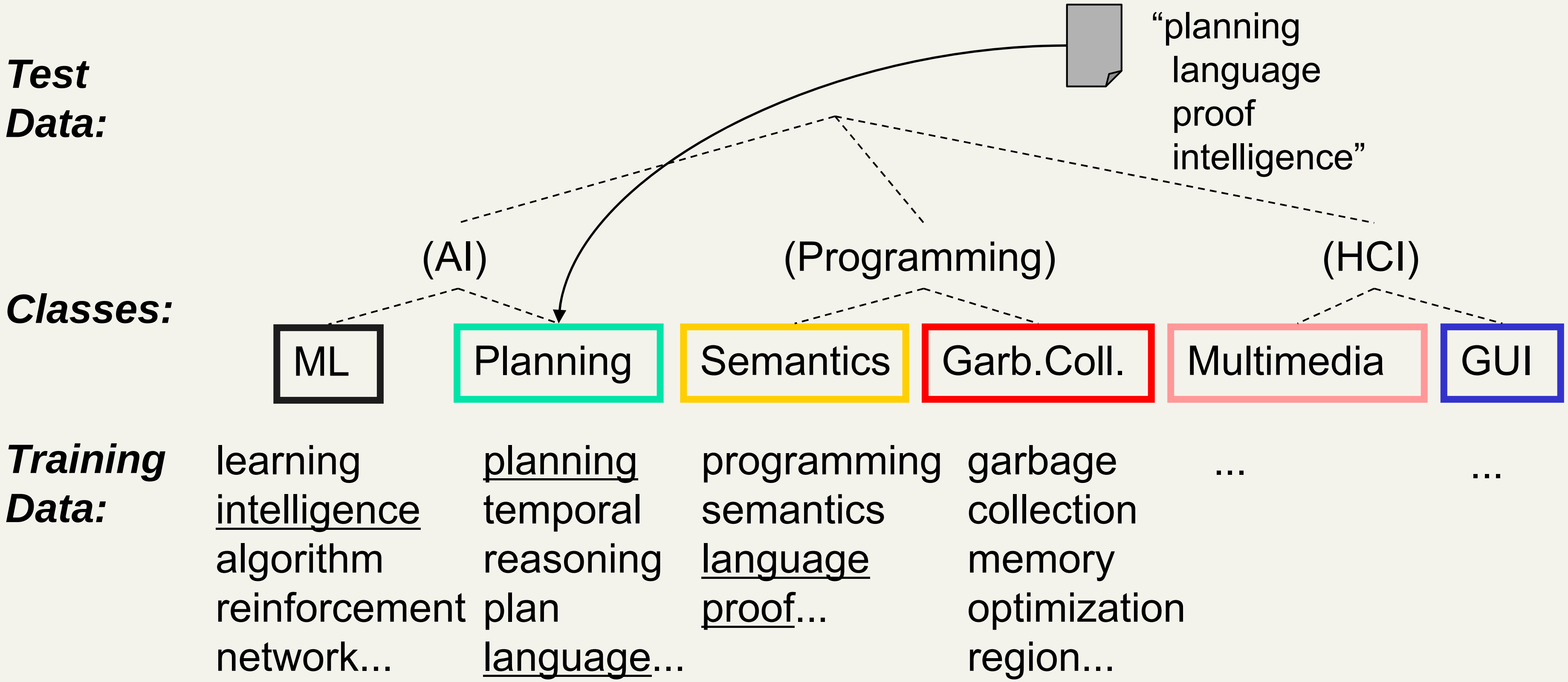
Text Classification: Naïve Bayes Algorithm

SEEM5680

Document Classification



(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.”)

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
  - 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 ("classifiers").

Supervised Classification

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

    $C = \{c_1, c_2, \ldots, c_J\}$
  - A training set D of labeled documents with each labeled document $\langle d,c \rangle \in X \times C$
- Determine:
  - A learning method or algorithm which will enable us to learn a classifier $\gamma: X \rightarrow C$
  - For a test document d, we assign it the class $\gamma(d) \in C$

More Text Classification Examples

Many search engine functionalities use classification

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)

- Automatic document classification
  - Hand-coded rule-based systems
    - 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
    - Standing queries: 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
    - Building and maintaining these rules is expensive

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
topicdefinition modifier	/wordtext = ballet
evidencetopic	** 0.50 STEM
topicdefinition modifier	/wordtext = dance
evidencetopic	** 0.50 WORD
topicdefinition modifier	/wordtext = opera
evidencetopic	** 0.30 WORD
topicdefinition 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 rely on machine learning (Autonomy, Microsoft, Enkata, Yahoo!, Google News, …)
    - 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 data can be built up (and refined) by amateurs
- Many commercial systems use a mixture of methods

Recall a few probability basics

- For events *a* and *b:*
- Bayes' Rule

$$p(a,b) = p(a \cap b) = p(a \mid b)\, p(b) = p(b \mid a)\, p(a)$$

$$p(\bar{a} \mid b)\, p(b) = p(b \mid \bar{a})\, p(\bar{a})$$

$$p(a \mid b) = \frac{p(b \mid a)\, p(a)}{p(b)} = \frac{p(b \mid a)\, p(a)}{\sum_{x=a,\bar{a}} p(b \mid x)\, p(x)}$$

Prior

Posterior

- Odds:

$$O(a) = \frac{p(a)}{p(\bar{a})} = \frac{p(a)}{1 - p(a)}$$

Probabilistic Methods

- Learning and classification methods based on probability theory.
- Bayes theorem plays a critical role in probabilistic learning and classification.
- Builds a *generative model* that approximates how data is produced
- Uses *prior* probability of each category given no information about an item.
- Categorization produces a *posterior* probability distribution over the possible categories given a description of an item.

Bayes' Rule for text classification

- For a document *d* and a class *c*

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

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

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 $c_j \in C$

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

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

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

MAP is “maximum a posteriori” = most likely class

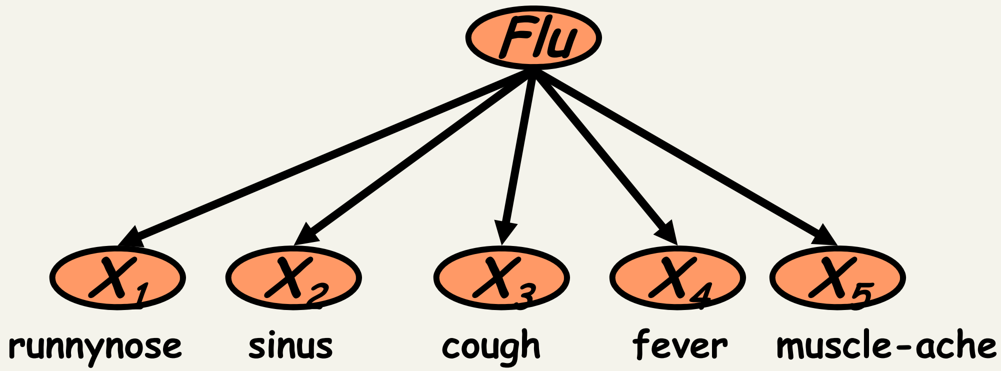
Naive Bayes Classifier: Naive Bayes Assumption

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

Naive 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 Naive Bayes Classifier



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

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

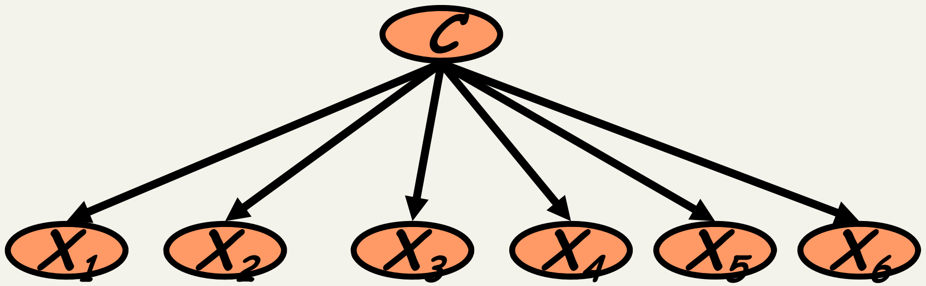
First Naive Bayes Model

- Model 1: Multivariate Bernoulli
  - 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 the document tells us nothing about chances that another word appears
- Model Learning

$$\hat{P}\left(X_w = true \middle| c_j\right) = \begin{array}{c}\text{fraction of documents of topic } c_j \\ \text{in which word } w \text{ appears}\end{array}$$

Multivariate Bernoulli Model

Learning the Model

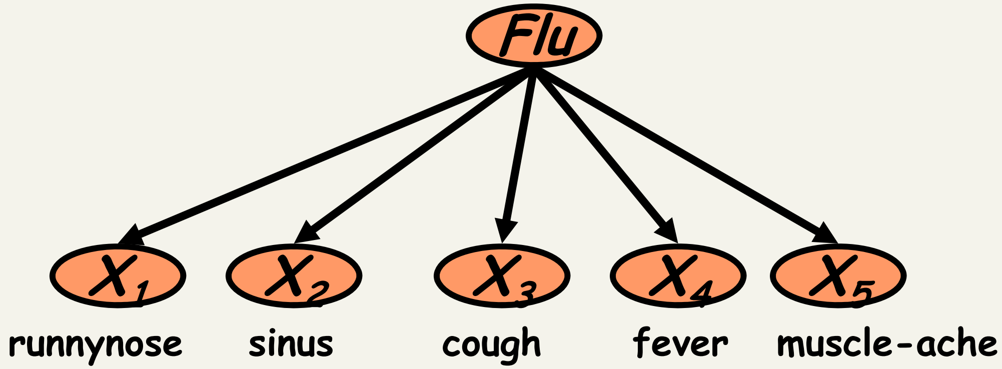


- First attempt: maximum likelihood estimates
  - simply use the frequencies in the data

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

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

Problem with Maximum 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 documents with the word ***muscle-ache*** and classified in the topic ***Flu***?

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

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

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

Smoothing to Avoid Overfitting

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

# of values of X_i

Second Model

- Model 2: Multinomial = Class conditional unigram
  - One feature X_i for each word position in document
    - feature's values are all words in dictionary
  - Value of X_i is the word in position i
  - Naive 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

    - Just have one multinomial feature predicting all words

Multinomial Naïve Bayes Model

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

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

Using Multinomial Naive Bayes Classifiers to Classify Text: Basic method

- 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

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
    - $docs_j \leftarrow$ subset of documents for which the target class is c_j
    - $$P(c_j) \leftarrow \frac{|docs_j|}{|\text{total \# documents}|}$$
  - $Text_j \leftarrow$ single document containing all $docs_j$
  - For each word x_k in *Vocabulary*
    - $n_k \leftarrow$ number of occurrences of x_k in $Text_j$
    - $$P(x_k \mid c_j) \leftarrow \frac{n_k + 1}{n + |Vocabulary|}$$

Multnomial Naive Bayes: Classifying

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

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

Multnomial Naive Bayes: Example

	docID	words in document	in c = China?
Training set	1	Chinese Beijing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	no
Test set	5	Chinese Chinese Chinese Tokyo Japan	?

$$P(c) = \frac{3}{4} \qquad P(\bar{c}) = \frac{1}{4}$$

$$P(\text{Chinese}|c) = \frac{(5+1)}{(8+6)} = \frac{6}{14} = \frac{3}{7} \qquad P(\text{Toyk}o|c) = P(\text{Japan}|c) = \frac{(0+1)}{(8+6)} = \frac{1}{14}$$

$$P(\text{Chinese}|\bar{c}) = \frac{(1+1)}{(3+6)} = \frac{2}{9} \qquad P(\text{Toyk}o|\bar{c}) = P(\text{Japan}|\bar{c}) = \frac{(1+1)}{(3+6)} = \frac{2}{9}$$

Multnomial Naive Bayes: Example

$$P(c) = \frac{3}{4} \qquad P(\bar{c}) = \frac{1}{4}$$

$$P(\text{Chinese}|c) = \frac{(5+1)}{(8+6)} = \frac{6}{14} = \frac{3}{7} \qquad P(\text{Toyk}o|c) = P(\text{Japan}|c) = \frac{(0+1)}{(8+6)} = \frac{1}{14}$$

$$P(\text{Chinese}|\bar{c}) = \frac{(1+1)}{(3+6)} = \frac{2}{9} \qquad P(\text{Toyk}o|\bar{c}) = P(\text{Japan}|\bar{c}) = \frac{(1+1)}{(3+6)} = \frac{2}{9}$$

$$P(c|d_5) \propto \frac{3}{4} \cdot \left(\frac{3}{7}\right)^3 \cdot \frac{1}{14} \cdot \frac{1}{14} \approx 0.0003$$

$$P(\bar{c}|d_5) \propto \frac{1}{4} \cdot \left(\frac{2}{9}\right)^3 \cdot \frac{2}{9} \cdot \frac{2}{9} \approx 0.0001$$

The classifier assigns the test document to $c = \text{China}$

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.

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.

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

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

Naive Bayes Classifier

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

- Simple interpretation: Each conditional parameter log $P(x_i|c_j)$ is a weight that indicates how good an indicator x_i is for c_j.
- The prior log $P(c_j)$ is a weight that indicates the relative frequency of c_j.
- The sum is then a measure of how much evidence there is for the document being in the class.
- We select the class with the most evidence for it

Feature Selection: Why?

- Text collections have a large number of features
  - 10,000 – 1,000,000 unique words … and more
- May allow using a particular classifier feasible
  - Some classifiers can't deal with 100,000 of features
- Reduces training time
  - Training time for some methods is quadratic or worse in the number of features
- Can improve generalization (performance)
  - Eliminates noise features
  - Avoids overfitting

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)

χ^2 statistic (CHI)

- $\chi 2$ is interested in $(f_o - f_e)^2/f_e$ summed over all table entries: is the observed number what you'd expect given the marginals?

$$\chi^2(j,a) = \sum (O-E)^2/E = (2-.25)^2/.25 + (3-4.75)^2/4.75$$

$$+(500-502)^2/502 + (9500-9498)^2/9498 = 12.9 \ (p < .001)$$

- The null hypothesis is rejected with confidence .999,
- since 12.9 > 10.83 (the value for .999 confidence).

	Term = jaguar	*Term ≠ jaguar*	
Class = auto	2 (0.25)	500 (502)	502
Class ≠ auto	3 (4.75)	9500 (9498)	9503
	5	10000	

expected: f_e

observed: f_o

χ^2 statistic (CHI)

There is a simpler formula for 2x2 χ^2:

$$\chi^2(t,c) = \frac{N \times (AD - CB)^2}{(A+C) \times (B+D) \times (A+B) \times (C+D)}$$

$A = \#(t,c)$	$C = \#(\neg t,c)$
$B = \#(t,\neg c)$	$D = \#(\neg t, \neg c)$

$N = A + B + C + D$

Value for complete independence of term and category?

Feature selection via Mutual Information

- In training set, choose *k* words which best discriminate (give most info on) the categories.
- The Mutual Information between a word w and a class c is:

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

where e_w =1 when the document contains the word w (0 otherwise); e_c = 1 when the document is in class c (0 otherwise)

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

Feature Selection

- Mutual Information
  - Clear information-theoretic interpretation
  - May select very slightly informative frequent terms that are not very useful for classification
- Chi-square
  - Statistical foundation
  - May select rare uninformative terms
- Just use the commonest terms?
  - No particular foundation
  - In practice, this is often 90% as good

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 fraction with quantities that are badly estimated

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

Naive Bayes vs. other methods

(a)	NB	Rocchio	kNN		SVM
micro-avg-L (90 classes)	80	85	86		89
macro-avg (90 classes)	47	59	60		60

(b)	NB	Rocchio	kNN	trees	SVM
earn	96	93	97	98	98
acq	88	65	92	90	94
money-fx	57	47	78	66	75
grain	79	68	82	85	95
crude	80	70	86	85	89
trade	64	65	77	73	76
interest	65	63	74	67	78
ship	85	49	79	74	86
wheat	70	69	77	93	92
corn	65	48	78	92	90
micro-avg (top 10)	82	65	82	88	92
micro-avg-D (118 classes)	75	62	n/a	n/a	87

Evaluation measure: F_1

Naive Bayes does pretty well, but some methods beat it consistently (e.g., SVM).

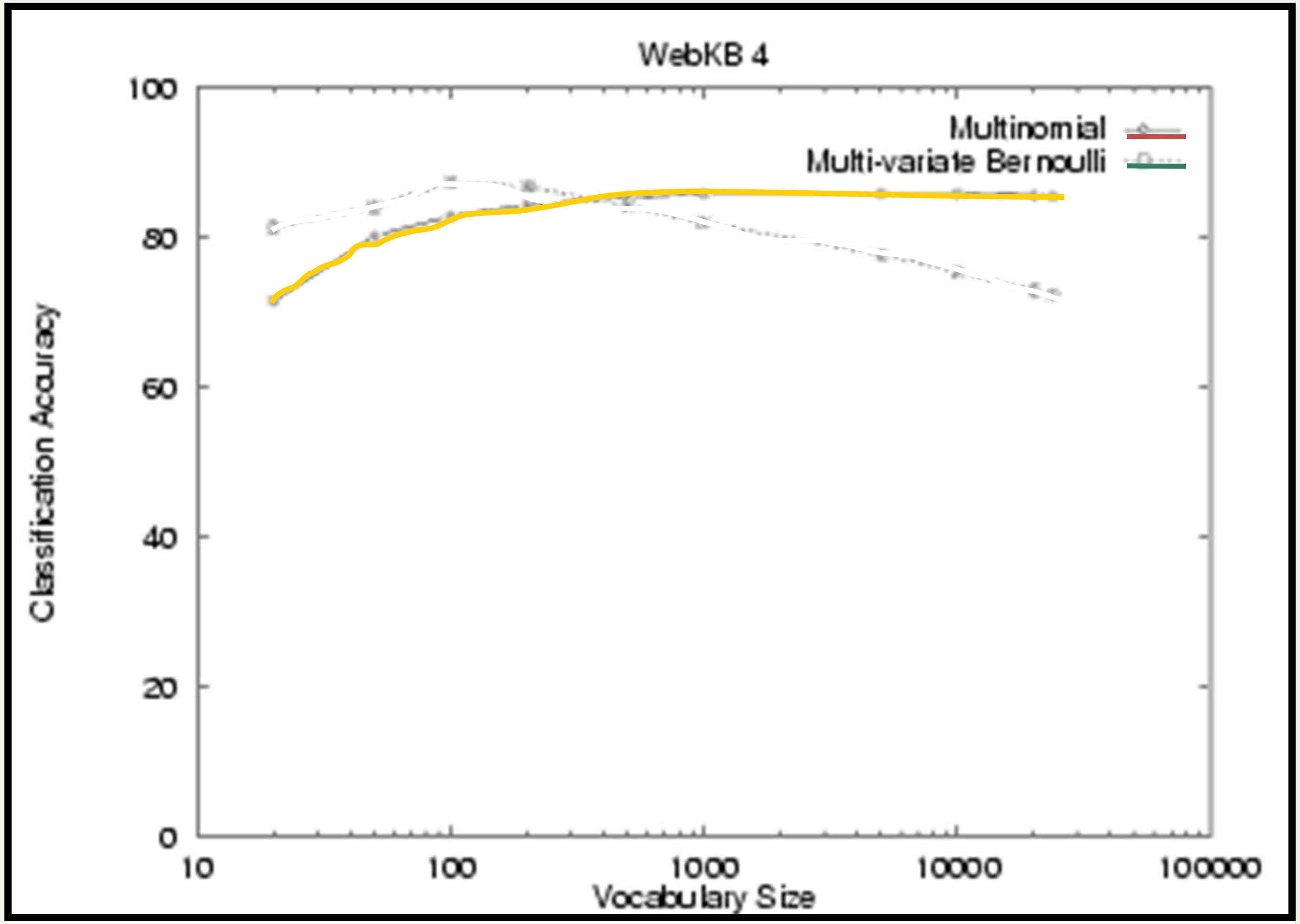
WebKB Experiment (1998)

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

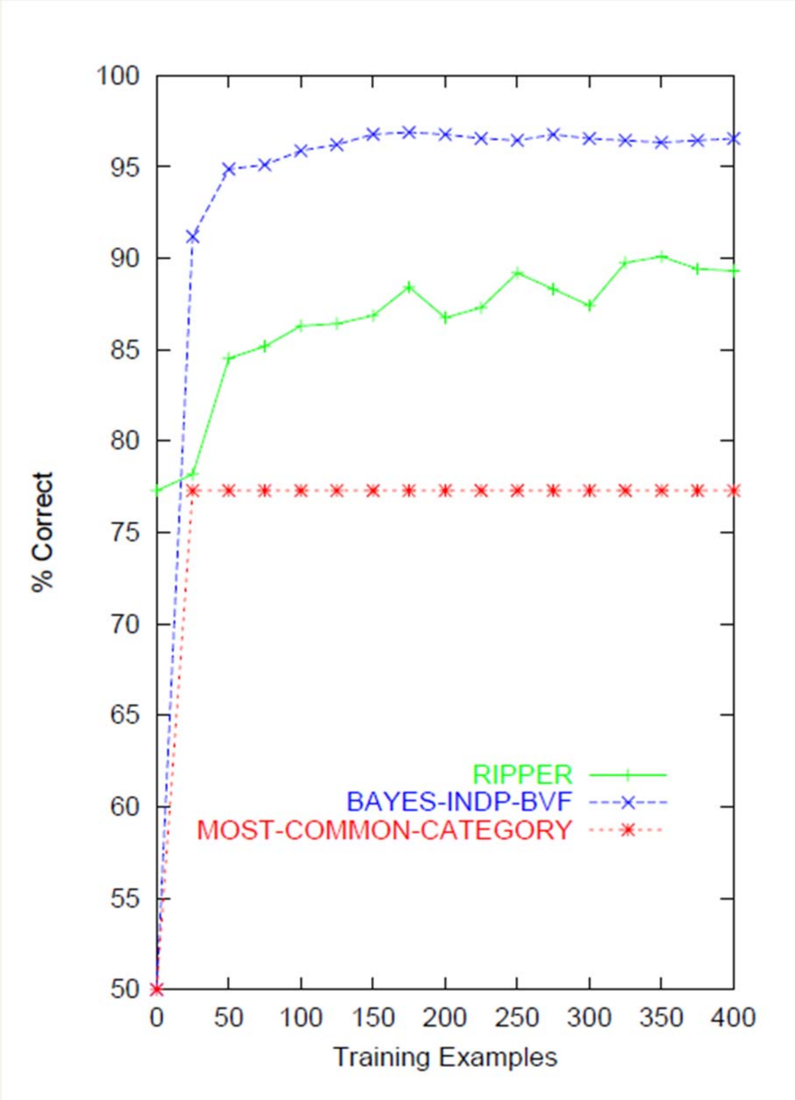
Research Projects

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

Naive Bayes on spam email



SpamAssassin

- Naive Bayes has found a home in spam filtering
  - Paul Graham's *A Plan for Spam*
    - A mutant with more mutant offspring...
  - Naive Bayes-like classifier with weird parameter estimation
  - Widely used in spam filters
    - Classic Naive Bayes superior when appropriately used
      - According to David D. Lewis
  - But also many other things: black hole lists, etc.

- Many email topic filters also use NB classifiers

Violation of NB Assumptions

- The independence assumptions do not really hold of documents written in natural language.
  - Conditional independence
  - Positional independence

Naive Bayes Posterior Probabilities

- Classification results of naive 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 $\Rightarrow$ accurate prediction, but correct probability estimation is NOT necessary for accurate prediction (just need right ordering of probabilities)

Naive Bayes is Not So Naive

- Naive Bayes won 1st and 2nd place in KDD-CUP 97 competition out of 16 systems
  - Goal: Financial services industry direct mail response prediction model: Predict if the recipient of mail will actually respond to the advertisement – 750,000 records.
- More robust to irrelevant features than many learning methods
  - Irrelevant Features cancel each other without affecting results
  - Decision Trees can suffer heavily from this.
- More robust to concept drift (changing class definition over time)
- Very good in domains with many equally important 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: Bayes Optimal Classifier
  - Never true for text, but possible in some domains
- Very Fast Learning and Testing (basically just count the data)
- Low Storage requirements

Resources

- 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 Naive 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 realistic experiments – you should use Reuters RCV1)