CS276B

Text Information Retrieval, Mining, and Exploitation

Lecture 4
Text Categorization I
Introduction and Naive Bayes
Jan 21, 2003

Is this spam?

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.

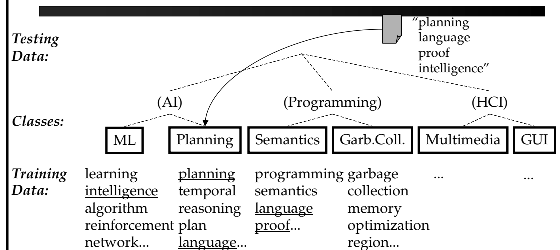
Change your life NOW !

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

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.
  - 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").

Document Classification



(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., "is a toner cartridge ad" :"isn't"

Methods (1)

- Manual classification
  - Used by Yahoo!, Looksmart, about.com, ODP, Medline
  - very accurate when job is done by experts
  - consistent when the problem size and team is small
  - difficult and expensive to scale
- Automatic document classification
  - Hand-coded rule-based systems
    - Used by CS dept's spam filter, Reuters, CIA, Verity, …
    - E.g., assign category if document contains a given boolean combination of words
    - Commercial systems have complex query languages (everything in IR query languages + *accumulators*)

Methods (2)

- Accuracy is often very high if a query has been carefully refined over time by a subject expert
- Building and maintaining these queries is expensive

- Supervised learning of document-label assignment function
  - Many new systems rely on machine learning (Autonomy, Kana, MSN, Verity, …)
    - k-Nearest Neighbors (simple, powerful)
    - Naive Bayes (simple, common method)
    - Support-vector machines (new, more powerful)
    - … plus many other methods
    - No free lunch: requires hand-classified training data
    - But can be built (and refined) by amateurs

Text Categorization: attributes

- Representations of text are very high dimensional (one feature for each word).
- High-bias algorithms that prevent overfitting in high-dimensional space are best.
- For most text categorization tasks, there are many irrelevant and many relevant features.
- Methods that combine evidence from many or all features (e.g. naive Bayes, kNN, neural-nets) tend to work better than ones that try to isolate just a few relevant features (standard decision-tree or rule induction)*

*Although one can compensate by using many rules

Bayesian Methods

- Our focus today
- 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

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

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

Maximum a posteriori Hypothesis

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

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

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

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 based on a tuple of attribute values $\langle x_1, x_2, \ldots, x_n \rangle$

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

$$c_{MAP} = \underset{c_j \in C}{\operatorname{argmax}} \frac{P(x_1, x_2, \ldots, x_n \mid c_j) P(c_j)}{P(c_1, c_2, \ldots, c_n)}$$

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

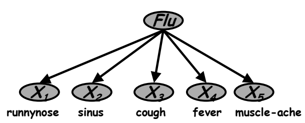
Naïve Bayes Classifier: Assumptions

- $P(c_j)$
  - Can be estimated from the frequency of classes in the training examples.
- $P(x_1,x_2,\ldots,x_n|c_j)$
  - $O(|X|^n \cdot |C|)$
  - Could only be estimated if a very, very large number of training examples was available.

Conditional Independence Assumption:

⇒ Assume that the probability of observing the conjunction of attributes is equal to the product of the individual probabilities.

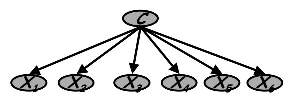
The Naïve Bayes Classifier



- **Conditional Independence Assumption:** features are independent of each other given the class:

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

Learning the Model

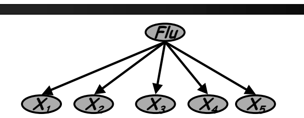


- Common practice:maximum likelihood
  - simply use the frequencies in the data

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

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

Problem with Max Likelihood



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

- What if we have seen no training cases where patient had no flu and muscle aches?

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

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

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

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 $X_i = x_{i,k}$

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

extent of "smoothing"

Using Naive Bayes Classifiers to Classify Text: Basic method

- Attributes are text positions, values are words.

$$c_{NB} = \operatorname*{argmax}_{c_j \in C} P(c_j) \prod_i P(x_i \mid c_j)$$
$$= \operatorname*{argmax}_{c_j \in C} P(c_j) P(x_1 = \text{"our"} \mid c_j) \cdots P(x_n = \text{"text"} \mid c_j)$$

- Naive Bayes assumption is clearly violated.
  - Example?
- Still too many possibilities
- Assume that classification is *independent* of the positions of the words
  - Use same parameters for each position

Text Classification Algorithms: 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 $Text_j$
    - $P(x_k \mid c_j) \leftarrow \dfrac{n_k + 1}{n + |Vocabulary|}$

Text Classification Algorithms: Classifying

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

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

Naive Bayes Time Complexity

- Training Time: $O(|D|L_d + |C||V|))$ where L_d is the average length of a document in D.
  - Assumes V and all D_i, n_i, and n_{ij} pre-computed in $O(|D|L_d)$ time during one pass through all of the data.
  - Generally just $O(|D|L_d)$ since usually $|C||V| < |D|L_d$
- 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.

Underflow Prevention

- 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.

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 generally very close to 0 or 1.

Two Models

- Model 1: Multivariate binomial
  - One feature X_w for each word in 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 document tells us nothing about chances that another word appears

Two Models

- Model 2: Multinomial
  - 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 document tells us nothing about value of 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

- Binomial model:

$$\hat{P}(X_w = t \mid c_j) = \text{fraction of documents of topic } c_j \text{ in which word } w \text{ appears}$$

- Multinomial model:

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

  - creating a mega-document for topic j by concatenating all documents in this topic
  - use frequency of w in mega-document

Feature selection via Mutual Information

- We might not want to use all words, but just reliable, good discriminators
- In training set, choose k words which best discriminate the categories.
- One way is in terms of Mutual Information:

$$I(w,c) = \sum_{e_w \in \{0,1\}} \sum_{e_c \in \{0,1\}} p(e_w, e_c) \log \frac{p(e_w, e_c)}{p(e_w)p(e_c)}$$

  - For each word w and each category c

Feature selection via MI (contd.)

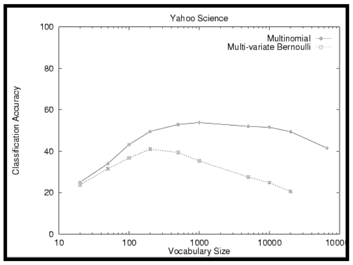
- For each category we build a list of k most discriminating terms.
- For example (on 20 Newsgroups):
  - ***sci.electronics:*** circuit, voltage, amp, ground, copy, battery, electronics, cooling, …
  - ***rec.autos:*** car, cars, engine, ford, dealer, mustang, oil, collision, autos, tires, toyota, …
- Greedy: does not account for correlations between terms
- In general feature selection is *necessary* for binomial NB, but not for multinomial NB

Evaluating Categorization

- Evaluation must be done on test data that are independent of the training data (usually a disjoint set of instances).
- *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.
- Results can vary based on sampling error due to different training and test sets.
- Average results over multiple training and test sets (splits of the overall data) for the best results.

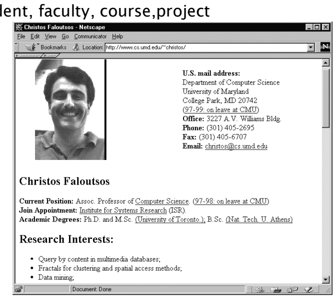
Example: AutoYahoo!

- Classify 13,589 Yahoo! webpages in “Science” subtree into 95 different topics (hierarchy depth 2)



Example: WebKB (CMU)

- Classify webpages from CS departments into:
  - student, faculty, course,project

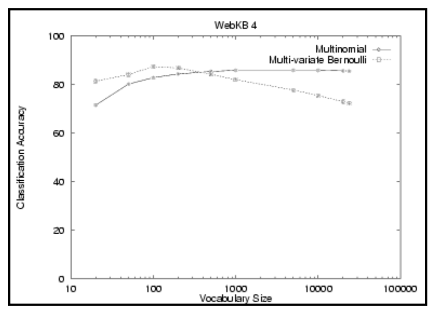


WebKB Experiment

- Train on ~5,000 hand-labeled web pages
  - Cornell, Washington, U.Texas, Wisconsin
- Crawl and classify a new site (CMU)
- 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



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

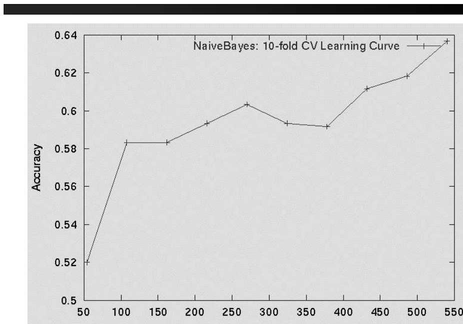
Research Projects

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

Sample Learning Curve
(Yahoo Science Data)



Importance of Conditional Independence

Assume a domain with 20 binary (true/false) attributes $A_1,\ldots, A_{20}$, and two classes c_1 and c_2.

Goal: for any case $A=A_1,\ldots,A_{20}$ estimate $P(A,c_i)$.

A) No independence assumptions:

Computation of 2^{21} parameters (one for each combination of values) !

- The training database will not be so large!
- Huge Memory requirements / Processing time.
- Error Prone (*small sample* error).

B) Strongest conditional independence assumptions (all attributes independent given the class) = Naive Bayes:

$P(A,c_i)=P(A_1,c_i)P(A_2,c_i)\ldots P(A_{20},c_i)$

Computation of $20*2^2 = 80$ parameters.

- Space and time efficient.
- Robust estimations.
- What if the conditional independence assumptions do not hold??

C) More relaxed independence assumptions

Tradeoff between A) and B)

Conditions for Optimality of Naive Bayes

Fact

Sometimes NB performs well even if the Conditional Independence assumptions are badly violated.

Questions

WHY? And WHEN?

Hint

Classification is about predicting the correct class label and NOT about accurately estimating probabilities.

Answer

Assume two classes c_1 and c_2. A new case A arrives.

NB will classify A to c_1 if:

$P(A, c_1)>P(A, c_2)$

	$P(A,c_1)$	$P(A,c_2)$	Class of A
Actual Probability	0.1	0.01	c_1
Estimated Probability by NB	0.08	0.07	c_1

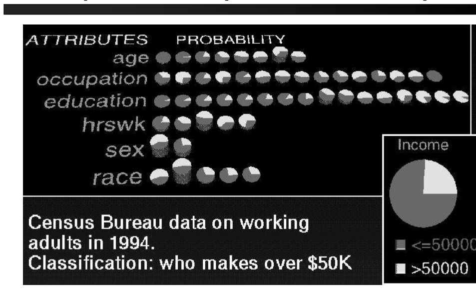
Besides the big error in estimating the probabilities the classification is still correct.

Correct estimation ⇒ accurate prediction
but NOT
~~accurate prediction ⇒ Correct estimation~~

Naive Bayes is Not So Naive

- Naïve Bayes: First and Second place in KDD-CUP 97 competition, among 16 (then) state of the art algorithms
  - Goal: Financial services industry direct mail response prediction model: Predict if the recipient of mail will actually respond to the advertisement – 750,000 records.
- Robust to Irrelevant Features
  - Irrelevant Features cancel each other without affecting results
  - Instead Decision Trees & Nearest-Neighbor methods can heavily suffer from this.
- Very good in Domains with many <u>equally important</u> features
  - Decision Trees suffer from *fragmentation* in such cases – especially if little data
- A good dependable baseline for text classification (but not the best)!
- Optimal if the Independence Assumptions hold: If assumed independence is correct, then it is the Bayes Optimal Classifier for problem
- Very Fast: Learning with one pass over the data; testing linear in the number of attributes, and document collection size
- Low Storage requirements
- Handles Missing Values

Interpretability of Naive Bayes

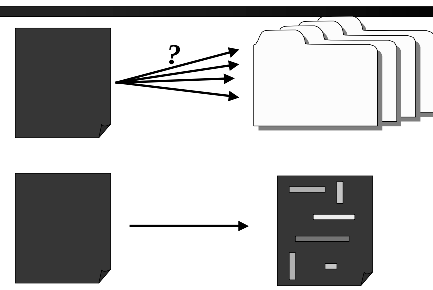


(From R.Kohavi, Silicon Graphics MineSet Evidence Visualizer)

Naive Bayes Drawbacks

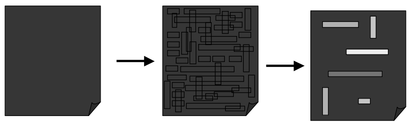
- Doesn't do higher order interactions
- Typical example: Chess end games
  - Each move completely changes the context for the next move
  - C4.5 ≈ 99.5 % accuracy : NB ≈ 87% accuracy.
- What if you have BOTH high order interactions AND few training data?
- Doesn't model features that do not equally contribute to distinguishing the classes.
  - If few features ONLY mostly determine the class, additional features usually decrease the accuracy.
  - Because NB gives same weight to all features.

Final example: Text classification *vs.* information extraction



Naive integration of IE & TC

- Use conventional classification algorithms to classify substrings of document as "*to be extracted*" or not.



- This has been tried, often with limited success [Califf, Freitag]
- But in some domains this naive technique is remarkably effective.

'Change of Address' email

From: Robert Kubinsky <robert@lousycorp.com>
Subject: Email update

Hi all - I'm moving jobs and wanted to stay in touch with everyone so....
My new email address is : robert@cubemedia.com
Hope all is well :)
>>R

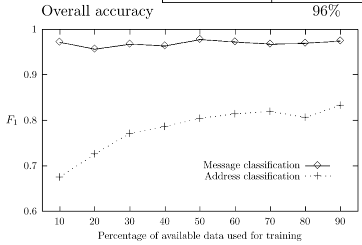
From: Robert Kubinsky <robert@lousycorp.com>
Subject: Email update

Hi all - I'm moving jobs and wanted to stay in touch with everyone so....
My new email address is : robert@cubemedia.com
Hope all is well :)
>>R

→ Modify address book, etc.

Kushmerick: CoA Results

	Words			Phrases		
	P	R	F_1	P	R	F_1
Message classification	.96	.66	.78	.98	.97	.98
Address classification	.96	.62	.76	.98	.68	.80
Overall accuracy				96%		



36 CoA messages
86 addresses
55 old, 31 new
5720 non-Coa

Resources

- Fabrizio Sebastiani. Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, 34(1):1-47, 2002.
- 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.
- Yiming Yang & Xin Liu, A re-examination of text categorization methods. *Proceedings of SIGIR*, 1999.