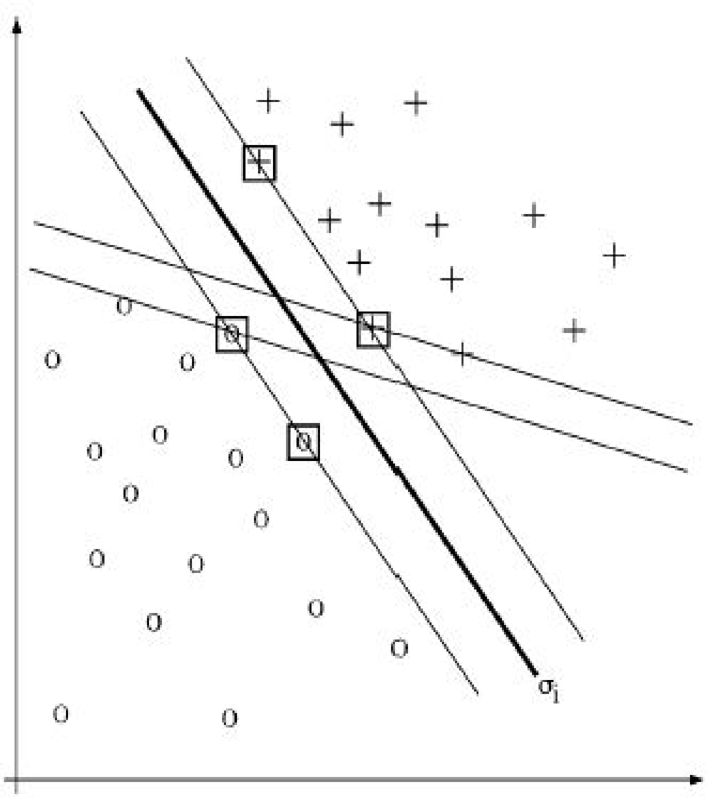


Nonlinear Case – An Example



Linear: Support Vector Machines (SVM)

- Find the hyperplane that maximizes the margin between negative and positive training examples
- Lines represent decision surfaces
- Decision surface σ_1 is the best possible one
  - middle element of the widest possible set of parallel decision surfaces
  - min. distance to any training example is maximum
  - Small boxes indicate the support vectors, the set of training examples that are used in the decision



From (Sebastiani, 2002)

Supporting Vector Machines

- Strengths
  - very effective classification
  - can scale up to data of high dimensionality
  - dimensionality reduction is normally not needed
- Weaknesses
  - can be computationally expensive, but efficient algorithms have been proposed

Key Components of Nearest Neighbor

- “Similar” item: We need a functional definition of “similarity” if we want to apply this automatically.

- How many neighbors do we consider?

- Does each neighbor get the same weight?

- All categories in neighborhood? Most frequent only? How do we make the final decision?

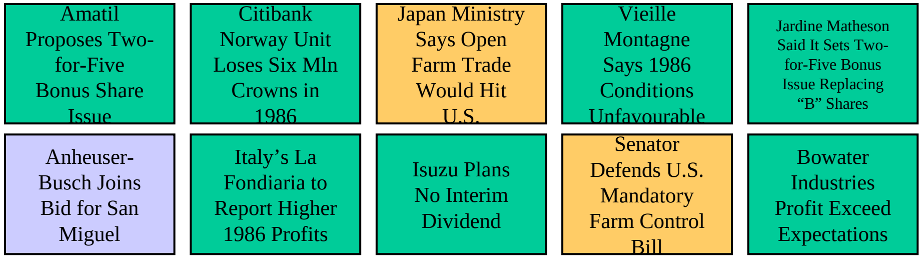
Nearest Neighbor Classification

- *Instance-Based Learning, Lazy Learning*
  - well-known approach to pattern recognition
  - initially by Fix and Hodges (1951)
  - theoretical error bound analysis by Duda & Hart (1957)
  - applied to text categorization in early 90's
  - strong baseline in benchmark evaluations
  - among top-performing methods in TC evaluations
  - scalable to large TC applications

1-Nearest Neighbor

- Looking back at our example
  - Did anyone try to find the most similar labeled item and then just guess the same color?
  - This is 1-Nearest Neighbor

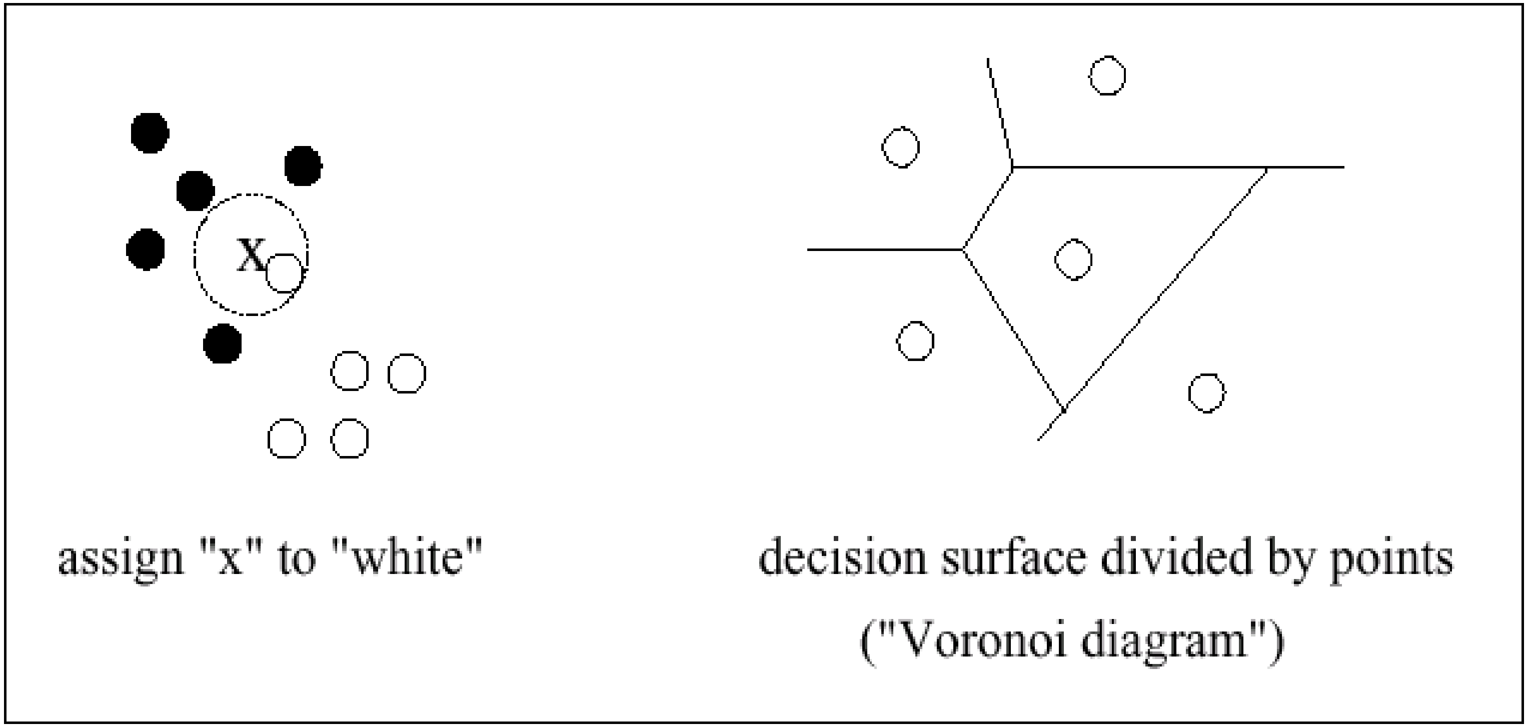
Senate Panel Studies Loan Rate, Set Aside Plans



🟩 Amatil Proposes Two-for-Five Bonus Share Issue	🟩 Citibank Norway Unit Loses Six Mln Crowns in 1986	🟧 Japan Ministry Says Open Farm Trade Would Hit U.S.	🟩 Vieille Montagne Says 1986 Conditions Unfavourable	🟩 Jardine Matheson Said It Sets Two-for-Five Bonus Issue Replacing “B” Shares
🟪 Anheuser-Busch Joins Bid for San Miguel	🟩 Italy’s La Fondiaria to Report Higher 1986 Profits	🟩 Isuzu Plans No Interim Dividend	🟧 Senator Defends U.S. Mandatory Farm Control Bill	🟩 Bowater Industries Profit Exceed Expectations

1-Nearest Neighbor (Graphically)

1-NN: assign "x" (new point) to the class of it nearest neighbor

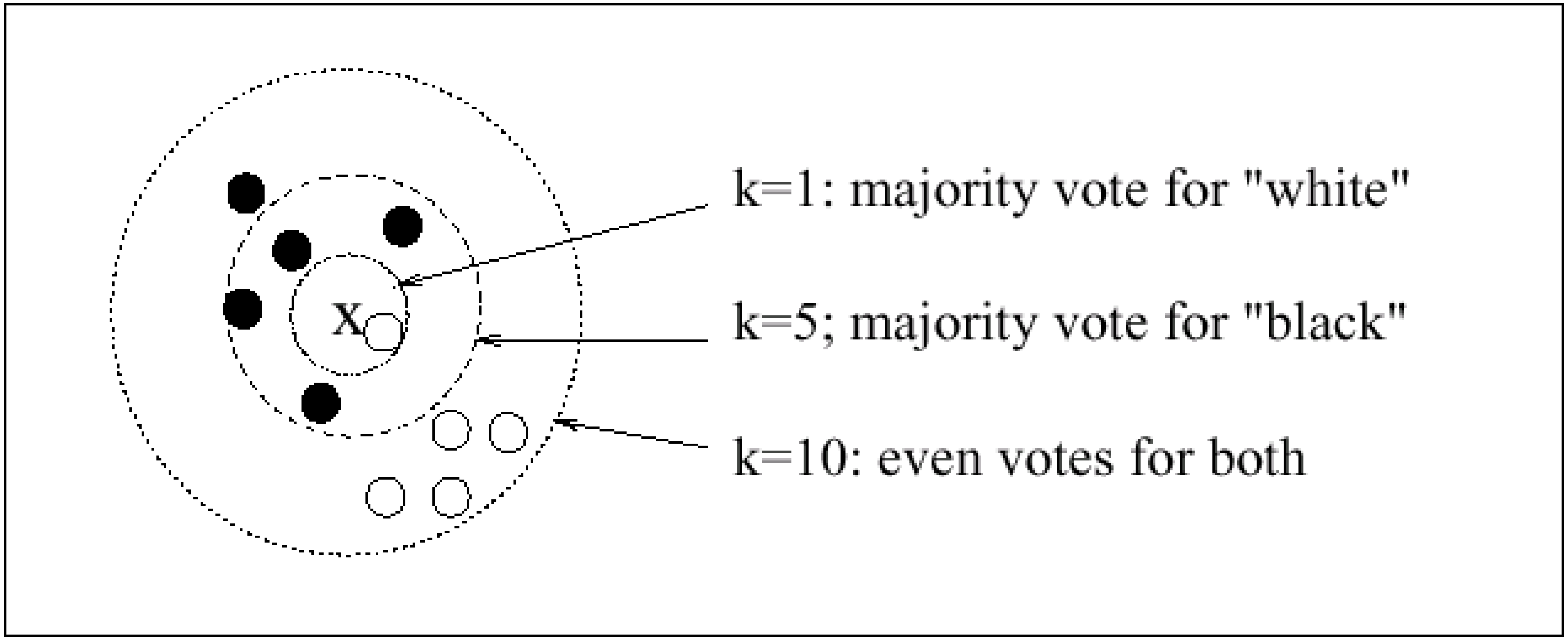


assign "x" to "white"

decision surface divided by points ("Voronoi diagram")

K-Nearest Neighbor: *Majority* Voting Scheme

K-Nearest Neighbor using a *majority* voting scheme



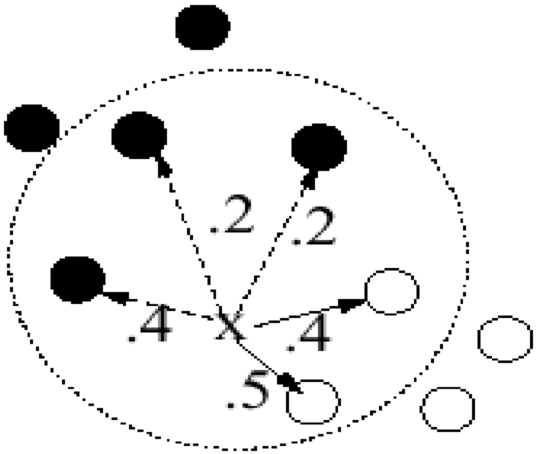
k=1: majority vote for "white"

k=5; majority vote for "black"

k=10: even votes for both

K-NN : 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.

Category Scoring for Weighted-Sum

- The score for a category is the sum of the similarity scores between the point to be classified and all of its k-neighbors that belong to the given category.

- To restate: $$score(c \mid x) = \sum_{d \in kNN\ of\ x} sim(x,d)\, I(d,c)$$

  where *x* is the new point; *c* is a class (*e.g. black* or *white);*
  d is a classified point among the k-nearest neighbors of *x*;
  sim(x,d) is the similarity between *x* and *d;*
  I(d,c) = 1 iff point *d* belongs to class *c*;
  I(d,c) = 0 otherwise.

The kth Nearest Neighbor Decision Rule (Fix and Hodges, 1951)

- Define a metric to measure “closeness” between any two points
- Fix *k* (empirically chosen)
- Given a new point *x* and a training set of classified points
  - Find the *k* nearest neighbors (kNN) to *x* in the training set
  - Classify *x* as class *y* if more of the nearest neighbors are in class *y* than in any other classes (*majority vote*)

kNN for Text Categorization (Yang, SIGIR-1994)

- Represent documents as points (vectors)
- Define a similarity measure for pair-wise documents
- Tune parameter *k* for optimizing classification effectiveness
- Choose a voting scheme (e.g., weighted sum) for scoring categories
- Threshold on the scores for classification decisions

Thresholding for Classification Decisions

- Alternative thresholding strategies:
  - Rcut: For each document to be categorized, rank candidate categories by score, and assign YES to the top-*m* categories (where *m* is some fixed number)
  - Pcut: Applies only when we have a whole batch of documents to be categorized. Make the category assignments proportional to the category distribution in the training set (i.e. if 1/4th of the training documents were in the category “Coffee Maker” then we will assign 1/4th of the documents in this batch to the “Coffee Maker” category)
  - Scut: For each category, choose a threshold score (empirically). Any document with a category score that surpasses its respective threshold will be predicted to be a member of that category

Key Components (Revisited)

- Functional definition of “similarity”
  - e.g. cos, Euclidean, kernel functions, ...
- How many neighbors do we consider?
  - Value of *k* determined *empirically* (see methodology section)
- Does each neighbor get the same weight?
  - Weighted-sum or not
- All categories in neighborhood? Most frequent only? How do we make the final decision?
  - Rcut, Pcut, or Scut

Pros of kNN

- Simple and effective (among top-5 in benchmark evaluations)
  - Non-linear classifier (vs linear)
  - Local estimation (vs global)
  - Non-parametric (very few assumptions about data)
  - Reasonable similarity measures (borrowed from IR)
- Computation (time & space) linear to the size of training data
- Low cost for frequent re-training, i.e., when categories and training documents need to be updated (common in Web environment and e-commerce applications)

Cons of kNN:

- Online response is typically slower than *eager learning* algorithms
  - Trade-off between off-line training cost and online search cost
- Scores are not normalized (probabilities)
  - Comparing directly to and combining with scores of other classifiers is an open problem
- Output not good in explaining why a category is relevant
  - Compared to DTree, for example (take this with a grain of salt)

Bayesian Methods

- Learning and classification methods based on probability theory
- Bayes theorem plays a critical role in probabilistic learning and classification
- Build 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 once more

$$P(C,X) = P(C \mid X)P(X) = P(X \mid C)P(C)$$

$$\boxed{P(C \mid X) = \frac{P(X \mid C)P(C)}{P(X)}}$$

Maximum a posteriori Decision Rule

$$h_{MAP} \equiv \underset{h \in H}{\operatorname{argmax}} P(h \mid D)$$

$$= \underset{h \in H}{\operatorname{argmax}} \frac{P(D \mid h) P(h)}{P(D)}$$

$$= \underset{h \in H}{\operatorname{argmax}} P(D \mid h) P(h)$$

Maximum likelihood Hypothesis

If all hypotheses are a priori equally likely, we only need to consider the *P*(*D*|*h*) term:

$$h_{ML} \equiv \underset{h \in H}{\operatorname{argmax}} P(D \mid h)$$

Naive Bayes Classifiers

Task: Classify a new instance *D* based on a tuple of attribute values $D = \langle x_1, x_2, \ldots, x_n \rangle$ into one of the classes $c_j \in C$

$$c_{MAP} = \underset{c_j \in C}{\operatorname{argmax}} P(c_j \mid x_1, x_2, \ldots, x_n)$$

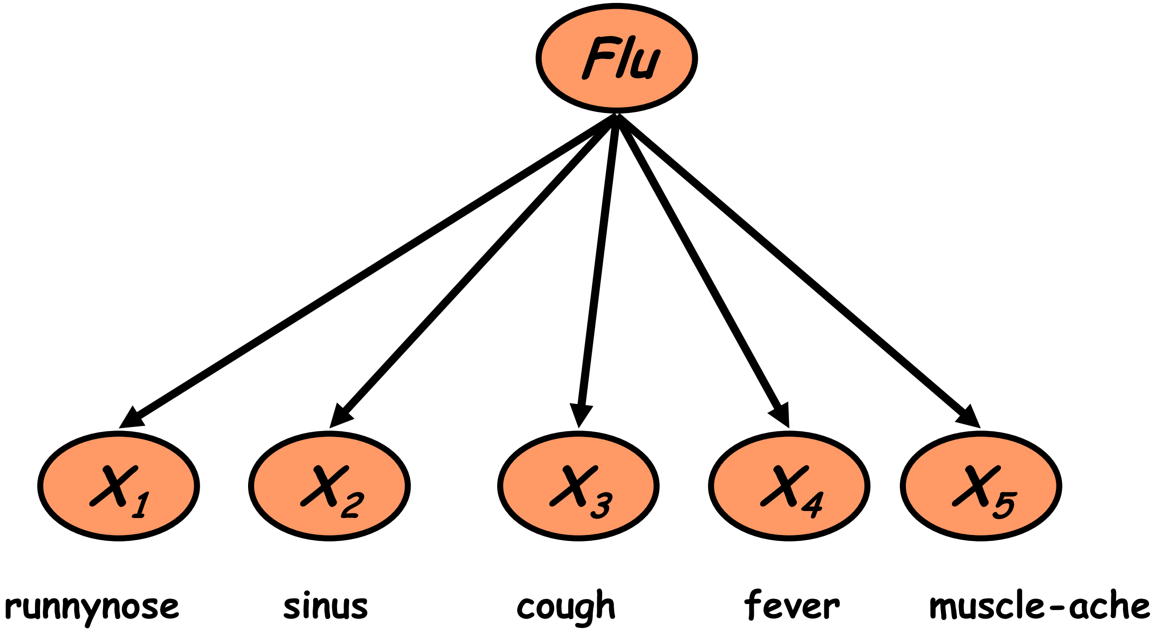
$$= \underset{c_j \in C}{\operatorname{argmax}} \frac{P(x_1, x_2, \ldots, x_n \mid c_j) P(c_j)}{P(x_1, x_2, \ldots, x_n)}$$

$$= \underset{c_j \in C}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_n \mid c_j) P(c_j)$$

Naïve Bayes Classifier: Assumption

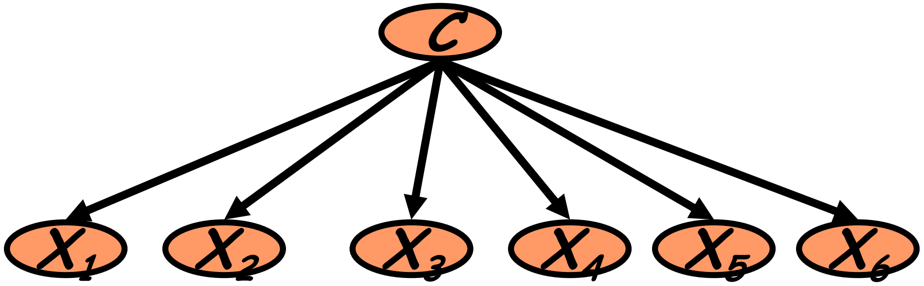
- $P(c_j)$: Can be estimated from the frequency of classes in the training examples
- $P(x_1,x_2,\ldots,x_n|c_j)$: $O(|X|^n \bullet |C|)$ parameters, Could only be estimated if a very, very large number of training examples was available
- Naïve Bayes Conditional Independence Assumption:
  - Assume that the probability of observing the conjunction of attributes is equal to the product of the individual probabilities $P(x_i|c_j)$.

The Naïve Bayes Classifier



$$P(X_1,\ldots,X_5 \mid C) = P(X_1 \mid C) \bullet P(X_2 \mid C) \bullet \cdots \bullet P(X_5 \mid C)$$

Learning the Model



- First attempt: maximum likelihood estimates
  - simply use the frequencies in the data

$$\hat{P}(c_j) = \frac{N(C = c_j)}{N}$$

$$\hat{P}(x_i \mid c_j) = \frac{N(X_i = x_i, C = c_j)}{N(C = c_j)}$$

Smoothing to Avoid Overfitting

$$\hat{P}(x_i \mid c_j) = \frac{N(X_i = x_i, C = c_j) + 1}{N(C = c_j) + k}$$

# of values of X_i

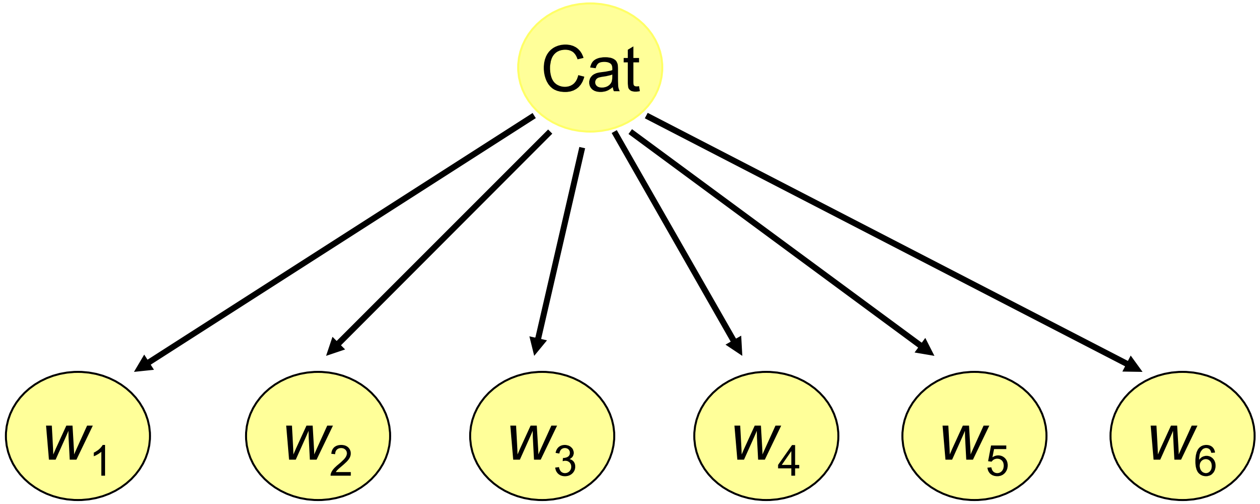
- Somewhat more subtle version

overall fraction in data where X_i=$x_{i,k}$

$$\hat{P}(x_{i,k} \mid c_j) = \frac{N(X_i = x_{i,k}, C = c_j) + mp_{i,k}}{N(C = c_j) + m}$$

extent of “smoothing”

Class Conditional Multinomial NB



- Effectively, the probability of each class is done as a class-specific unigram language model

Basic NB Classifiers to Classify Text

- Attributes are text positions, values are words

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} P(c_j) \prod_i P(x_i \mid c_j)$$

$$= \underset{c_j \in C}{\operatorname{argmax}} P(c_j) P(x_1 = \text{"our"} \mid c_j) \cdots P(x_n = \text{"text"} \mid c_j)$$

- Still too many possibilities
- Assume that classification is *independent* of the positions of the words
  - Use same parameters for each position
  - Result is bag of words model (over tokens not types)

Naïve Bayes: Learning

- From training corpus, extract *Vocabulary*
- Calculate required $P(c_j)$ and $P(x_k \mid c_j)$ terms
  - For each c_j in C do
    - $docs_j \leftarrow$ subset of documents for which the target class is c_j
    - $P(c_j) \leftarrow \dfrac{|docs_j|}{|\text{total \# documents}|}$

$Text_j \leftarrow$ single document containing all $docs_j$

for each word x_k in *Vocabulary*

$n_k \leftarrow$ number of occurrences of x_k in $Text_j$

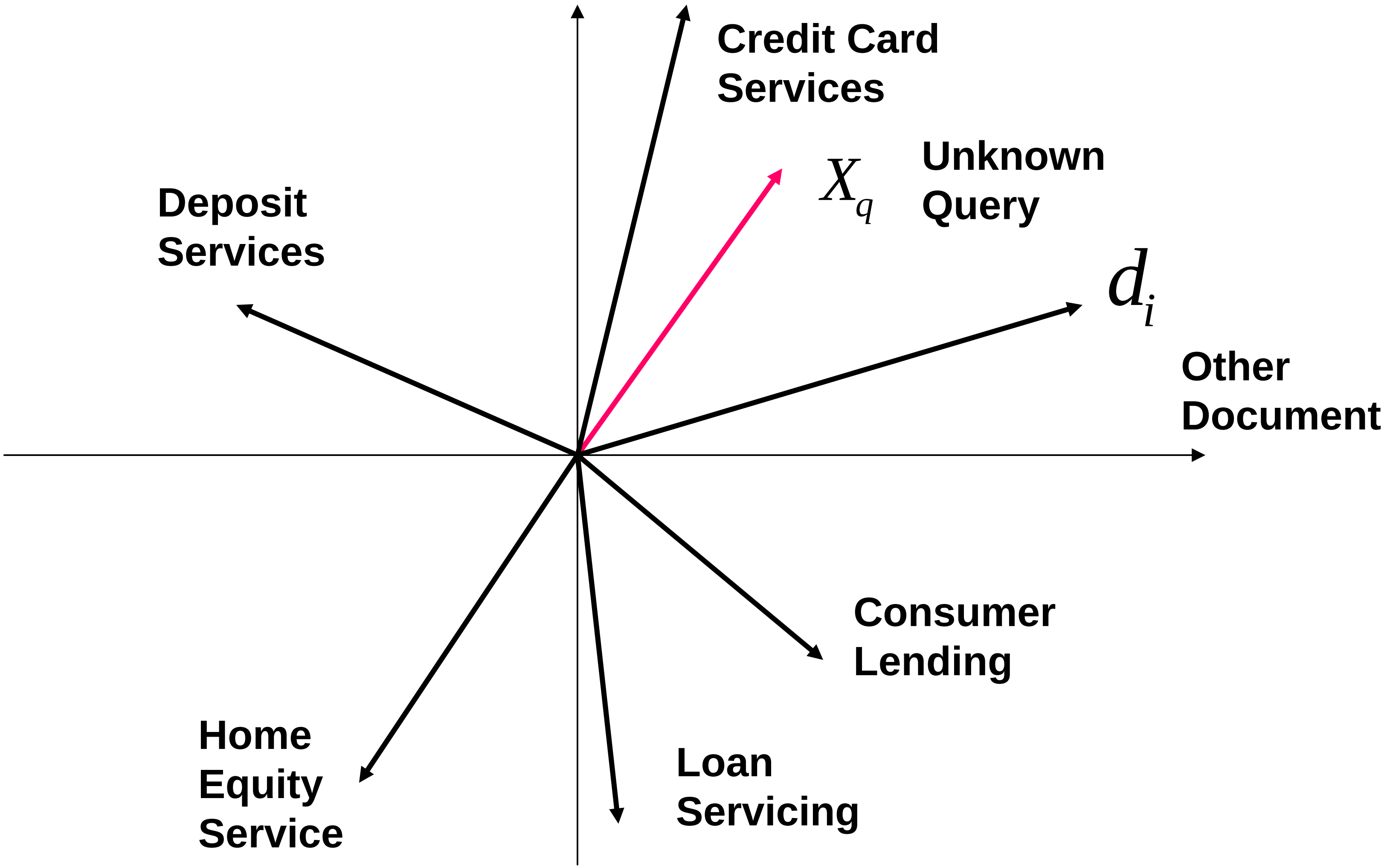
$$P(x_k \mid c_j) \leftarrow \frac{n_k + \alpha}{n + \alpha\,|Vocabulary|}$$

Naïve Bayes: Classifying

- positions ← all word positions in current document which contain tokens found in *Vocabulary*
- Return c_{NB}, where

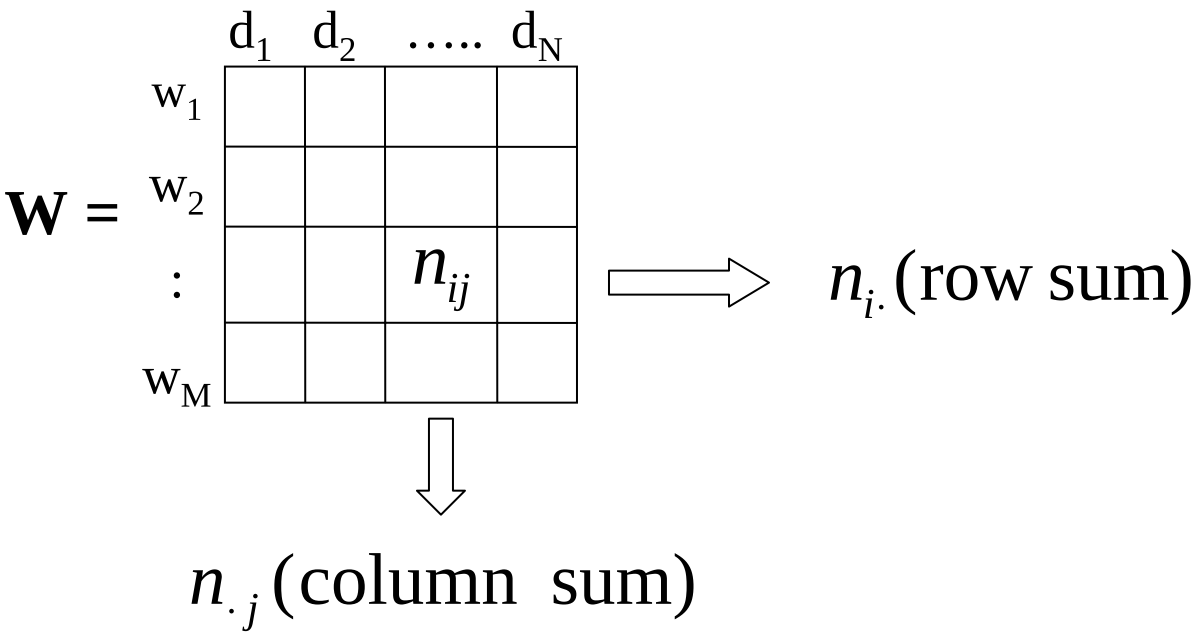
$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} P(c_j) \prod_{i \in positions} P(x_i \mid c_j)$$

Vector Space Representation



Word-Document Co-Occurrence

- Given - *N* documents, vocabulary size *M*
- Generate a word-documents co-occurrence matrix **W**



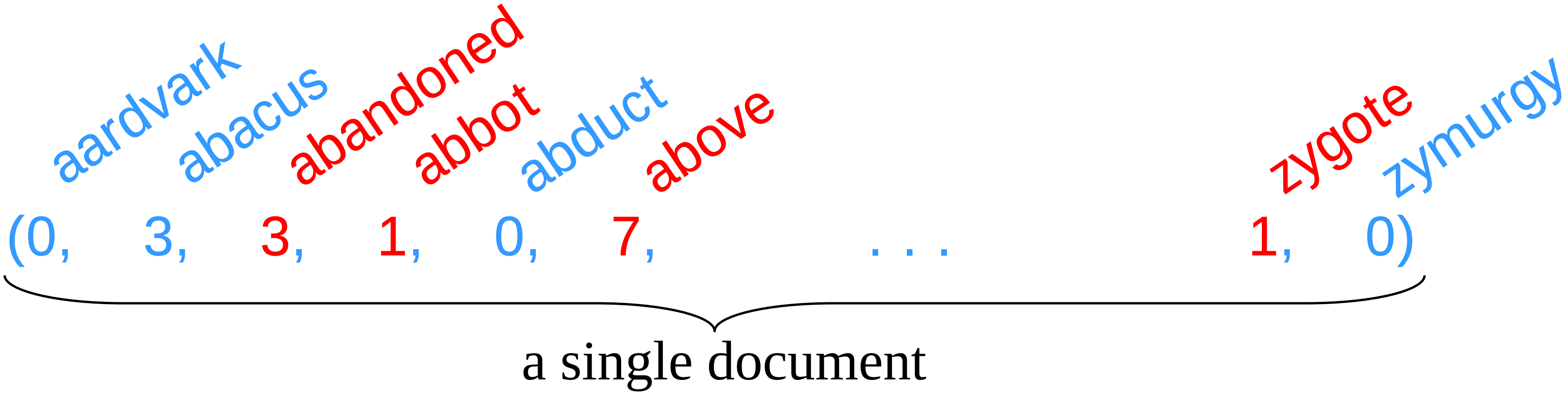
$$\mathbf{W} = \begin{array}{c|c|c|c|c|} & d_1 & d_2 & \ldots\ldots & d_N \\ \hline w_1 & & & & \\ \hline w_2 & & & & \\ \hline : & & & n_{ij} & \\ \hline w_M & & & & \\ \hline \end{array}$$

$n_{i\cdot}$ (row sum)

$n_{\cdot j}$ (column sum)

LSA Count in the Column Vector

- A trick from Information Retrieval
  - Each **document** (paragraph or sentence) in the training document corpus is a length-*M* vector



LSA Mathematical Framework

- LSA Matrix (also known as Routing Matrix) *C*

$c_{ij} = (1-\varepsilon_i) n_{ij} / n_{\cdot j}$ (scaling and normalization)

- – number of times word w_i occurs in A_j : n_{ij}
- – total number of words present in A_j : $n_{\cdot j}$ (column sum)
- – total number of w_i occurs in *A* : $n_{i\cdot}$ (row sum)
- – "indexing" power of w_i in corpus *A* : $\eta_i = 1-\varepsilon_i$
- – normalized entropy:

$$\varepsilon_i = -\frac{1}{\log N} \sum_{j=1}^{N} \frac{n_{ij}}{n_{i\cdot}} \log \frac{n_{ij}}{n_{i\cdot}} \qquad 0 \le \varepsilon_i \le 1$$

$$\begin{cases} \varepsilon_i = 0 \quad \text{if } n_{ij} = n_{i\cdot} \quad \text{maximum indexing power} \\ \varepsilon_i = 1 \quad \text{if } n_{ij} = \frac{n_{i\cdot}}{N} \quad \text{no power (equally probable)} \end{cases}$$

Semantic Similarity Measure

- To find similarity between two documents, project them in LS space
- Then calculate the cosine measure between their projection
- With this measure, various problems can be addressed e.g., natural language understanding, cognitive modeling etc.

Confidence Scoring

- Inner Product: $s(x,y)=x \bullet y^t$
- Cosine: $s(x,y)=\left. x \bullet y^t \middle/ |x||y| \right.$ or $\cos^{-1}[s(x,y)]$
- Confidence Scoring: Sigmoid function fitting

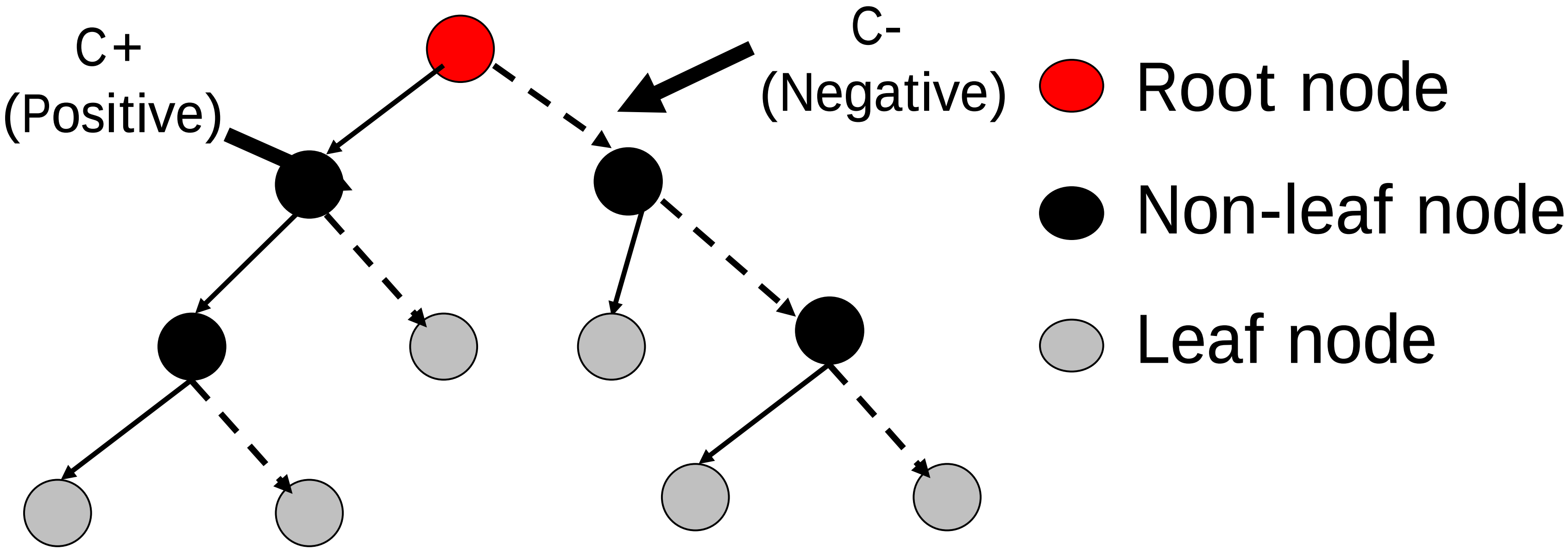
$$Conf\ (s;\alpha,\beta)=[1+e^{-(\alpha s+\beta)}]^{-1}$$

- Other Scores
  - Euclidean, Manhattan, etc.
- Generalized Scores
  - between any two vectors: $s(x,y)=f(x,y;\Gamma)$

Similarity in LSA

- The vector of a passage is the vector sum of the vectors standing for the words it contains
- Similarity of any two words or two passages is computed as the cosine between them in the semantic space:
  - Identical meaning: value of cosine = 1
  - Unrelated meaning: value of cosine = 0
  - Opposite meaning: value of cosine = -1
- Number of dimensions used is an important issue
  - Small dimensions (small singular values) represent local unique components
  - Large dimensions capture similarities and differences

A Simple Binary Tree Classifier



(*X*: feature vector, *W*: parameters of the classifier)

$$f\left(X,W\right)=\sum_{i=1}^{D} w_i x_i - w_0$$

Decision rule:

$$\begin{cases} f\left(X,W\right)\geq 0, & \text{label C}+ \\ \text{Otherwise} & \text{label C-} \end{cases}$$

Multi-Class vs. Binary Decision Rule

- Multi-class (MC) classification

$$C(X) = \arg\max_{j} g_j(X;W), \quad 1 \le j \le m$$

$$g_j(X;W) > g_{i \ne j}(X;W) \quad X \in C_j$$

- Special case: Binary classifier with LDF (*C+: positive class, C-: negative class*)

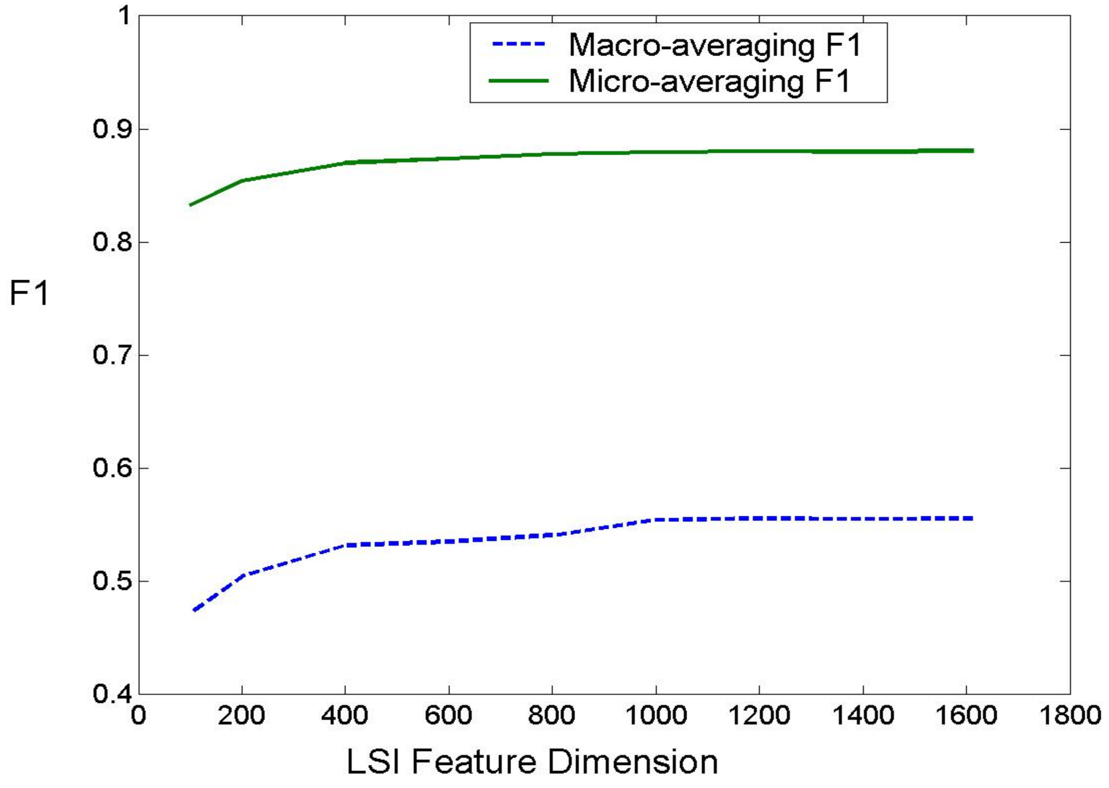
$$\begin{cases} f(W,X) \ge 0 & \text{label C+} \\ \text{Others} & \text{label C-} \end{cases}$$

➢ **Decision rule is a** ***discrete, non-differential function of the classifier parameters (need MFoM to optimize)***

Task and Experimental Setup

- *ModApte* split version of *Reuters-21578* corpus
  - – lexicon: 10118 words, remove 319 stop-words and words occurred less than 4 times
  - – corpus clean-up: remove documents which are not labeled by topics, miss topics, or are labeled by topics only occurred in training or test corpus
  - – final experiments setup: 7,770 training documents, 3,019 test documents, 90 topics
  - – some topics have little data for training or testing and with conflict labels in some cases

Performance vs. LSI Feature Dimension



- MFoM Classifier performs better than the best SVM

Experimental Results
- Properties of MFoM Learning

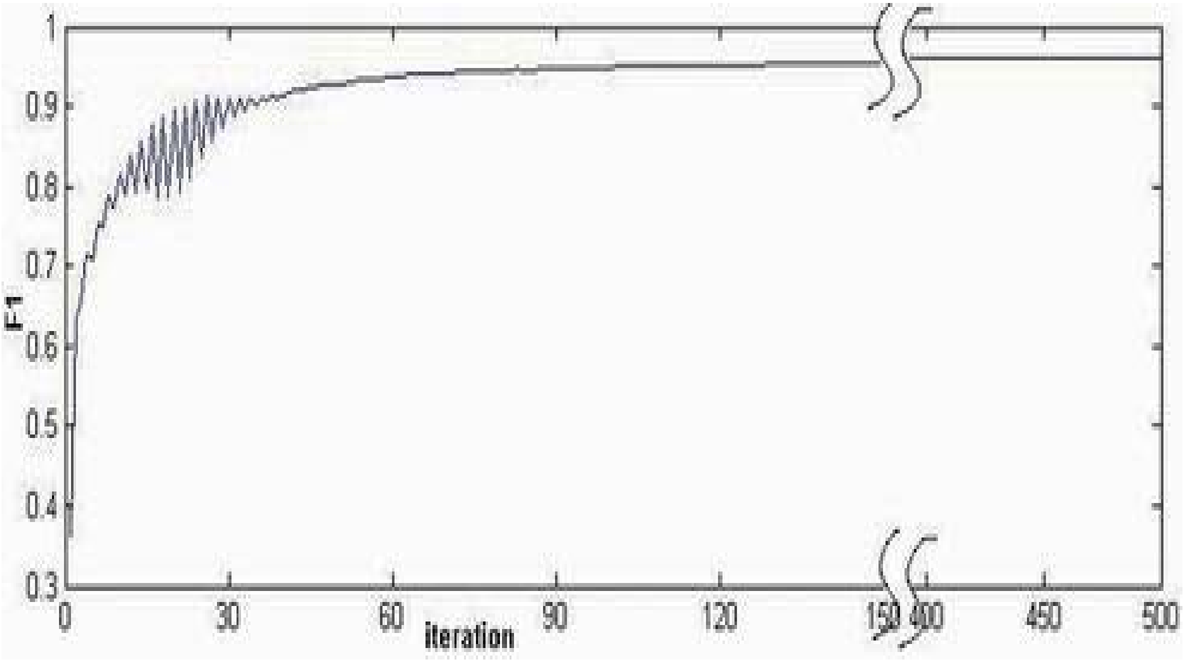
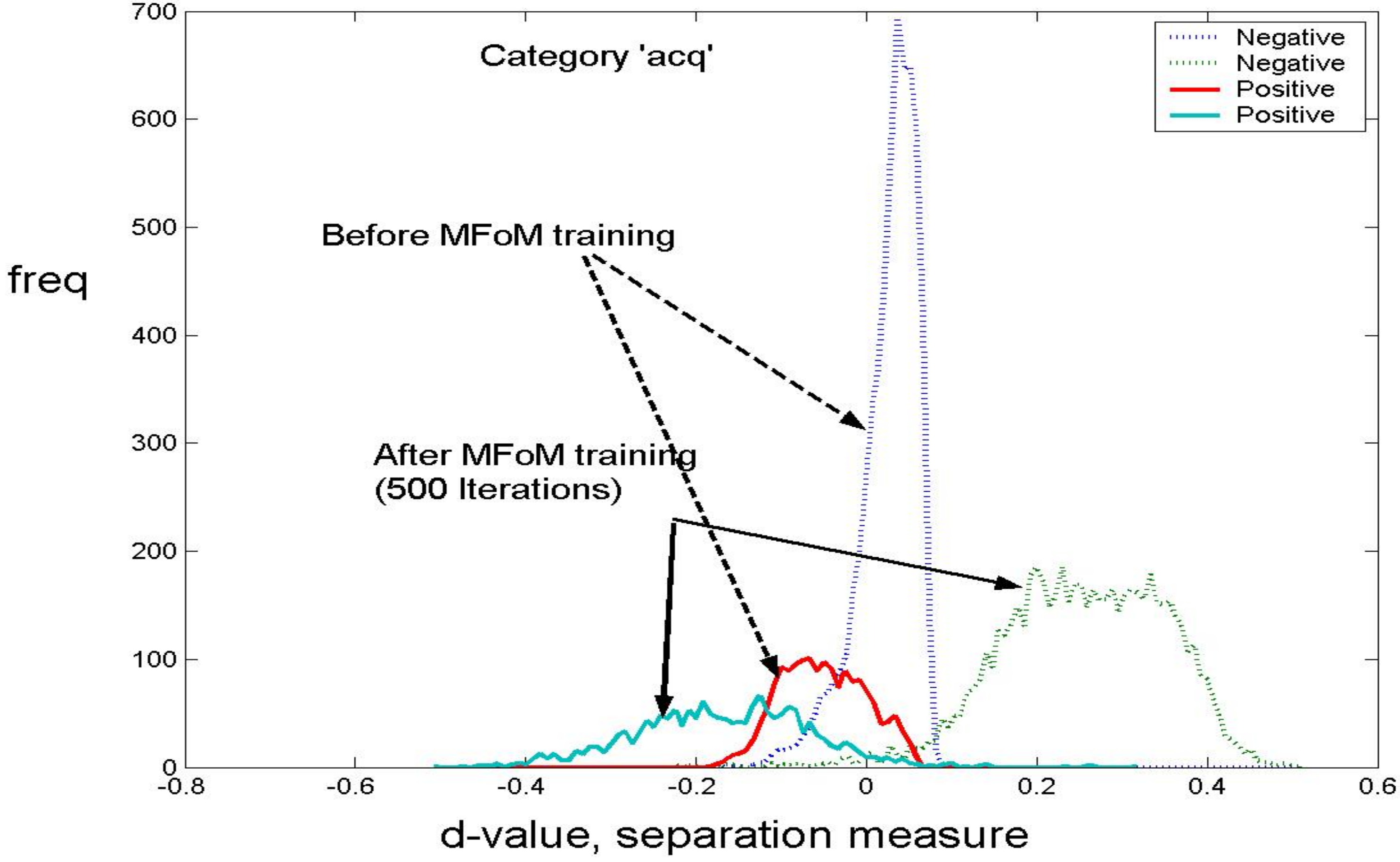


Figure 3. GPD convergence for category ‘acq’ (feature dimension: 400, X-axis: number of the iteration, Y-axis: F_1 measure for the positive class over training samples)

Separation before and after MFoM (Gao, Wu and Lee, *SIGIR-2003*)



Performance Comparison (SIGIR2003)

	k-NN	SVM	Binary F_1-MFoM
micR	0.834	0.812	0.857
micP	0.881	0.914	0.914
micF_1	0.857	0.860	0.884
macF_1	0.524	0.525	0.556

Binary vs. MC TC (ICML04)

Category	# of Training instances	Binary MFoM	MC MFoM
Income	9	0.429	0.600
Oat	8	0.167	0.500
Platinum	5	0.286	0.833
Potato	3	0.333	0.750
Sun-meal	1	0.000	0.667

F_1 -based comparison:

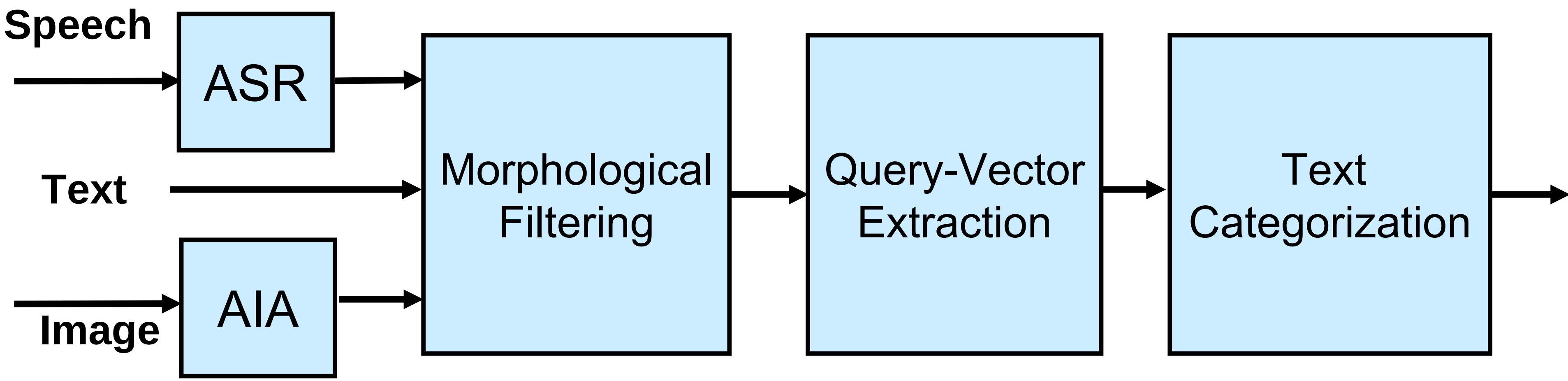
Multi-Class MFoM works much better for small training sizes

From Text to Multimedia Documents

- Property of raw multimedia patterns
  - Mostly fuzzy low-level signal representations
  - Hard to locate segmentation and object boundaries
- Definition of common sets of fundamental units
  - No obvious fundamental alphabets and words
  - Precision and coverage of multimedia tokenization
- Extraction of multimedia document feature vectors
  - Dimensionality, discrimination ability and trainability
- What are the missing links?
  - Shannon's information theory perspective (1951)
  - Finding acoustic, audio, visual "alphabets" and "words"

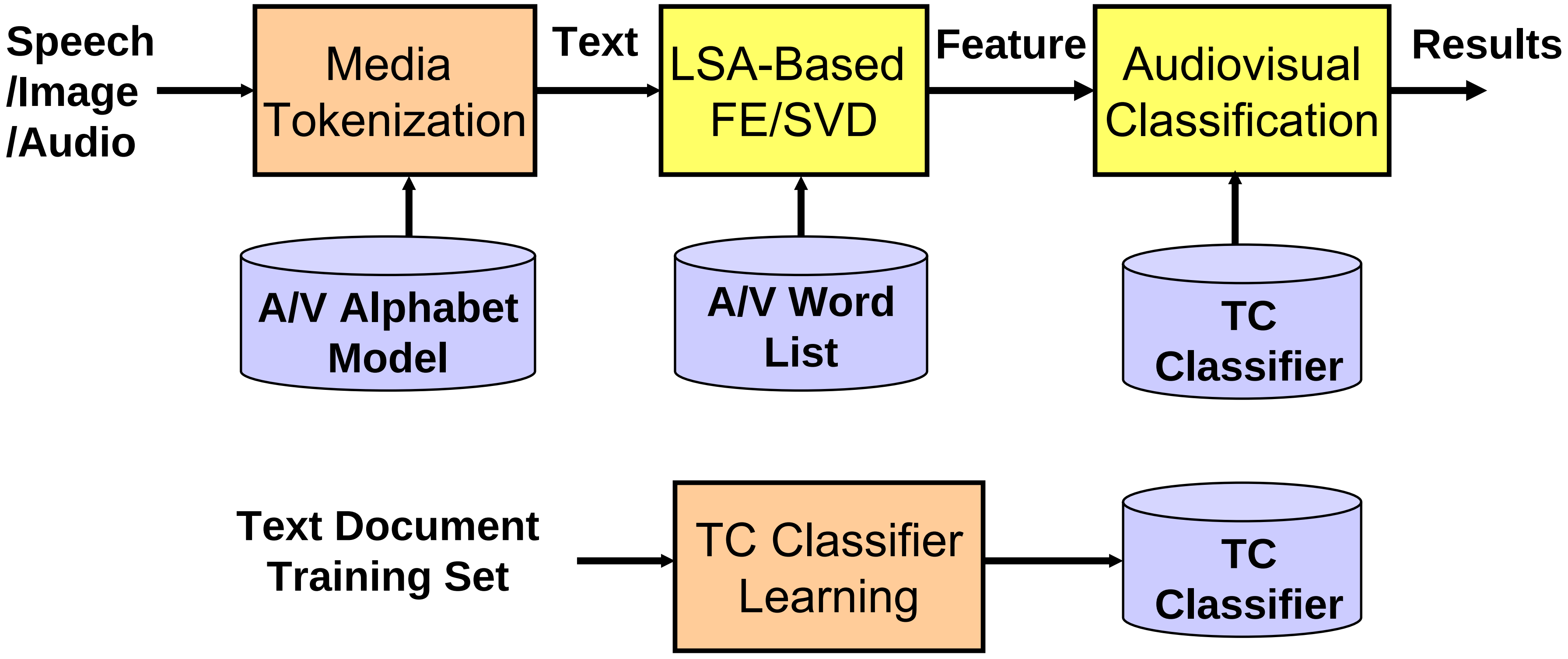
Event Representation & Topic Classification

- Video: speech, audio, image, text, and others



ASR: Automatic Speech Recognition
AIA: Automatic Image Annotation

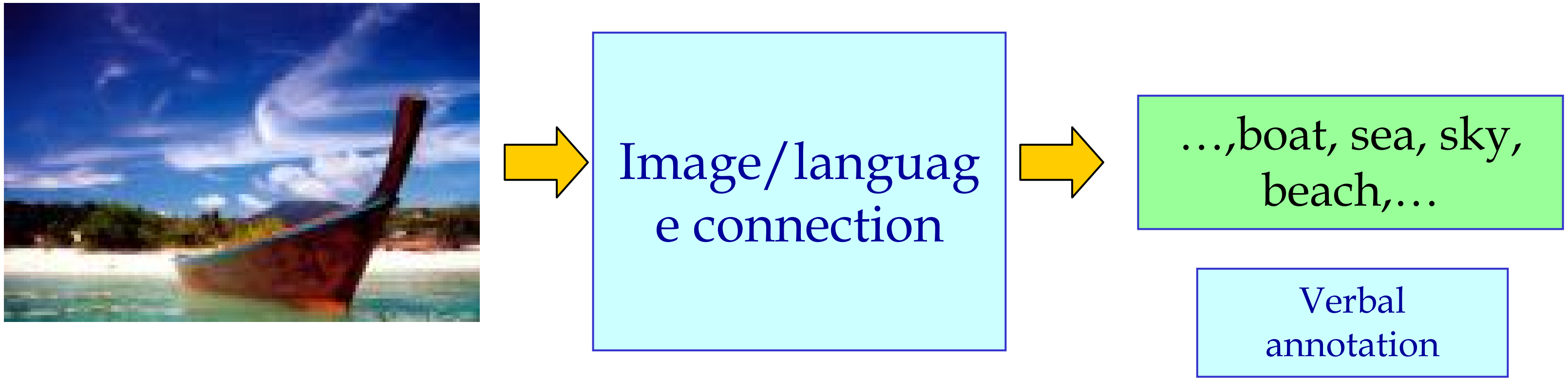
Common Technology Thread: DSP, Feature Extraction & Classifier Learning



First Step: Define alphabets and training alphabet models

Automatic Image Annotation (AIA)

- A process associating concepts or keywords to images describing their visual content
- AIA can be used to make queries based on image concepts (Google-style keyword search)

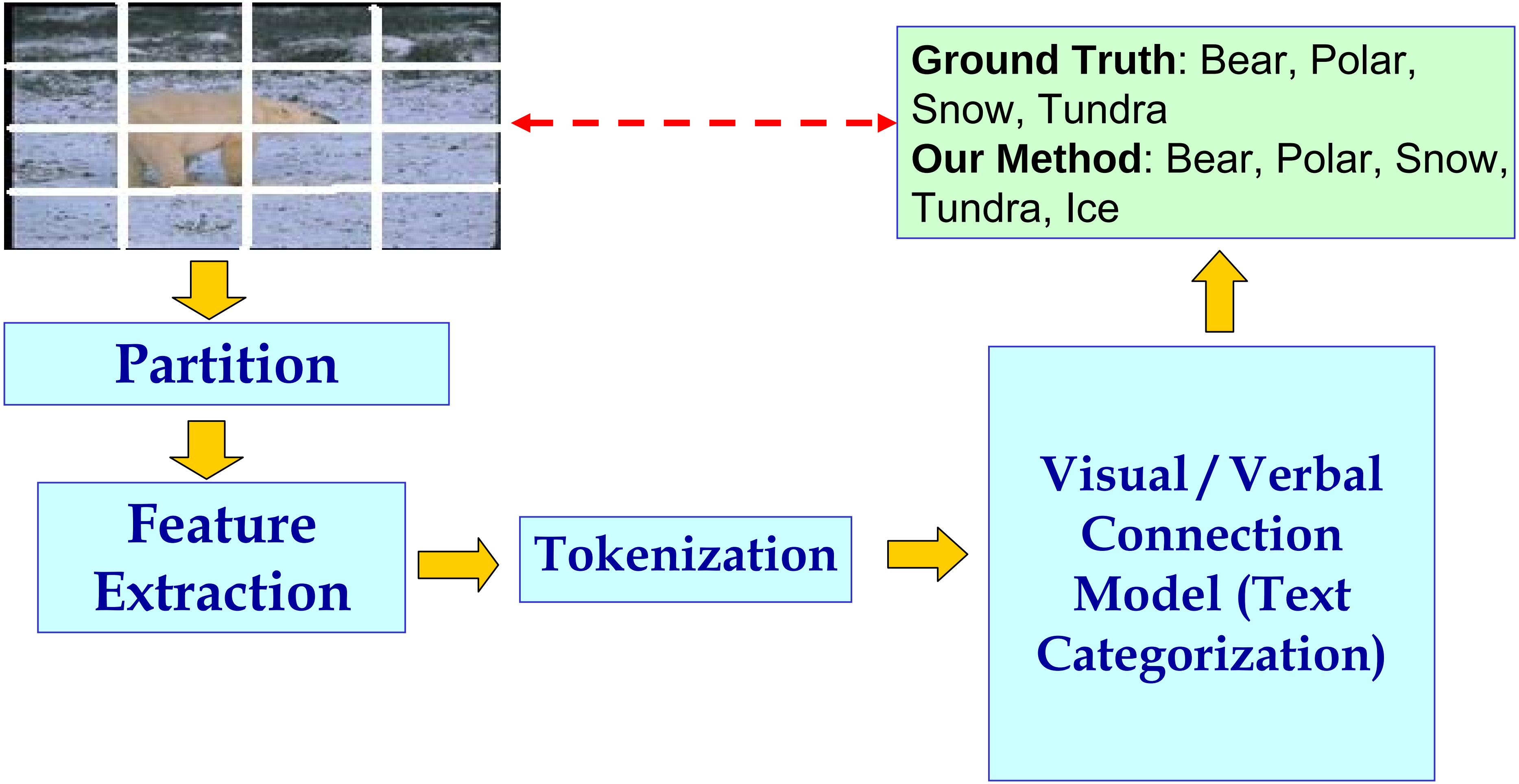


Image/language connection

…,boat, sea, sky, beach,…

Verbal annotation

Automatic Image Annotation



Ground Truth: Bear, Polar, Snow, Tundra
Our Method: Bear, Polar, Snow, Tundra, Ice

Partition

Feature Extraction

Tokenization

Visual / Verbal Connection Model (Text Categorization)

Text Representation of Images

- Given a visual lexicon, $A=\{A_1,A_2, \ldots, A_M\}$, with M visual terms, an image document can be represented by $V=\{V_1,V_2, \ldots, V_M\}$, each component being statistics of visual term occurred in the particular image document
- SVD can be applied to reduce the dimension, M
- Semantic concept modeling for image annotation
  - Semantic concept set, $C=\{C_j,\ 1\le j\le N\}$, N: total concepts. Each concept has a discriminant function, $g_j(X;\Lambda_j)$, to be trained. Multiple relevant keywords are assigned to an image X, according to the rule,

Music and Speech Connection

- Krishna and Sreenivas (2004) drew parallels between music and speech
  - Speech recognition ≈ music transcription
  - Instrument recognition ≈ speaker recognition
  - “Cocktail” separation ≈ instrument separation
  - Genre classification ≈ language classification
- Perceptual results do exist that give support to the link between music and language, but the debate is still continuing

Some references

- If you only read one article/reference:
  - Sebastiani, F. Machine learning in automated text categorization. ACM Computing Surveys, 34(1):1-47, 2002
- Worth having a look at:
  - Yang, Y. and Pedersen, J.O. A comparative study on feature selection in text categorization. In Proceedings of the 14th International Conference on Machine Learning, pages 412-420, 1997.
  - Dumais, S. and Chen, H. Hierarchical classification of web content. In Proceedings of the 23rd ACM SIGIR Conference, pages 256-263, 2000.
  - Lewis, D. D. An evaluation of phrasal and clustered representations on a text categorization task. In Proceedings of the 15th ACM SIGIR Conference, pages 37-50, 1992.

Some References (Cont.)

- Mitchell, T. Machine learning. McGraw-Hill, 1997.
- Yang, Y. and Liu, X. A re-examination of text categorization methods. In Proceedings of ACM SIGIR, 1999.
- Lewis, D.D. Evaluating and optimizing autonomous text classification systems. In Proceedings of ACM SIGIR, 1995.
- Joachims, T. Text categorization with support vector machines: learning with many relevant features. In Proceedings of 10th European Conference on Machine Learning, pages 137-142, 1998.
- Hearst, M.A. Trends and discoveries: support vector macines. In IEEE Intelligent Systems, July/August 1998, pages 18-28.
- Yang, Y., Slattery, S., Ghani, R. A study of approaches to hypertext categorization. Journal of Intelligent Information Systems, 18(2/3):219-241, 2002.
- Gao, S., Wu, W., Chua, T.-S., Lee, C.-H. "A maximal figure-of-merit learning approach to trext categorization,". *Proc. of SIGIR*, 2003.

Summary

- Today's Class
  - Text categorization
- Next Classes
  - Information retrieval
  - Labs 4-5 on PoS tagging and document clustering
  - Spring break: March 16-20, catch-up time
  - After break: IR, PCFG, probabilistic parsing
- Reading Assignments
  - Manning and Schutze, Chapters 14-16