CIS 833 – Information Retrieval and Text Mining

Lecture 23

Text Classification

November 19, 2015

Credits for slides: Allan, Arms, Manning, Lund, Noble, Page.

Planning

- PageRank implementation: due Dec 1st
- Exam review: Dec 1st
- Final exam: Dec 3rd
- Project presentation: Dec. 17th (9:40 AM – 11:30 AM)
- Project report: Dec. 18th

Textbook Material

- Next – Text Classification
  - Chapter 13: Text Classification and Naïve Bayes
  - Chapter 14: Vector Space Classification
  - Chapter 15: Support Vector Machines

Learning Algorithms for Classification Tasks

- Relevance Feedback (Rocchio)
- k-Nearest Neighbors (simple, powerful)
- Naive Bayes (simple, common method)
- Support-vector machines (new, more powerful)
- … plus many other methods

Generative and Discriminative Models: An analogy

- The task is to determine the language that someone is speaking

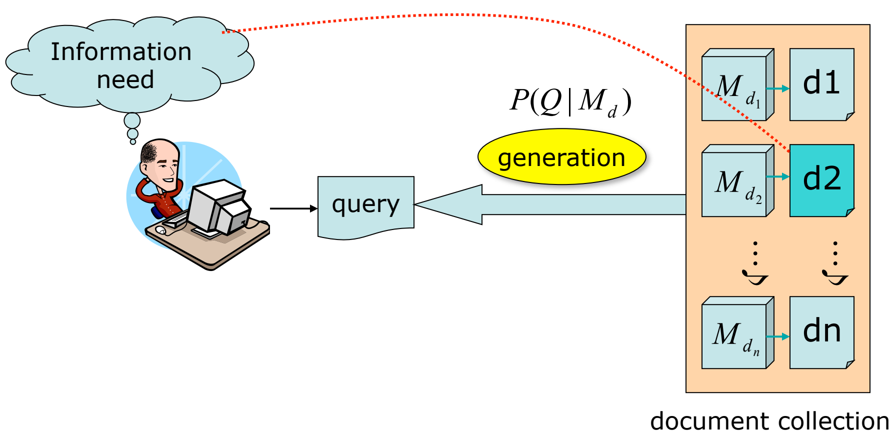
Generative and Discriminative Models: An analogy

- The task is to determine the language that someone is speaking
- Generative approach:
  - is to learn each language and determine as to which language the speech belongs to
- Discriminative approach:
  - is to determine the linguistic differences without learning any language – a much easier task!

Taxonomy of ML Models

- Generative Methods
  - Model class-conditional pdfs and prior probabilities
  - "Generative" since sampling can generate synthetic data points
  - Popular models
    - Gaussians, Naïve Bayes, Mixtures of multinomials
    - Mixtures of Gaussians, Mixtures of experts, Hidden Markov Models (HMM)
    - Sigmoidal belief networks, Bayesian networks, Markov random fields
- Discriminative Methods
  - Directly estimate posterior probabilities
  - No attempt to model underlying probability distributions
  - Focus computational resources on given task – better performance
  - Popular models
    - Logistic regression, SVMs (Kernel methods)
    - Traditional neural networks, Nearest neighbor
    - Conditional Random Fields (CRF)

Language Models



How might a query look like that would ask for a specific document? Find the document that most likely generated the query! Rank document d based on $P(M_d \mid Q)$

Generative Probabilistic Models

- Assume a simple (usually unrealistic) probabilistic method by which the data was generated.
- For categorization, each category has a different parameterized generative model that characterizes that category.

- **Training**: Use the data for each category to estimate the parameters of the generative model for that category.

- **Testing**: Use Bayesian analysis to determine the category model that most likely generated a specific test instance.

Bayes Theorem

- Bayes theorem plays a critical role in probabilistic learning and classification.
- 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.

$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}$$

Bayes Classifiers for Categorical Data

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

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

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

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

Example	Color	Shape	Class
1	red	circle	positive
2	red	circle	positive
3	red	square	negative
4	blue	circle	negative

attributes

values

Joint Distribution

- The joint probability distribution for a set of random variables, $X_1, \ldots, X_n$ gives the probability of every combination of values: $P(X_1, \ldots, X_n)$

positive

	circle	square
red	0.20	0.02
blue	0.02	0.01

negative

	circle	square
red	0.05	0.30
blue	0.20	0.20

Example	Color	Shape	Class
1	red	circle	positive
2	red	circle	positive
3	red	square	negative
4	blue	circle	negative

Joint Distribution

- The joint probability distribution for a set of random variables, $X_1,\ldots,X_n$ gives the probability of every combination of values: $P(X_1,\ldots,X_n)$

positive

	circle	square
red	0.20	0.02
blue	0.02	0.01

negative

	circle	square
red	0.05	0.30
blue	0.20	0.20

- The probability of all possible conjunctions can be calculated by summing the appropriate subset of values from the joint distribution.

$$P(red \wedge circle) = ?$$

$$P(red) = ?$$

Joint Distribution

- The joint probability distribution for a set of random variables, $X_1,\ldots,X_n$ gives the probability of every combination of values: $P(X_1,\ldots,X_n)$

positive

	circle	square
red	0.20	0.02
blue	0.02	0.01

negative

	circle	square
red	0.05	0.30
blue	0.20	0.20

- The probability of all possible conjunctions can be calculated by summing the appropriate subset of values from the joint distribution.

$$P(red \wedge circle) = 0.20 + 0.05 = 0.25$$

$$P(red) = 0.20 + 0.02 + 0.05 + 0.3 = 0.57$$

Joint Distribution

- The joint probability distribution for a set of random variables, $X_1,\ldots,X_n$ gives the probability of every combination of values: $P(X_1,\ldots,X_n)$

positive

	circle	square
red	0.20	0.02
blue	0.02	0.01

negative

	circle	square
red	0.05	0.30
blue	0.20	0.20

- The probability of all possible conjunctions can be calculated by summing the appropriate subset of values from the joint distribution.

$$P(red \wedge circle) = 0.20 + 0.05 = 0.25$$

$$P(red) = 0.20 + 0.02 + 0.05 + 0.3 = 0.57$$

- Therefore, all conditional probabilities can also be calculated.

Joint Distribution

- The joint probability distribution for a set of random variables, $X_1,\ldots,X_n$ gives the probability of every combination of values: $P(X_1,\ldots,X_n)$

positive

	circle	square
red	0.20	0.02
blue	0.02	0.01

negative

	circle	square
red	0.05	0.30
blue	0.20	0.20

- The probability of all possible conjunctions can be calculated by summing the appropriate subset of values from the joint distribution.

$$P(red \wedge circle) = 0.20 + 0.05 = 0.25$$

$$P(red) = 0.20 + 0.02 + 0.05 + 0.3 = 0.57$$

- Therefore, all conditional probabilities can also be calculated.

$$P(positive \mid red \wedge circle) = ?$$

Joint Distribution

- The joint probability distribution for a set of random variables, $X_1,\ldots,X_n$ gives the probability of every combination of values: $P(X_1,\ldots,X_n)$

positive

	circle	square
red	0.20	0.02
blue	0.02	0.01

negative

	circle	square
red	0.05	0.30
blue	0.20	0.20

- The probability of all possible conjunctions can be calculated by summing the appropriate subset of values from the joint distribution.

$$P(red \wedge circle) = 0.20 + 0.05 = 0.25$$

$$P(red) = 0.20 + 0.02 + 0.05 + 0.3 = 0.57$$

- Therefore, all conditional probabilities can also be calculated.

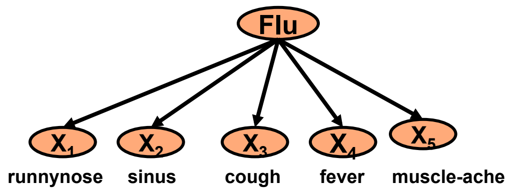
$$P(positive \mid red \wedge circle) = \frac{P(positive \wedge red \wedge circle)}{P(red \wedge circle)} = \frac{0.20}{0.25} = 0.80$$

Bayes Classifiers

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

- $P(c_j)$
  - Can be estimated from the frequency of classes in the training examples.
- $P(x_1, x_2, \ldots, x_n \mid c_j)$
  - $O(|X|^n |C|)$ parameters
  - Could only be estimated if a very, very large number of training examples was available.
  - Need to make some sort of independence assumptions about the features to make learning tractable.

The Naïve Bayes Classifier

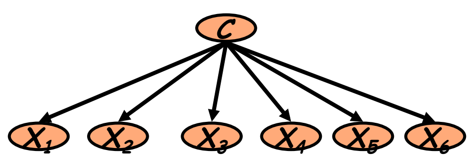


- **Conditional Independence Assumption:** attributes are independent of each other given the class:

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

- Multi-valued variables: multivariate model
- Binary variables: multivariate Bernoulli model

Learning the Model

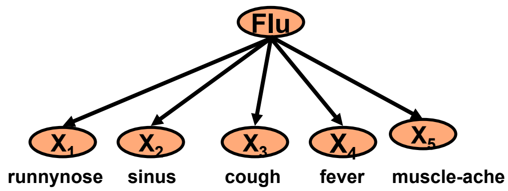


- Maximum likelihood estimates
  - simply use the frequencies in the data

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

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

Problem with 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=$x_{i,k}$

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

extent of "smoothing"

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.

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

Probability Estimation Example

Ex	Size	Color	Shape	Class
1	small	red	circle	positive
2	large	red	circle	positive
3	small	red	triangle	negative
4	large	blue	circle	negative

Probability	positive	negative
P(*Y*)		
P(small | *Y*)		
P(medium | *Y*)		
P(large | *Y*)		
P(red | *Y*)		
P(blue | *Y*)		
P(green | *Y*)		
P(square | *Y*)		
P(triangle | *Y*)		
P(circle | *Y*)		

Probability Estimation Example

Ex	Size	Color	Shape	Class
1	small	red	circle	positive
2	large	red	circle	positive
3	small	red	triangle	negative
4	large	blue	circle	negative

Probability	positive	negative
P(*Y*)	0.5	0.5
P(small | *Y*)	0.5	0.5
P(medium | *Y*)	0.0	0.0
P(large | *Y*)	0.5	0.5
P(red | *Y*)	1.0	0.5
P(blue | *Y*)	0.0	0.5
P(green | *Y*)	0.0	0.0
P(square | *Y*)	0.0	0.0
P(triangle | *Y*)	0.0	0.5
P(circle | *Y*)	1.0	0.5

Naïve Bayes Example

Probability	positive	negative
P(*Y*)	0.5	0.5
P(small | *Y*)	0.4	0.4
P(medium | *Y*)	0.1	0.2
P(large | *Y*)	0.5	0.4
P(red | *Y*)	0.9	0.3
P(blue | *Y*)	0.05	0.3
P(green | *Y*)	0.05	0.4
P(square | *Y*)	0.05	0.4
P(triangle | *Y*)	0.05	0.3
P(circle | *Y*)	0.9	0.3

Test Instance:
<medium ,red, circle>

$$c_{MAP} = \operatorname{argmax}_c \hat{P}(c) \prod_i \hat{P}(x_i \mid c)$$

Naïve Bayes Example

Probability	positive	negative
P(Y)	0.5	0.5
P(medium | Y)	0.1	0.2
P(red | Y)	0.9	0.3
P(circle | Y)	0.9	0.3

Test Instance:
<medium ,red, circle>

P(positive | X) =?

P(negative | X) =?

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

Naïve Bayes Example

Probability	positive	negative
P(Y)	0.5	0.5
P(medium | Y)	0.1	0.2
P(red | Y)	0.9	0.3
P(circle | Y)	0.9	0.3

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

Test Instance:
<medium ,red, circle>

P(positive | X) = P(positive)*P(medium | positive)*P(red | positive)*P(circle | positive) / P(X)
0.5 * 0.1 * 0.9 * 0.9
= 0.0405 / P(X) = 0.0405 / 0.0495 = 0.8181

P(negative | X) = P(negative)*P(medium | negative)*P(red | negative)*P(circle | negative) / P(X)
0.5 * 0.2 * 0.3 * 0.3
= 0.009 / P(X) = 0.009 / 0.0495 = 0.1818

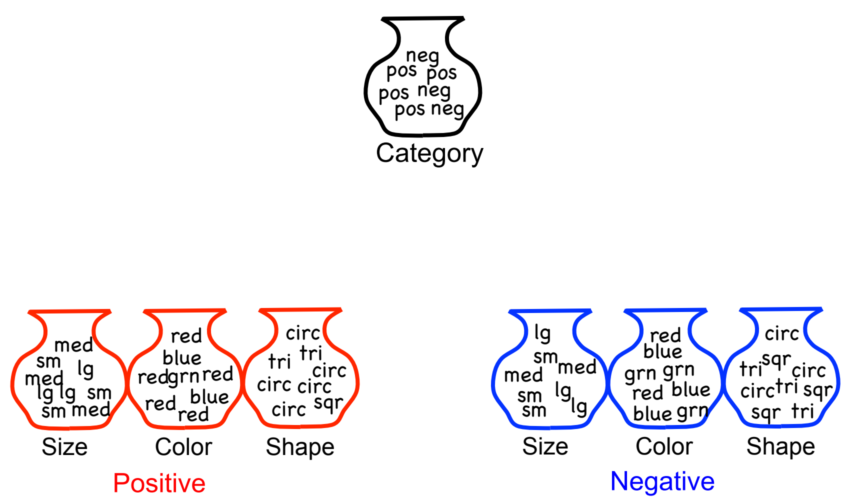
P(positive | X) + P(negative | X) = 0.0405 / P(X) + 0.009 / P(X) = 1

P(X) = (0.0405 + 0.009) = 0.0495

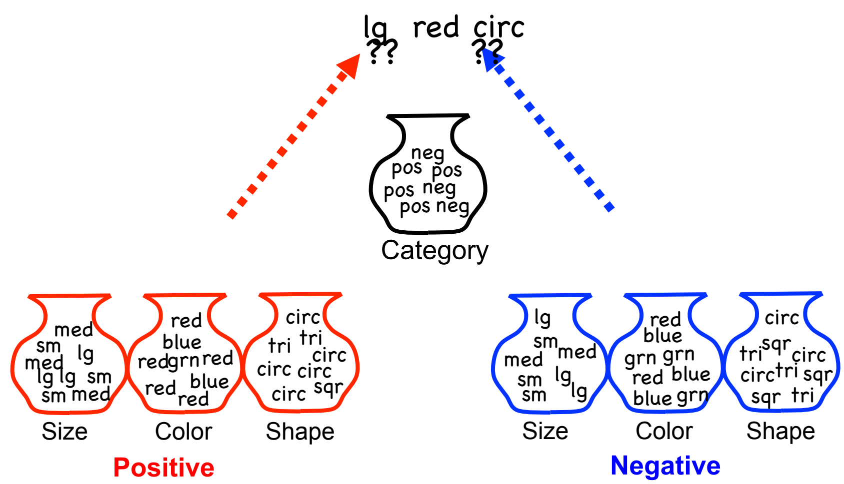
Question

- How can we see the multivariate Naïve Bayes model as a generative model?

Naïve Bayes Generative Model



Naïve Bayes Inference Problem



Naïve Bayes for Text Classification

Two models:

- Multivariate Bernoulli Model
- Multinomial Model

Model 1: Multivariate Bernoulli

- One feature X_w for each word in dictionary
- X_w = true (1) 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

- Parameter estimation

$$\hat{P}(X_w = 1 | c_j) = ?$$

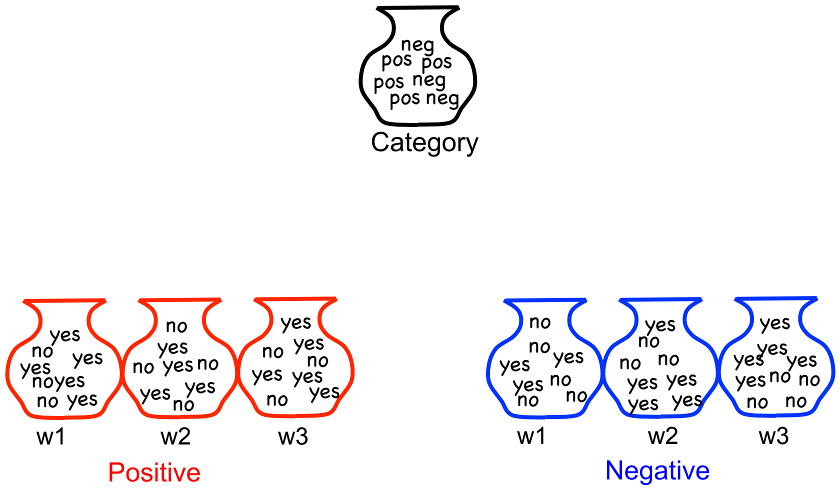
Model 1: Multivariate Bernoulli

- One feature X_w for each word in dictionary
- X_w = true (1) 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

- Parameter estimation

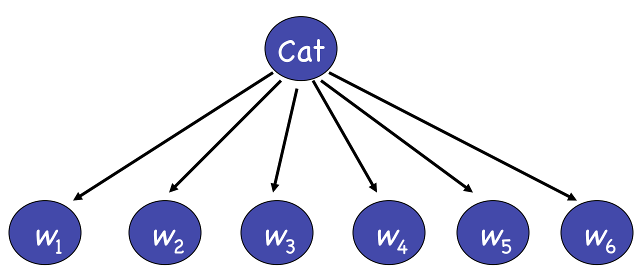
$$\hat{P}(X_w = 1 | c_j) = \text{fraction of documents of topic } c_j \text{ in which word } w \text{ appears}$$

Naïve Bayes Generative Model



Model 2: Multinomial

Naïve Bayes via a class conditional language model



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

Multinomial Distribution

“The binomial distribution is the probability distribution of the number of "successes" in n independent Bernoulli trials, with the same probability of "success" on each trial. In a multinomial distribution, each trial results in exactly one of some fixed finite number k of possible outcomes, with probabilities p_1, ..., pk (so that $pi \geq 0$ for $i = 1, ..., k$ and there sum is 1), and there are n independent trials. Then let the random variables Xi indicate the number of times outcome number i was observed over the n trials. $X=(X1,\ldots,Xn)$ follows a multinomial distribution with parameters n *and* **p,** where **p = (p_1, ..., pk).”** **(Wikipedia)**

$$f(x_1,\ldots,x_k;n,p_1,\ldots,p_k) = \Pr(X_1 = x_1 \text{ and } \ldots \text{ and } X_k = x_k)$$

$$= \begin{cases} \dfrac{n!}{x_1!\cdots x_k!} p_1^{x_1}\cdots p_k^{x_k}, & \text{when } \sum_{i=1}^k x_i = n \\ 0 & \text{otherwise,} \end{cases}$$

Multinomial Naïve Bayes

- Class conditional unigram language
  - Attributes are text positions, values are words.
  - 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
  - Naïve Bayes assumption:
    - Given the document’s topic, word in one position in the document tells us nothing about words in other positions

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

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

  - Too many possibilities!

Multinomial Naive Bayes Classifiers

- Second assumption:
  - Classification is *independent* of the positions of the words (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*

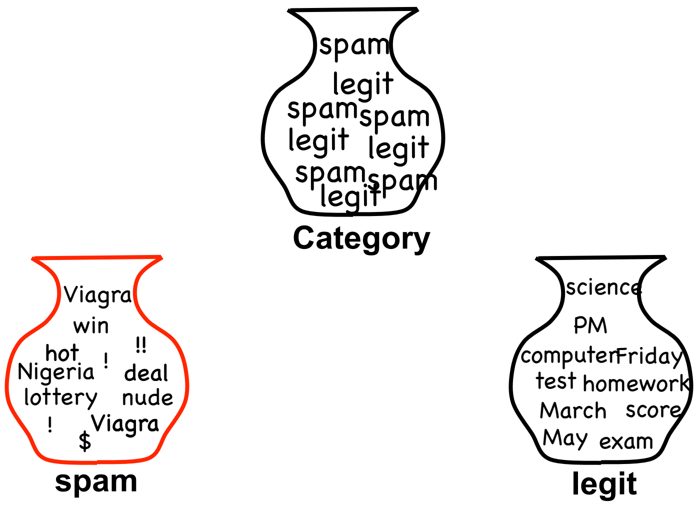
- Use same parameters for each position
- Result is bag of words model (over tokens)
- Just have one multinomial feature predicting all words

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

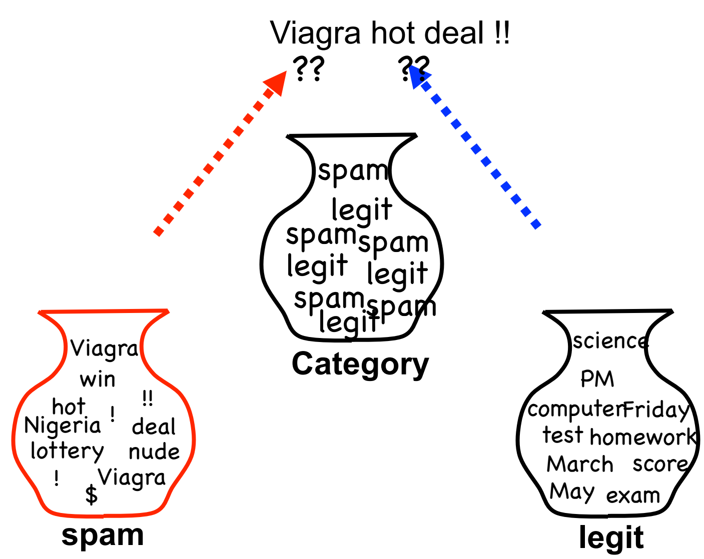
Multinomial Naïve Bayes for Text

- Modeled as generating a bag of words for a document in a given category by repeatedly sampling with replacement from a vocabulary $V = \{w_1, w_2, \ldots w_m\}$ based on the probabilities $P(w_j \mid c_i)$.
- Smooth probability estimates with Laplace *m*-estimates assuming a uniform distribution over all words ($p = 1/|V|$) and $m = |V|$

Multinomial Naïve Bayes as a Generative Model for Text



Naïve Bayes Inference Problem



Naïve Bayes Classification

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

Parameter Estimation

- Multivariate Bernoulli model:

$$\hat{P}(X_w = 1 \mid c_j) = \text{fraction of documents of topic } c_j \text{ in which word } w \text{ appears}$$

- Multinomial model:

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

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

A Variant of the Multinomial Model

- Represent each document d as an M-dimensional vector of counts $tf_{t1,d},...,tf_{tM,d}$,where $tf_{ti,d}$ is the term frequency of t_i in d.

$$P(d|c) = P(\langle \mathrm{tf}_{t_1,d}, \ldots, \mathrm{tf}_{t_M,d} \rangle | c) = \prod_{1 \le i \le M} P(X = t_i | c)^{\mathrm{tf}_{t_i,d}}$$

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

$$f(x_1, \ldots, x_k; n, p_1, \ldots, p_k) = \Pr(X_1 = x_1 \text{ and } \ldots \text{ and } X_k = x_k)$$

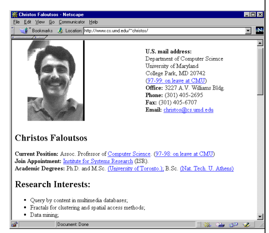
$$= \begin{cases} \dfrac{n!}{x_1! \cdots x_k!} p_1^{x_1} \cdots p_k^{x_k}, & \text{when } \sum_{i=1}^k x_i = n \\ 0 & \text{otherwise,} \end{cases}$$

Classification

- Multinomial vs Multivariate Bernoulli?
- Multinomial model is almost always more effective in text applications!

WebKB Experiment (1998)

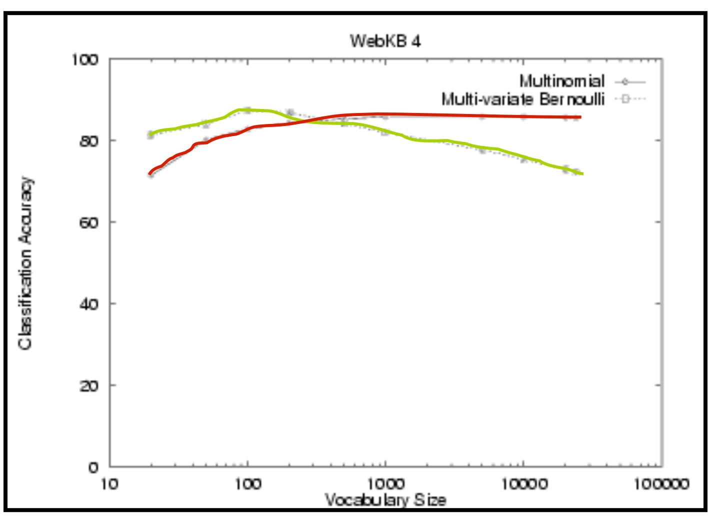
- Classify webpages from CS departments into:
  - student, faculty, course, project, etc.
- Train on ~5,000 hand-labeled web pages
  - Cornell, Washington, U.Texas, Wisconsin
- Crawl and classify a new site (CMU)



- 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



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 100,000 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 for NB

- In general feature selection is *necessary* for multivariate Bernoulli NB.
- Otherwise you suffer from noise, multi-counting

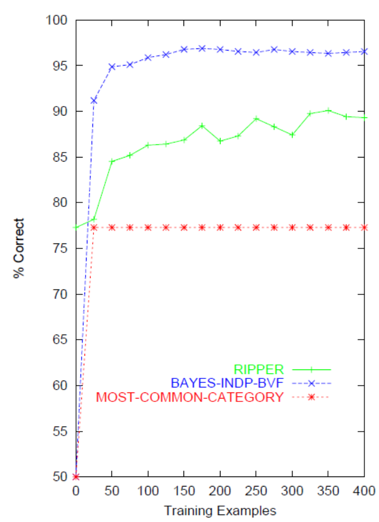
- "Feature selection" really means something different for multinomial NB. It means dictionary truncation
  - The multinomial NB model only has 1 feature
- This "feature selection" normally isn't needed for multinomial NB, but may help a fraction with quantities that are badly estimated

Naïve Bayes - Spam Assassin

- Naïve 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

Naïve Bayes on Spam Email



http://www.cs.utexas.edu/users/jp/research/email.paper.pdf

Violation of NB Assumptions

- Conditional independence
- “Positional independence”

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

Naive Bayes is Not So Naive

- 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 can heavily suffer from this.

- 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: If assumed independence is correct, then it is the Bayes Optimal Classifier for problem
- Very Fast: Learning with one pass of counting over the data; testing linear in the number of attributes, and document collection size
- Low Storage requirements