Part 9: Text Classification; The Naïve Bayes algorithm

Francesco Ricci

Most of these slides comes from the course:

Information Retrieval and Web Search, Christopher Manning and Prabhakar Raghavan
Content

- Introduction to Text Classification
- Bayes rule
- Naïve Bayes text classification
- Feature independence assumption
- Multivariate and Multinomial approaches
- Smoothing (avoid overfitting)
- Feature selection
  - Chi square and Mutual Information
- Evaluating NB classification.
Relevance feedback revisited

- In relevance feedback, the user marks a number of documents as relevant/nonrelevant
- We then try to use this information to return better search results
- Suppose we just tried to learn a filter for non-relevant documents
- This is an instance of a text classification problem:
  - Two “classes”: relevant, non-relevant
  - For each document, decide whether it is relevant or non-relevant
- The notion of classification is very general and has many applications within and beyond information retrieval.
Standing queries

- The path from information retrieval to text classification:
  - You have an information need, 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 classification not ranking
- Such queries are called **standing queries**
  - Long used by “information professionals”
  - A modern mass instantiation is **Google Alerts**.
Google alerts

Welcome to Google Alerts

Google Alerts are email updates of the latest relevant Google results (web, news, etc.) based on your choice of query or topic.

Some handy uses of Google Alerts include:

- monitoring a developing news story
- keeping current on a competitor or industry
- getting the latest on a celebrity or event
- keeping tabs on your favorite sports teams

Create an alert with the form on the right.

You can also sign in to manage your alerts

Create a Google Alert

Enter the topic you wish to monitor.

Search terms: [input field]

Type: [Comprehensive]

How often: [once a day]

Email length: [up to 20 results]

Your email: [input field]

Create Alert

Google will not sell or share your email address.
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
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 – the representation determines what information is used for solving the classification task
  - A fixed set of classes:
    \( C = \{c_1, c_2, \ldots, c_J\} \)

- Determine:
  - The class of \( x \): \( c(x) \in C \), where \( c(x) \) is a classification function whose domain is \( X \) and whose range is \( C \)
  - We want to know how to build classification functions ("classifiers").
Document Classification

Test Data:

Classes:
- 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."
Many search engine functionalities use classification.

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 on a person/product. e.g., “like”, “hate”, “neutral”
- Labels may be domain-specific. e.g., "interesting-to-me" : "not-interesting-to-me”
  e.g., “contains adult language” : “doesn’t”
  e.g., language identification: English, French, Chinese, ...
  e.g., “link spam” : “not link spam”
  e.g., "key-phrase" : "not key-phrase"
Classification Methods (1)

- Manual classification
  - Used by Yahoo! (originally; now present but downplayed), 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 systems**
  - One technique used by CS dept’s spam filter, Reuters, CIA, etc.
  - Companies (Verity) provide “IDE” for writing such rules
  - Example: *assign category if document contains a given Boolean combination of words*
  - **Standing queries:** Commercial systems have complex query languages (everything in IR query languages + 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!
Verity topic (a classification rule)

- Note:
  - maintenance issues (author, etc.)
  - Hand-weighting of terms
  - But it is easy to explain the results.
Classification Methods (3)

- **Supervised learning** of a document-label assignment function
- Many systems partly rely on machine learning (Autonomy, MSN, Verity, Enkata, Yahoo!, ...)
  - 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**
- Note that many commercial systems use a mixture of methods.
Recall a few probability basics

- For events $a$ and $b$:
- **Bayes’ Rule**

\[
p(a, b) = p(a \cap b) = p(a | b) p(b) = p(b | a) p(a)
\]

\[
p(a | b) = \frac{p(b | a) p(a)}{p(b)} = \frac{p(b | a) p(a)}{\sum_{x=a,\bar{a}} p(b | x) p(x)}
\]

- **Odds:**

\[
O(a) = \frac{p(a)}{p(\bar{a})} = \frac{p(a)}{1 - p(a)}
\]
Bayes’ Rule Example

\[ P(C, E) = P(C \mid E)P(E) = P(E \mid C)P(C) \]

\[ P(C \mid E) = \frac{P(E \mid C)P(C)}{P(E)} \]

\[
P(\text{pass exam } \mid \text{ attend classes}) = ?
\]

= \[ P(\text{pass exam}) \times \frac{P(\text{attend classes } \mid \text{ pass exam})}{P(\text{attend classes})} \]

= 0.7 \times \frac{0.9}{0.78}

= 0.7 \times 1.15 = 0.81

Initial estimation

Correction based on a ratio
Example explained

P(pass) = 0.7
P(attend) = P(attend | pass)P(pass) + P(attend | not pass)P(not pass)
= 0.9*0.7 + 0.5*0.3 = 0.63 + 0.15 = 0.78
p(pass | attend) = p(pass)*p(attend | pass)/p(attend)
= 0.7 * 0.9/0.78 = 0.81
Bayesian Methods

- Our focus this lecture
- Learning and classification methods based on **probability theory**
- **Bayes theorem** plays a critical role in probabilistic learning and classification
- Uses **prior** probability of each category given no information about an item
- Obtains a **posterior probability** distribution over the possible categories **given a description of an item**.
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 $c_j \in C$

$$c_{MAP} = \arg\max_{c_j \in C} P(c_j | x_1, x_2, \ldots, x_n)$$

$$= \arg\max_{c_j \in C} \frac{P(x_1, x_2, \ldots, x_n | c_j)P(c_j)}{P(x_1, x_2, \ldots, x_n)}$$

$$= \arg\max_{c_j \in C} P(x_1, x_2, \ldots, x_n | c_j)P(c_j)$$

Maximum A Posteriori class
Naïve Bayes Assumption

- $P(c_j)$
  - Can be estimated from the frequency of classes in the training examples

- $P(x_1, x_2, ..., x_n | c_j)$
  - $O(|X|^n |C|)$ parameters (assuming $X$ finite)
  - Could only be estimated if a very, very large number of training examples was available – or?

- Naïve Bayes Conditional Independence Assumption:
  - Assume that the probability of observing the conjunction of attributes is equal to the product of the individual probabilities $P(x_i | c_j)$.

$$c_{MAP} = \arg\max_{c_j \in C} P(c_j) \prod_{i=1}^{n} P(x_i | c_j)$$
The Naïve Bayes Classifier

- **Conditional Independence Assumption:**
  - features detect term presence and are independent of each other given the class:
  
  \[ P(x_1, \ldots, x_5 | C) = P(x_1 | C) \cdot P(x_2 | C) \cdot \ldots \cdot P(x_5 | C) \]

- This model is appropriate for **binary** variables
  - Multivariate Bernoulli model
    - = many variables
    - = only 2 values – T or F
First attempt: maximum likelihood estimates
- simply use the frequencies in the data

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

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

Estimated conditional probability that the attribute \(X_i\) (e.g. Fever) has the value \(x_i\) (True or False) – we will also write \(P(\text{Fever} \mid c_j)\) instead of \(P(\text{Fever} = T \mid c_j)\)
Problem with Max Likelihood

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

\[
P(x_1,\ldots,x_5 \mid C) = P(x_1 \mid C) \cdot P(x_2 \mid C) \cdots \cdot P(x_5 \mid C)
\]

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

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

\[
\hat{P}(X_5 = T \mid C = \text{flu}) = \frac{N(X_5 = T, C = \text{flu})}{N(C = \text{flu})} = 0
\]

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

\[
\hat{P}(x_i \mid c_j) = \frac{N(X_i = x_i, C = c_j) + mP(X_i = x_i)}{N(C = c_j) + m}
\]

overall fraction in data where \(X_i=x_i\)

extent of “smoothing” 23
Example

<table>
<thead>
<tr>
<th>docID</th>
<th>words in document</th>
<th>in $c = \text{China}$?</th>
</tr>
</thead>
<tbody>
<tr>
<td>training set 1</td>
<td>Chinese Beijing Chinese</td>
<td>yes</td>
</tr>
<tr>
<td>2</td>
<td>Chinese Chinese Shanghai</td>
<td>yes</td>
</tr>
<tr>
<td>3</td>
<td>Chinese Macao</td>
<td>yes</td>
</tr>
<tr>
<td>4</td>
<td>Tokyo Japan Chinese</td>
<td>no</td>
</tr>
<tr>
<td>test set 5</td>
<td>Chinese Chinese Chinese Tokyo Japan</td>
<td>?</td>
</tr>
</tbody>
</table>

\[
\hat{P}(\text{Chinese}|c) = \frac{3 + 1}{3 + 2} = \frac{4}{5}
\]
\[
\hat{P}(\text{Japan}|c) = \hat{P}(\text{Tokyo}|c) = \frac{0 + 1}{3 + 2} = \frac{1}{5}
\]
\[
\hat{P}(\text{Beijing}|c) = \hat{P}(\text{Macao}|c) = \hat{P}(\text{Shanghai}|c) = \frac{1 + 1}{3 + 2} = \frac{2}{5}
\]
\[
\hat{P}(\text{Chinese}|\overline{c}) = \frac{1 + 1}{1 + 2} = \frac{2}{3}
\]
\[
\hat{P}(\text{Japan}|\overline{c}) = \hat{P}(\text{Tokyo}|\overline{c}) = \frac{1 + 1}{1 + 2} = \frac{2}{3}
\]
\[
\hat{P}(\text{Beijing}|\overline{c}) = \hat{P}(\text{Macao}|\overline{c}) = \hat{P}(\text{Shanghai}|\overline{c}) = \frac{0 + 1}{1 + 2} = \frac{1}{3}
\]

\[
\hat{P}(c|d_5) \propto \hat{P}(c) \cdot \hat{P}(\text{Chinese}|c) \cdot \hat{P}(\text{Japan}|c) \cdot \hat{P}(\text{Tokyo}|c) \\
\quad \cdot (1 - \hat{P}(\text{Beijing}|c)) \cdot (1 - \hat{P}(\text{Shanghai}|c)) \cdot (1 - \hat{P}(\text{Macao}|c))
\]
\[
= \frac{3}{4} \cdot \frac{4}{5} \cdot \frac{1}{5} \cdot \frac{1}{5} \cdot \left(1 - \frac{2}{5}\right) \cdot \left(1 - \frac{2}{5}\right) \cdot \left(1 - \frac{2}{5}\right)
\]
\[
\approx 0.005
\]
Exercise

<table>
<thead>
<tr>
<th>docID</th>
<th>words in document</th>
<th>in c = China?</th>
</tr>
</thead>
<tbody>
<tr>
<td>training set</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Chinese Beijing Chinese</td>
<td>yes</td>
</tr>
<tr>
<td>2</td>
<td>Chinese Chinese Shanghai</td>
<td>yes</td>
</tr>
<tr>
<td>3</td>
<td>Chinese Macao</td>
<td>yes</td>
</tr>
<tr>
<td>4</td>
<td>Tokyo Japan Chinese</td>
<td>no</td>
</tr>
<tr>
<td>test set</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Chinese Chinese Chinese Tokyo Japan</td>
<td>?</td>
</tr>
</tbody>
</table>

\[
\hat{P}(\text{Chinese}|c) = \frac{3+1}{3+2} = \frac{4}{5}
\]

\[
\hat{P}(\text{Japan}|c) = \hat{P}(\text{Tokyo}|c) = \frac{0+1}{3+2} = \frac{1}{5}
\]

\[
\hat{P}(\text{Beijing}|c) = \hat{P}(\text{Macao}|c) = \hat{P}(\text{Shanghai}|c) = \frac{1+1}{3+2} = \frac{2}{5}
\]

\[
\hat{P}(\text{Chinese}|\overline{c}) = \frac{1+1}{1+2} = \frac{2}{3}
\]

\[
\hat{P}(\text{Japan}|\overline{c}) = \hat{P}(\text{Tokyo}|\overline{c}) = \frac{1+1}{1+2} = \frac{2}{3}
\]

\[
\hat{P}(\text{Beijing}|\overline{c}) = \hat{P}(\text{Macao}|\overline{c}) = \hat{P}(\text{Shanghai}|\overline{c}) = \frac{0+1}{1+2} = \frac{1}{3}
\]

Estimate the probability that the test document does not belong to class c.
Exercise

<table>
<thead>
<tr>
<th>docID</th>
<th>words in document</th>
<th>in $c = \text{China}$?</th>
</tr>
</thead>
<tbody>
<tr>
<td>training set 1</td>
<td>Chinese Beijing Chinese</td>
<td>yes</td>
</tr>
<tr>
<td>2</td>
<td>Chinese Chinese Shanghai</td>
<td>yes</td>
</tr>
<tr>
<td>3</td>
<td>Chinese Macao</td>
<td>yes</td>
</tr>
<tr>
<td>4</td>
<td>Tokyo Japan Chinese</td>
<td>no</td>
</tr>
<tr>
<td>test set 5</td>
<td>Chinese Chinese Chinese Tokyo Japan</td>
<td>?</td>
</tr>
</tbody>
</table>

$$
P(\text{Chinese}|c) = \frac{3 + 1}{3 + 2} = \frac{4}{5}$$

$$
P(\text{Japan}|c) = P(\text{Tokyo}|c) = \frac{0 + 1}{3 + 2} = \frac{1}{5}$$

$$
P(\text{Beijing}|c) = P(\text{Macao}|c) = P(\text{Shanghai}|c) = \frac{1 + 1}{3 + 2} = \frac{2}{5}$$

$$
P(\text{Chinese}|\overline{c}) = \frac{1 + 1}{1 + 2} = \frac{2}{3}$$

$$
P(\text{Japan}|\overline{c}) = P(\text{Tokyo}|\overline{c}) = \frac{1 + 1}{1 + 2} = \frac{2}{3}$$

$$
P(\text{Beijing}|\overline{c}) = P(\text{Macao}|\overline{c}) = P(\text{Shanghai}|\overline{c}) = \frac{0 + 1}{1 + 2} = \frac{1}{3}$$

$$
\hat{P}(\overline{c}|d_5) \propto 1/4 \cdot 2/3 \cdot 2/3 \cdot 2/3 \cdot (1-1/3) \cdot (1-1/3) \cdot (1-1/3)
\approx 0.022
$$
**Multinomial Naive Bayes Classifiers: Basic method**

- Attributes ($X_i$) are text positions, values ($x_i$) are words:

$$c_{NB} = \arg \max_{c_j \in C} P(c_j) \prod_{i=1}^{n} P(x_i \mid c_j)$$

$$= \arg \max_{c_j \in C} P(c_j) P(X_1 = "Our" \mid c_j) \cdots P(X_n = "text." \mid c_j)$$

- Still too many possibilities

- Assume that classification is *independent* of the positions of the words

$$P(X_i = w \mid c) = P(X_j = w \mid c)$$
Multinomial Naïve Bayes: Learning

- From training corpus, extract *Vocabulary*
- Calculate required $P(c_j)$ and $P(x_k | c_j)$ terms
  - For each class $c_j$ 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*
    - $n_{jk} \leftarrow$ number of occurrences of $x_k$ in $Text_j$
    - $n_j \leftarrow$ number of words in $Text_j$
      $$P(x_k | c_j) \leftarrow \frac{n_{jk} + \alpha}{n_j + \alpha |Vocabulary|}$$
      Assume is $= 1$; this is for smoothing
Multinomial Naïve Bayes: Classifying

- positions ← all word positions in current document which contain tokens found in Vocabulary

- Return $c_{NB}$ such that:

$$c_{NB} = \arg\max_{c_j \in C} P(c_j) \prod_{i \in \text{positions}} P(x_i | c_j)$$
Example

<table>
<thead>
<tr>
<th>docID</th>
<th>words in document</th>
<th>in $c = \text{China}$?</th>
</tr>
</thead>
<tbody>
<tr>
<td>training set</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Chinese Beijing Chinese</td>
<td>yes</td>
</tr>
<tr>
<td>2</td>
<td>Chinese Chinese Shanghai</td>
<td>yes</td>
</tr>
<tr>
<td>3</td>
<td>Chinese Macao</td>
<td>yes</td>
</tr>
<tr>
<td>4</td>
<td>Tokyo Japan Chinese</td>
<td>no</td>
</tr>
<tr>
<td>test set</td>
<td>Chinese Chinese Chinese Tokyo Japan</td>
<td>?</td>
</tr>
</tbody>
</table>

\[
\hat{P}(\text{Chinese} \mid c) = \frac{5 + 1}{8 + 6} = \frac{6}{14} = \frac{3}{7}
\]
\[
\hat{P}(\text{Tokyo} \mid c) = \hat{P}(\text{Japan} \mid c) = \frac{0 + 1}{8 + 6} = \frac{1}{14}
\]
\[
\hat{P}(\text{Chinese} \mid \neg c) = \frac{1 + 1}{3 + 6} = \frac{2}{9}
\]
\[
\hat{P}(\text{Tokyo} \mid \neg c) = \hat{P}(\text{Japan} \mid \neg c) = \frac{1 + 1}{3 + 6} = \frac{2}{9}
\]

\[
\hat{P}(c \mid d_5) \propto 3/4 \cdot (3/7)^3 \cdot 1/14 \cdot 1/14 \approx 0.0003.
\]
\[
\hat{P}(\neg c \mid d_5) \propto 1/4 \cdot (2/9)^3 \cdot 2/9 \cdot 2/9 \approx 0.0001.
\]
Naive Bayes: Time Complexity

- **Training Time**: if $L_d$ is the average length of a document in $D$
  - $O(|D|L_d + |C||V|)$
  - Assumes that $V$ and all $docs_j$, $n_j$, and $n_{jk}$ are 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: log space

- 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

\[
c_{NB} = \arg\max_{c_j \in C} \left( \log P(c_j) + \sum_{i \in \text{positions}} \log P(x_i | c_j) \right)
\]

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

Sounds familiar?
Summary - Two Models: **Multivariate Bernoulli**

- One feature $X_w$ for each word in dictionary
- $X_w = \text{true in document } d \text{ if } w \text{ appears in } d$
- **Naive Bayes assumption:**
  - Given the document’s topic (class), appearance of one word in the document tells us nothing about chances that another word appears (independence)
Summary - Two Models: **Multinomial**

- One feature $X_i$ for each word positions 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 (class), word in one position in the document tells us nothing about words in other positions

**Second assumption:**
- Word appearance does not depend on position - for all positions $i,j$, word $w$, and class $c$

$$P(X_i = w | c) = P(X_j = w | c)$$

- Just have one multinomial feature predicting all words.
Parameter estimation

- **Multivariate Bernoulli model:**
  \[
  \hat{P}(X_w = \text{true} \mid c_j) = \frac{\text{fraction of documents of topic } c_j \text{ in which word } w \text{ appears}}{
  \text{fraction of times in which word } w \text{ appears across all documents of topic } c_j}
  \]

- **Multinomial model:**
  \[
  \hat{P}(X_i = w \mid c_j) = \frac{\text{fraction of times in which word } w \text{ appears across all documents of topic } c_j}{\text{frequency of documents of topic } c_j}\]

- Can create a mega-document for topic \( j \) by concatenating all documents in this topic
- Use frequency of \( w \) in mega-document.
Classification

- Multinomial vs Multivariate Bernoulli?

- **Multinomial** model is almost always more **effective** in text applications!
  - See results figures later

- See *IIR* sections 13.2 and 13.3 for worked examples with each model
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 unfeasible
  - 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* ([http://faculty.vassar.edu/lowry/webtext.html chapter 8](http://faculty.vassar.edu/lowry/webtext.html))
  - **Information theory:**
    - How much information does the value of one categorical variable give you about the value of another
    - *Mutual information*
\( \chi^2 \) statistic (CHI) – testing independence of class and term

<table>
<thead>
<tr>
<th></th>
<th>Term = jaguar</th>
<th>Term ( \neq ) jaguar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class = auto</td>
<td>2</td>
<td>500</td>
</tr>
<tr>
<td>Class ( \neq ) auto</td>
<td>3</td>
<td>9500</td>
</tr>
</tbody>
</table>

- If the term "jaguar" is **independent** from the class "auto" we should have:
  - \( P(C(d) = \text{auto}, d \text{ contains jaguar}) = P(C(d) = \text{auto}) \times P(d \text{ contains jaguar}) \)
  - \( 2/10005 =? 502/10005 \times 5/10005 \)
  - \( 0.00019 =? 0.0501 \times 0.00049 = 0.000025 \) NOT REALLY!

- To be independent we should have more documents that contain jaguar but are not in class auto (38).
**χ² statistic (CHI)**

- χ² 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 (the two variables are independent) is rejected with confidence .999,
- since 12.9 > 10.83 (the critical value for .999 confidence for a 1 degree of freedom χ² distribution).

<table>
<thead>
<tr>
<th></th>
<th>Term = jaguar</th>
<th>Term ≠ jaguar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class = auto</td>
<td>2 (0.25)</td>
<td>500 (502)</td>
</tr>
<tr>
<td>Class ≠ auto</td>
<td>3 (4.75)</td>
<td>9500 (9498)</td>
</tr>
</tbody>
</table>

Expected value, for instance in the up-left cell is:
\[
P(c=auto) * P(T=jaguar) * \# of cases = \frac{502}{10005} * \frac{5}{10005} * 10005 = 0.2508
\]
\( \chi^2 \) statistic (CHI)

There is a “simpler” formula for 2x2 \( \chi^2 \):

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

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>C = #(¬t, c)</td>
</tr>
<tr>
<td>B = #(t, ¬c)</td>
<td>D</td>
</tr>
</tbody>
</table>

\[
N = A + B + C + D
\]
Feature selection via Mutual Information

- In training set, choose $k$ words which give most info on the knowledge of the categories.
- The **Mutual Information** between a word, class is:

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

- $U=1$ ($U=0$) means the document (does not) contains $w$
- $C=1$ ($C=0$) the document is (not) in class $c$
- For each word $w$ and each category $c$
- $I(X,Y) = H(X) - H(X|Y)$
  $$= -\sum_i p(x_i) \log p(x_i) + \sum_i p(y_j) H(X|Y = y_j)$$
- $H$ is called the Entropy
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
- Why?
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 statistically correlated but uninformative terms

- Just use the commonest terms?
  - No particular foundation
  - In practice, this is often 90% as good.
Example
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).

- **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 (*and positive and negative examples have similar cardinalities*)
  - Otherwise **F measure** for each class

- 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.
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:

<table>
<thead>
<tr>
<th></th>
<th>Student</th>
<th>Faculty</th>
<th>Person</th>
<th>Project</th>
<th>Course</th>
<th>Departmt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extracted</td>
<td>180</td>
<td>66</td>
<td>246</td>
<td>99</td>
<td>28</td>
<td>1</td>
</tr>
<tr>
<td>Correct</td>
<td>130</td>
<td>28</td>
<td>194</td>
<td>72</td>
<td>25</td>
<td>1</td>
</tr>
<tr>
<td>Accuracy:</td>
<td>72%</td>
<td>42%</td>
<td>79%</td>
<td>73%</td>
<td>89%</td>
<td>100%</td>
</tr>
</tbody>
</table>

Actually this is not accuracy but ...
Most relevant features: MI

<table>
<thead>
<tr>
<th>Faculty</th>
<th>Students</th>
<th>Courses</th>
</tr>
</thead>
<tbody>
<tr>
<td>associate</td>
<td>resume</td>
<td>homework</td>
</tr>
<tr>
<td>chair</td>
<td>advisor</td>
<td>syllabus</td>
</tr>
<tr>
<td>member</td>
<td>student</td>
<td>assignments</td>
</tr>
<tr>
<td>ph</td>
<td>working</td>
<td>exam</td>
</tr>
<tr>
<td>director</td>
<td>stuff</td>
<td>grading</td>
</tr>
<tr>
<td>fax</td>
<td>links</td>
<td>midterm</td>
</tr>
<tr>
<td>journal</td>
<td>homepage</td>
<td>pm</td>
</tr>
<tr>
<td>recent</td>
<td>interests</td>
<td>instructor</td>
</tr>
<tr>
<td>received</td>
<td>personal</td>
<td>due</td>
</tr>
<tr>
<td>award</td>
<td>favorite</td>
<td>final</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Departments</th>
<th>Research Projects</th>
<th>Others</th>
</tr>
</thead>
<tbody>
<tr>
<td>departmental</td>
<td>investigators</td>
<td>type</td>
</tr>
<tr>
<td>colloquia</td>
<td>group</td>
<td>jan</td>
</tr>
<tr>
<td>apartment</td>
<td>members</td>
<td>enter</td>
</tr>
<tr>
<td>seminars</td>
<td>researchers</td>
<td>random</td>
</tr>
<tr>
<td>schedules</td>
<td>laboratory</td>
<td>program</td>
</tr>
<tr>
<td>webmaster</td>
<td>develop</td>
<td>net</td>
</tr>
<tr>
<td>events</td>
<td>related</td>
<td>time</td>
</tr>
<tr>
<td>facilities</td>
<td>arpa</td>
<td>format</td>
</tr>
<tr>
<td>people</td>
<td>affiliated</td>
<td>access</td>
</tr>
<tr>
<td>postgraduate</td>
<td>project</td>
<td>begin</td>
</tr>
</tbody>
</table>
Naïve Bayes on spam email
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
Resources

- IIR 13
  - Clear simple explanation of Naïve Bayes