Introduction to

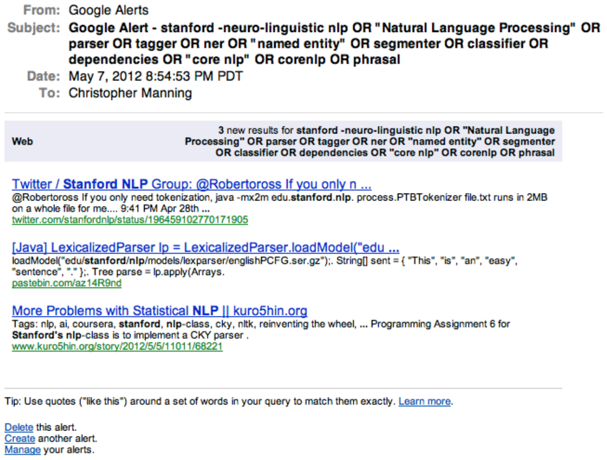
Information Retrieval

CS276: Information Retrieval and Web Search

Lecture 10: Text Classification;
The Naive Bayes algorithm

Standing queries

- The path from IR to text classification:
  - You have an information need to monitor, say:
    - Unrest in the Niger delta region
  - You want to rerun an appropriate query periodically to find new news items on this topic
  - You will be sent new documents that are found
    - I.e., it's not ranking but classification (relevant vs. not relevant)
- Such queries are called **standing queries**
  - Long used by "information professionals"
  - A modern mass instantiation is **Google Alerts**
- Standing queries are (hand-written) text



From: Google Alerts
Subject: **Google Alert - stanford -neuro-linguistic nlp OR "Natural Language Processing" OR parser OR tagger OR ner OR "named entity" OR segmenter OR classifier OR dependencies OR "core nlp" OR corenlp OR phrasal**
Date: May 7, 2012 8:54:53 PM PDT
To: Christopher Manning

Web 3 new results for **stanford -neuro-linguistic nlp OR "Natural Language Processing" OR parser OR tagger OR ner OR "named entity" OR segmenter OR classifier OR dependencies OR "core nlp" OR corenlp OR phrasal**

Twitter / **Stanford NLP** Group: @Robertoross If you only n ...
@Robertoross If you only need tokenization, java -mx2m edu.**stanford.nlp**. process.PTBTokenizer file.txt runs in 2MB on a whole file for me.... 9:41 PM Apr 28th ...
twitter.com/stanfordnlp/status/196459102770171905

[Java] LexicalizedParser lp = LexicalizedParser.loadModel("edu ...
loadModel("edu/**stanford**/**nlp**/models/lexparser/englishPCFG.ser.gz");. String[] sent = { "This", "is", "an", "easy", "sentence", "." };. Tree parse = lp.apply(Arrays.
pastebin.com/az14R9nd

More Problems with Statistical **NLP** || kuro5hin.org
Tags: nlp, ai, coursera, **stanford**, **nlp**-class, cky, nltk, reinventing the wheel, ... Programming Assignment 6 for **Stanford's nlp**-class is to implement a CKY parser .
www.kuro5hin.org/story/2012/5/5/11011/68221

Tip: Use quotes ("like this") around a set of words in your query to match them exactly. Learn more.

Delete this alert.
Create another alert.
Manage your alerts.

Spam filtering
Another text classification task

From: "" <takworlld@hotmail.com>
Subject: real estate is the only way... gem oalvgkay

Anyone can buy real estate with no money down

Stop paying rent TODAY !

There is no need to spend hundreds or even thousands for similar courses

I am 22 years old and I have already purchased 6 properties using the methods outlined in this truly INCREDIBLE ebook.

Change your life NOW !

===
Click Below to order:
http://www.wholesaledaily.com/sales/nmd.htm

Text classification

- Today:
  - Introduction to Text Classification
    - Also widely known as "text categorization"
    - Same thing
  - Naïve Bayes text classification
    - Including a little on Probabilistic Language Models

Categorization/Classification

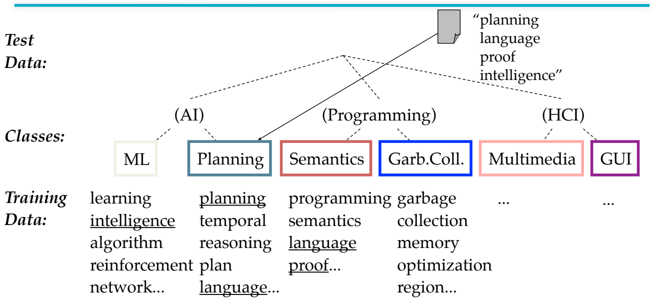
- Given:
  - A description of an instance, $d \in X$
    - X is the instance language or instance space.
      - Issue: how to represent text documents.
      - Usually some type of high-dimensional space – bag of words
  - A fixed set of classes:
    $C = \{c_1, c_2,\ldots, c_J\}$
- Determine:
  - The category of d: $\gamma(d) \in C$, where $\gamma(d)$ is a classification function whose domain is X and whose range is C.
    - We want to know how to build classification functions

Machine Learning: Supervised Classification

- Given:
  - A description of an instance, d ∈ X
    - X is the instance language or instance space.
  - A fixed set of classes:

    C = {c_1, c_2,…, c_j}
  - A training set D of labeled documents with each labeled document ⟨d,c⟩ ∈ X×C
- Determine:
  - A learning method or algorithm which will enable us to learn a classifier γ:X→C
  - For a test document d, we assign it the class γ(d) ∈ C

Document Classification


Test Data:
“planning language proof intelligence”
Classes:
(AI)
(Programming)
(HCI)
ML
Planning
Semantics
Garb.Coll.
Multimedia
GUI
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 also, you get papers on ML approaches to Garb. Coll.)

More Text Classification Examples

- Assigning labels to documents or web-pages:
- Labels are most often topics such as Yahoo-categories
  - "finance," "sports," "news>world>asia>business"
- Labels may be genres
  - "editorials" "movie-reviews" "news"
- Labels may be opinion on a person/product
  - “like”, “hate”, “neutral”
- Labels may be domain-specific
  - "interesting-to-me" : "not-interesting-to-me”
  - “contains adult language” : “doesn’t”
  - language identification: English, French, Chinese, …
  - search vertical: about Linux versus not
  - “link spam” : “not link spam”

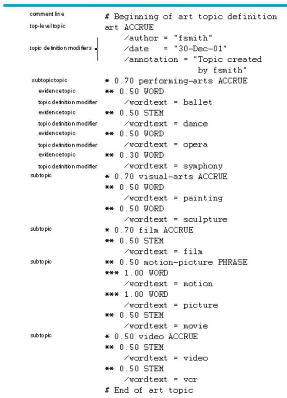
Classification Methods (1)

- Manual classification
  - Used by the original Yahoo! Directory
  - Looksmart, about.com, ODP, PubMed
  - Very accurate when job is done by experts
  - Consistent when the problem size and team is small
  - Difficult and expensive to scale
    - Means we need automatic classification methods for big problems

Classification Methods (2)

- Hand-coded rule-based classifiers
  - One technique used by CS dept’s spam filter, Reuters, CIA, etc.
  - It’s what Google Alerts is doing
    - Widely deployed in government and enterprise
  - Companies provide “IDE” for writing such rules
  - E.g., assign category if document contains a given boolean combination of words
  - Commercial systems have complex query languages (everything in IR query languages +score accumulators)
  - Accuracy is often very high if a rule has been carefully refined over time by a subject expert

A Verity topic
A complex classification rule



```
comment line                # Beginning of art topic definition
top-level topic             art ACCRUE
                                /author = "fsmith"
topic definition modifiers      /date   = "30-Dec-01"
                                /annotation = "Topic created
                                                by fsmith"
subtopictopic               * 0.70 performing-arts ACCRUE
  evidencetopic             ** 0.50 WORD
  topic definition modifier     /wordtext = ballet
  evidencetopic             ** 0.50 STEM
  topic definition modifier     /wordtext = dance
  evidencetopic             ** 0.50 WORD
  topic definition modifier     /wordtext = opera
  evidencetopic             ** 0.30 WORD
  topic definition modifier     /wordtext = symphony
subtopic                    * 0.70 visual-arts ACCRUE
                            ** 0.50 WORD
                                /wordtext = painting
                            ** 0.50 WORD
                                /wordtext = sculpture
subtopic                    * 0.70 film ACCRUE
                            ** 0.50 STEM
                                /wordtext = film
subtopic                    ** 0.50 motion-picture PHRASE
                            *** 1.00 WORD
                                /wordtext = motion
                            *** 1.00 WORD
                                /wordtext = picture
                            ** 0.50 STEM
                                /wordtext = movie
subtopic                    * 0.50 video ACCRUE
                            ** 0.50 STEM
                                /wordtext = video
                            ** 0.50 STEM
                                /wordtext = vcr
                            # End of art topic
```

- Note:
  - maintenance issues (author, etc.)
  - Hand-weighting of terms

[Verity was bought by Autonomy.]

Classification Methods (3)

- Supervised learning of a document-label assignment function
  - Many systems partly or wholly rely on machine learning (Autonomy, Microsoft, Enkata, Yahoo!, …)
    - k-Nearest Neighbors (simple, powerful)
    - Naive Bayes (simple, common method)
    - Support-vector machines (new, generally more powerful)
    - … plus many other methods
  - No free lunch: requires hand-classified training data
  - But data can be built up (and refined) by

Relevance feedback

- In relevance feedback, the user marks a few documents as relevant/nonrelevant
- The choices can be viewed as classes or categories
- The IR system then uses these judgments to build a better model of the information need
- So, relevance feedback can be viewed as a form of text classification (deciding between several classes)

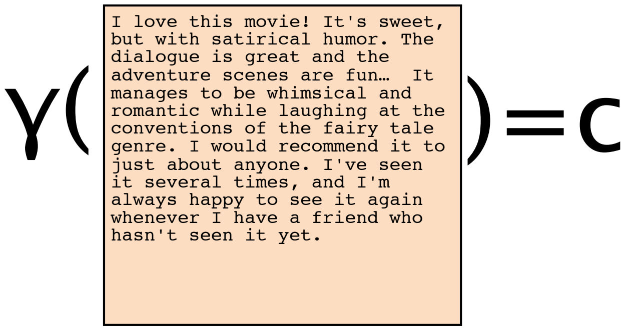
Probabilistic relevance feedback

- Rather than reweighting in a vector space…
- If user has told us some relevant and some nonrelevant documents, then we can proceed to build a probabilistic classifier
    - such as the Naive Bayes model we will look at today:
  - $P(t_k|R) = |\mathbf{D}_{rk}| \;/\; |\mathbf{D}_r|$
  - $P(t_k|NR) = |\mathbf{D}_{nrk}| \;/\; |\mathbf{D}_{nr}|$
    - t_k is a term; $\mathbf{D}_r$ is the set of known relevant documents; $\mathbf{D}_{rk}$ is the subset that contain t_k; $\mathbf{D}_{nr}$ is the set of known nonrelevant documents; $\mathbf{D}_{nrk}$ is the subset that contain t_k.

Bayesian Methods

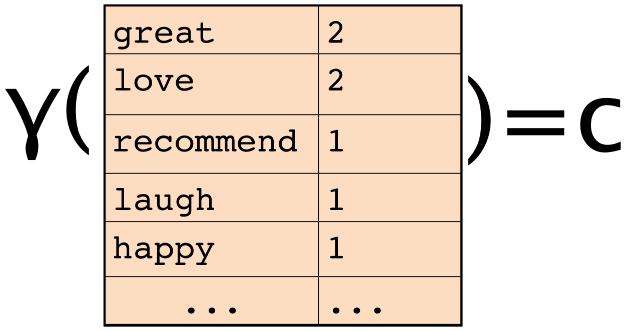
- Learning and classification methods based on probability theory
- Bayes theorem plays a critical role
- Builds a generative model that approximates how data is produced
- Has prior probability of each category given no information about an item.
- Model produces a posterior probability
  - Distribution over the possible categories given an item
- Naïve Bayes methods use a bag of words as the item description

The bag of words representation



I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun… It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet.

The bag of words representation



great	2
love	2
recommend	1
laugh	1
happy	1
...	...

Bayes' Rule for text classification

- For a document d and a class c

$$P(c,d) = P(c \mid d)P(d) = P(d \mid c)P(c)$$

$$\boxed{P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)}}$$

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 $c_j \in C$

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

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

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

MAP is “maximum a posteriori” = most likely class

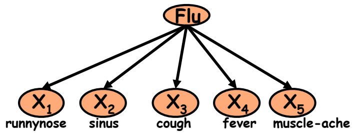
Naïve Bayes Classifier: Naïve Bayes Assumption

- $P(c_j)$
  - 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•|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 Multivariate Bernoulli NB Classifier

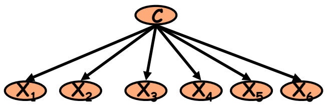


- **Conditional Independence Assumption:** features detect term presence and are independent of each other given the class:

$$P(X_1, \mathrm{K}, X_5 \mid C) = P(X_1 \mid C) \bullet P(X_2 \mid C) \bullet \mathrm{L} \bullet P(X_5 \mid C)$$

- This model is appropriate for binary

Learning the Model

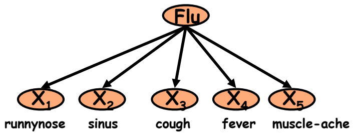


- First attempt: maximum likelihood estimates
  - simply use the frequencies in the data

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

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

Problem with Maximum Likelihood



$$P(X_1, \mathrm{K}, X_5 \mid C) = P(X_1 \mid C) \bullet P(X_2 \mid C) \bullet \mathrm{L} \bullet P(X_5 \mid C)$$

- What if we have seen no training documents with the word **muscle-ache** and classified in the topic **Flu**?

$$\hat{P}(X_5 = t \mid C = nf) = \frac{N(X_5 = t, C = nf)}{N(C = nf)} = 0$$

- Zero probabilities cannot be conditioned away, no matter the other evidence!

$$1 = \arg\max_c \hat{P}(c) \prod_i \hat{P}(x_i \mid c)$$

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

Stochastic Language Models

- Model probability of generating strings (each word in turn) in a language (commonly all strings over alphabet ∑). E.g., a unigram model

Model M

0.2	the
0.1	a
0.01	man
0.01	woman
0.03	said
0.02	likes
…	

the	man	likes	the	woman
0.2	0.01	0.02	0.2	0.01

multiply

P(s | M) = 0.00000008

Stochastic Language Models

- Model probability of generating any string

Model M1

0.2	the
0.01	class
0.0001	sayst
0.0001	pleaseth
0.0001	yon
0.0005	maiden
0.01	woman

Model M2

0.2	the
0.0001	class
0.03	sayst
0.02	pleaseth
0.1	yon
0.01	maiden
0.0001	woman

the	class	pleaseth	yon	maiden
0.2	0.01	0.0001	0.0001	0.0005
0.2	0.0001	0.02	0.1	0.01

P(s|M2) > P(s|M1)

Unigram and higher-order models

P (🔴 🟡 🔴 🔵)

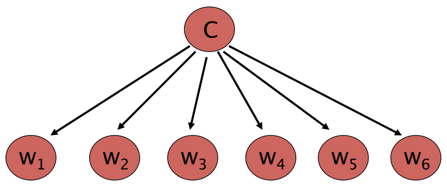
- = P (🔴) P (🟡 | 🔴) P (🔴 | 🔴 🟡) P (🔵 | 🔴 🟡 🔴)
- Unigram Language Models

  P (🔴) P (🟡) P (🔴) P (🔵)

Easy. Effective!

- Bigram (generally, n-gram) Language Models P (🔴) P (🟡 | 🔴) P (🔴 | 🟡) P (🔵 | 🔴)
- Other Language Models
  - Grammar-based models (PCFGs), etc.
    - Probably not the first thing to try in IR

Naïve Bayes via a class conditional language model = multinomial NB



- The probability of the words is done as a class-specific unigram language model

Using Multinomial Naive Bayes Classifiers to Classify Text: Basic

- Attributes are text positions, values are words.

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

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

- 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)

Naive Bayes and Language Modeling

- Naïve Bayes classifiers can use any sort of feature
  - URL, email address, dictionaries, network features
- But if, as in the previous slides
  - We use **only** word features
  - we use **all** of the words in the text (not a subset)
- Then
  - Naïve Bayes is basically the same as language modeling

Multinomial Naive 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
    - $docs_j$ ← subset of documents for which the target class is c_j
    - $P(c_j) \neg \frac{|docs_j|}{|\text{total \# documents}|}$
  - $Text_j$ ← single document containing all $docs_j$
  - for each word x_k in *Vocabulary*
    - n_k ← number of occurrences of x_k in $Text_j$
    - $P(x_k \mid c_j) \neg \frac{n_k + \alpha}{n + \alpha \, |Vocabulary|}$

Naive Bayes: Classifying

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

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

Naive Bayes: Time Complexity

- Training Time: O(|D|L_{ave} + |C||V|))
  where L_{ave} is the average length of a document in D.
  - Assumes all counts are pre-computed in O(|D|L_{ave}) time during one pass through all of the data.
  - Generally just O(|D|L_{ave}) since usually |C||V| < |D|L_{ave}
- Test Time: O(|C| L_t)
  where L_t is the average length of a test document.
- Very efficient overall, linearly proportional to the time needed to just read in all the data.

Why?

Underflow Prevention: using logs

- Multiplying lots of probabilities, which are between 0 and 1 by definition, 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 with highest final un-normalized log probability score is still the most probable.

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} [\log P(c_j) + \sum_{i \in positions} \log P(x_i \mid c_j)]$$

- Note that model is now just max of sum of weights…

Example

	Do	Words	Class
Training	1	Chinese Beijing Chinese	c
	2	Chinese Chinese Shanghai	c
	3	Chinese Macao	c
	4	Tokyo Japan Chinese	j
Test	5	Chinese Chinese Chinese Tokyo	?

Two Naive Bayes Models

- Model 1: Multivariate Bernoulli
  - One feature X_w for each word in dictionary
    - for loop iterates over dictionary
  - X_w = true in document d if w appears in d
  - Naive Bayes assumption:
    - Given the document's topic, appearance of one word in the document tells us nothing about chances that another word appears
- This is the model used in the binary independence model in classic probabilistic relevance feedback on hand-classified data

Two Models

- Model 2: Multinomial = Class conditional unigram
  - One feature X_i for each word pos in document
    - feature's values are all words in dictionary
  - Value of X_i is the word in position i
  - Naïve Bayes assumption:
    - Given the document's topic, word in one position in the document tells us nothing about words in other positions
  - Second assumption:
    - Word appearance does not depend on position

$$P(X_i = w \mid c) = P(X_j = w \mid c)$$

for all positions i,j, word w, and class c

Parameter estimation

- Multivariate Bernoulli model:
- $\hat{P}(X_w = t \mid c_j) =$ fraction of documents of topic c_j in which word w appears
- Multinomial model:

$\hat{P}(X_i = w \mid c_j) =$ fraction of times in which word w appears among all words in documents of topic c_j

  - Can create a mega-document for topic j by concatenating all documents in this topic
  - Use frequency of w in mega-document

Which to use for classification?

- Multinomial vs Multivariate Bernoulli?
- Multinomial model is almost always more effective in text applications!
  - See results figures later
- There has been exploration of multinomial naïve bayes variants which often work better in practice
  - Binarized multinomial Naïve Bayes, etc.
  - Topic of PA4

Feature Selection: Why?

- Text collections have a large number of features
  - 10,000 – 1,000,000 unique words … and more
- May make using a particular classifier feasible
  - Some classifiers can't deal with 1,000,000 features
- Reduces training time
  - Training time for some methods is quadratic or worse in the number of features
- Makes runtime models smaller and faster

Feature Selection: How?

- Two ideas:
  - Hypothesis testing statistics:
    - Are we confident that the value of one categorical variable is associated with the value of another
    - Chi-square test (χ^2)
  - Information theory:
    - How much information does the value of one categorical variable give you about the value of another
    - Mutual information
- They're similar, but χ^2 measures confidence in association, (based on available statistics), while MI measures extent of association (assuming perfect knowledge of probabilities)

Feature Selection: Frequency

- The simplest feature selection method:
  - Just use the commonest terms
  - No particular foundation
  - But it make sense why this works
    - They're the words that can be well-estimated and are most often available as evidence
  - In practice, this is often 90% as good as better methods

Feature selection for NB

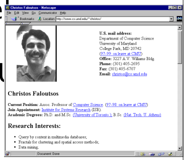
- In general feature selection is necessary for multivariate Bernoulli NB.
- Otherwise you suffer from noise, multi-counting
- "Feature selection" really means something different for multinomial NB. It means dictionary truncation
  - The multinomial NB model only has 1 feature
- This "feature selection" normally isn't needed for multinomial NB, but may help a

Evaluating Categorization

- Evaluation must be done on test data that are independent of the training data (usually a disjoint set of instances).
  - Sometimes use cross-validation (averaging results over multiple training and test splits of the overall data)
- It's easy to get good performance on a test set that was available to the learner during training (e.g., just memorize the test set).
- Measures: precision, recall, F1, classification accuracy
- Classification accuracy: c/n where n is the total number of test instances and c is the number of test instances correctly classified by the system.
  - Adequate if one class per document
  - Otherwise F measure for each class

WebKB Experiment (1998)

- Classify webpages from CS departments into:
  - student, faculty, course,project
- Train on ~5,000 hand-labeled we
  - Cornell, Washington, U.Texas, Wisconsin
- Crawl and classify a new site (CMU



Christos Faloutsos - Netscape

U.S. mail address:
Department of Computer Science
University of Maryland
College Park, MD 20742
(97-99: on leave at CMU)
Office: 3227 A.V. Williams Bldg.
Phone: (301) 405-2695
Fax: (301) 405-6707
Email: christos@cs.umd.edu

Christos Faloutsos

Current Position: Assoc. Professor of Computer Science. (97-98: on leave at CMU)
Join Appointment: Institute for Systems Research (ISR).
Academic Degrees: Ph.D. and M.Sc. (University of Toronto.); B.Sc. (Nat. Tech. U. Athens)

Research Interests:

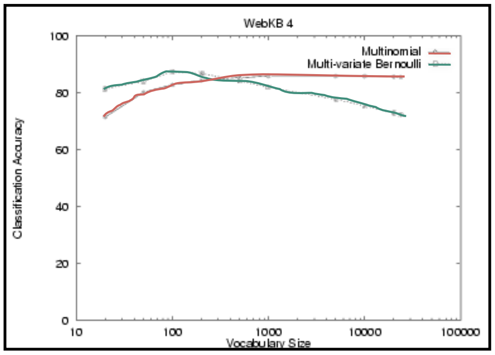
- Query by content in multimedia databases;
- Fractals for clustering and spatial access methods;
- Data mining;

File Edit View Go Communicator Help
Bookmarks Location: http://www.cs.umd.edu/~christos/
Document: Done

- Results:

	Student	Faculty	Person	Project	Course	Departmt
Extracted	180	66	246	99	28	1
Correct	130	28	194	72	25	1
Accuracy:	72%	42%	79%	73%	89%	100%

NB Model Comparison: WebKB



Faculty

associate	0.00417
chair	0.00303
member	0.00288
ph	0.00287
director	0.00282
fax	0.00279
journal	0.00271
recent	0.00260
received	0.00258
award	0.00250

Students

resume	0.00516
advisor	0.00456
student	0.00387
working	0.00361
stuff	0.00359
links	0.00355
homepage	0.00345
interests	0.00332
personal	0.00332
favorite	0.00310

Courses

homework	0.00413
syllabus	0.00399
assignments	0.00388
exam	0.00385
grading	0.00381
midterm	0.00374
pm	0.00371
instructor	0.00370
due	0.00364
final	0.00355

Departments

departmental	0.01246
colloquia	0.01076
epartment	0.01045
seminars	0.00997
schedules	0.00879
webmaster	0.00879
events	0.00826
facilities	0.00807
eople	0.00772
postgraduate	0.00764

Research Projects

investigators	0.00256
group	0.00250
members	0.00242
researchers	0.00241
laboratory	0.00238
develop	0.00201
related	0.00200
arpa	0.00187
affiliated	0.00184
project	0.00183

Others

type	0.00164
jan	0.00148
enter	0.00145
random	0.00142
program	0.00136
net	0.00128
time	0.00128
format	0.00124
access	0.00117
begin	0.00116

SpamAssassin

- Naïve Bayes has found a home in spam filtering
  - Paul Graham's A Plan for Spam
    - A Naive Bayes-like classifier with weird parameter estimation
  - Widely used in spam filters
  - But many features beyond words:
    - black hole lists, etc.
    - particular hand-crafted text patterns

Naïve Bayes in Spam Filtering

- SpamAssassin Features:
  - Basic (Naïve) Bayes spam probability
  - Mentions: Generic Viagra
  - Mentions millions of (dollar) ((dollar) NN,NNN,NNN.NN)
  - Phrase: impress ... girl
  - Phrase: 'Prestigious Non-Accredited Universities'
  -
  - From: starts with many numbers
  - Subject is all capitals
  - HTML has a low ratio of text to image area
  - Relay in RBL, http://www.mail-abuse.com/enduserinfo_rbl.html
  - RCVD line looks faked
  - http://spamassassin.apache.org/tests_3_3_x.html

Naïve Bayes Posterior Probabilities

- Classification results of naïve Bayes (the class with maximum posterior probability) are usually fairly accurate.
- However, due to the inadequacy of the conditional independence assumption, the actual posterior-probability numerical estimates are not.
  - Output probabilities are commonly very close to 0 or 1.
- Correct estimation ⇒ accurate prediction, but correct probability estimation is NOT necessary for

Naive Bayes is Not So Naive

- Very Fast Learning and Testing (basically just count the data)
- Low Storage requirements
- Very good in domains with many equally important features
- More robust to irrelevant features than many learning methods

  Irrelevant Features cancel each other without affecting results
- More robust to concept drift (changing class definition over time)
- Naive Bayes won 1st and 2nd place in KDD-CUP 97 competition out of 16 systems

  Goal: Financial services industry direct mail response prediction: Predict if the recipient of mail will actually respond

Resources for today's lecture

- IIR 13
- Fabrizio Sebastiani. Machine Learning in Automated Text Categorization. ACM Computing Surveys, 34(1):1-47, 2002.
- Yiming Yang & Xin Liu, A re-examination of text categorization methods. Proceedings of SIGIR, 1999.
- Andrew McCallum and Kamal Nigam. A Comparison of Event Models for Naive Bayes Text Classification. In AAAI/ICML-98 Workshop on Learning for Text Categorization, pp. 41-48.
- Tom Mitchell, Machine Learning. McGraw-Hill, 1997.
  - Clear simple explanation of Naïve Bayes
- Open Calais: Automatic Semantic Tagging
  - Free (but they can keep your data), provided by Thompson/Reuters (ex-ClearForest)
- Weka: A data mining software package that includes an implementation of Naive Bayes
- Reuters-21578 – the most famous text classification evaluation set
  - Still widely used by lazy people (but now it's too small for