Introduction to Information Retrieval

CS276: Information Retrieval and Web Search

Lecture 10: Text Classification; The Naive Bayes algorithm

Introduction to Information Retrieval

Standing queries

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

Spam filtering

Another text classification task

From: "<takworld@hotmail.com>
Subject: real estate is the only way... gem oalvg kayak
Anyone can buy real estate with no money down
Stop paying rent TODAY!
There is no need to spend hundreds or even thousands for similar courses
I am 22 years old and I have already purchased 6 properties using the methods outlined in this truly INCREDIBLE ebook.
Change your life NOW!

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

Text classification

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

Categorization/Classification

- Given:
  - A description of an instance, \(d \in X\)
    - \(X\) is the instance language or instance space.
    - Issue: how to represent text documents.
    - Usually some type of high-dimensional space - bag of words
  - A fixed set of classes:
    - \(C = \{c_1, c_2, \ldots, c_J\}\)

- Determine:
  - The category of \(d\): \(\gamma(d) \in C\), where \(\gamma(d)\) is a classification function whose domain is \(X\) and whose range is \(C\).
Machine Learning: 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 \( \gamma(d) \in C \)

Document Classification

<table>
<thead>
<tr>
<th>Classes:</th>
<th>Classes:</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML</td>
<td>Planning</td>
</tr>
<tr>
<td>Planning</td>
<td>Semantics</td>
</tr>
<tr>
<td>Garb.Coll</td>
<td>Multimedia</td>
</tr>
<tr>
<td>GUI</td>
<td>GUI</td>
</tr>
</tbody>
</table>

(Note: in real life there is often a hierarchy, not present in the above problem statement; and also, you get papers on ML approaches to Garb. Coll.)

More Text Classification Examples

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

Classification Methods (1)

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

Classification Methods (2)

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

A Verity topic

A complex classification rule

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

[Verity was bought by Autonomy.]
Classification Methods (3)

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

Relevance feedback

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

Probabilistic relevance feedback

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

Bayesian Methods

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

The bag of words representation

- \( \gamma(\cdot) = c \)

The bag of words representation

- \( \gamma(\cdot) = c \)

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

- For a document \(d\) and a class \(c\)

\[
P(c, d) = P(c \mid d) P(d) = P(d \mid c) P(c)
\]

\[
P(c \mid d) = \frac{P(d \mid c) P(c)}{P(d)}
\]

Naive Bayes Classifiers

Task: Classify a new instance \(d\) based on a tuple of attribute values into one of the classes \(\mathbf{C} = \{x_1, x_2, \ldots, x_n\}\)

\[
c_{MAP} = \arg \max_{c \in \mathbf{C}} P(c_j \mid x_1, x_2, K, x_n) = \arg \max_{c \in \mathbf{C}} \frac{P(x_1, x_2, K, x_n \mid c_j) P(c_j)}{P(x_1, x_2, K, x_n)} = \arg \max_{c \in \mathbf{C}} P(x_1, x_2, K, x_n \mid c_j) P(c_j)
\]

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.
- \(P(x_j, x_{2j}, \ldots, x_n \mid c_j)\)
  - \(O(|X| \cdot |C|)\) 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 \(P(x_i \mid c_j)\).

The Multivariate Bernoulli NB Classifier

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

\[
P(X_{1j} = 1, X_{2j} = 1, \ldots, X_{nj} = 1 \mid c_j) = P(X_{1j} = 1 \mid c_j) \cdot P(X_{2j} = 1 \mid c_j) \cdot \ldots \cdot P(X_{nj} = 1 \mid c_j)
\]

- This model is appropriate for binary

Learning the Model

- First attempt: maximum likelihood estimates
  - simply use the frequencies in the data
  - \(\hat{P}(c_j) = \frac{N(C = c_j)}{N}\)
  - \(\hat{P}(x_i \mid c_j) = \frac{N(X_i = x_i, C = c_j)}{N(C = c_j)}\)

Problem with Maximum Likelihood

- \(P(X_1, X_2, K, X_n \mid C) = P(X_1 \mid C) \cdot P(X_2 \mid C) \cdot \ldots \cdot P(X_n \mid C)\)
- What if we have seen no training documents with the word muscle-ache and classified in the topic Flu?

\[
\hat{P}(X_1 = t \mid C = nf) = \frac{N(X_1 = t, C = nf)}{N(C = nf)} = 0
\]

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

\[
1 = \arg \max_c \hat{P}(c) \prod_i \hat{P}(x_i \mid c)
\]
Smoothing to Avoid Overfitting

\[ \hat{P}(x_i \mid c_j) = \frac{N(X_i = x_i, C = c_j) + 1}{N(C = c_j) + k} \]

- Somewhat more subtle version

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

Stochastic Language Models

- Model probability of generating any string

<table>
<thead>
<tr>
<th>Model M1</th>
<th>Model M2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2 the</td>
<td>0.2 the</td>
</tr>
<tr>
<td>0.01 class</td>
<td>0.01 class</td>
</tr>
<tr>
<td>0.0001 says</td>
<td>0.0001 says</td>
</tr>
<tr>
<td>0.0001 pleaseth</td>
<td>0.0001 pleaseth</td>
</tr>
<tr>
<td>0.0001 yon</td>
<td>0.0001 yon</td>
</tr>
<tr>
<td>0.0005 maiden</td>
<td>0.0005 maiden</td>
</tr>
<tr>
<td>0.01 woman</td>
<td>0.0001 woman</td>
</tr>
</tbody>
</table>

\[ P(s \mid M_2) > P(s \mid M_1) \]

Stochastic Language Models

- Model probability of generating strings (each word in turn) in a language (commonly all strings over alphabet \( \Sigma \)).

Stochastic Language Models

Unigram and higher-order models

\[ P(o_1 \cdot \cdot o_n) = P(o_1) P(o_2 \mid o_1) P(o_3 \mid o_1, o_2) \cdot \cdot P(o_n \mid o_1, \cdot \cdot, o_{n-1}) \]

\[ P(o) \cdot P(o) \cdot P(o) \cdot P(o) \cdot P(o) \]

- Unigram Language Models

\[ P(o) \cdot P(o) \cdot P(o) \cdot P(o) \cdot P(o) \]

- Bigram (generally, n-gram) Language Models

\[ P(o) \cdot P(o) \cdot P(o) \cdot P(o) \cdot P(o) \]

- Other Language Models

Naïve Bayes via a class conditional language model = multinomial NB

\[ P(o_1 \cdot \cdot o_n) = \prod_{i=1}^{n} P(x_i \mid c_j) \]

\[ c_{NB} = \arg \max_{c_j} \prod_{i=1}^{n} P(x_i \mid c_j) \]

Naïve Bayes via a class conditional language model = multinomial NB

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

Using Multinomial Naive Bayes Classifiers to Classify Text: Basic

- Attributes are text positions, values are words.

\[ c_{NB} = \arg \max_{c_j} \prod_{i=1}^{n} P(x_i \mid c_j) \]

\[ = \arg \max_{c_j} \prod_{i=1}^{n} P(x_i = \text{"text"} \mid c_j) \]

- Still too many possibilities

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

- Use same parameters for each position

- Result is bag of words model (over tokens not types)
Introduction to Information Retrieval

Naive Bayes and Language Modeling

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

Naive Bayes: Classifying

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

\[
c_{NB} = \arg\max_{c \in C} P(c) \prod_{i \in \text{positions}} P(x_i | c)
\]

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.

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

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

Multinomial Naive Bayes: Learning

- From training corpus, extract Vocabulary
- Calculate required \(P(c)\) and \(P(x_i | c)\) terms
  - For each \(c_i\) in \(C\) do
    - \(doc_{c_i}\) ← subset of documents for which the target class is \(c_i\)
    - \(P(c_i) = \frac{|doc_{c_i}|}{\text{total # documents}}\)
  - \(Text_j\) ← single document containing all \(doc_{c_i}\)
  - for each word \(x_k\) in Vocabulary
    - \(n_{k,c_i}\) ← number of occurrences of \(x_k\) in \(Text_j\)
    - \(P(x_k | c_i) = \frac{n_{k,c_i} + \alpha}{\sum_{c_j} n_{k,c_j} + \alpha |\text{Vocabulary}|}\)

Naive Bayes: Time Complexity

- Training Time: \(O(|D| L_{\text{ave}} + |C| |V|)\)
  - Assumes all counts are pre-computed in \(O(|D| L_{\text{ave}})\) time during one pass through all of the data.
  - Generally just \(O(|D| L_{\text{ave}})\) since usually \(|C| |V| < |D| L_{\text{ave}}\)
- Test Time: \(O(|C| L_t)\)
  - \(L_t\) is the average length of a test document.

Very efficient overall, linearly proportional to the time needed to just read in all the data.

Example

<table>
<thead>
<tr>
<th></th>
<th>Do</th>
<th>Words</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test</td>
<td>5</td>
<td>Chinese Chinese Chinese Tokyo</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Chinese Chinese Shanghai</td>
<td>c</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Chinese Macao</td>
<td>c</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Tokyo Japan Chinese</td>
<td>j</td>
</tr>
<tr>
<td>Training</td>
<td>1</td>
<td>Chinese Beijing Chinese</td>
<td>c</td>
</tr>
</tbody>
</table>
Two Naive Bayes Models

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

- Model 2: Multinomial = Class conditional unigram
  - One feature $X_i$ for each word pos in document
    - feature's values are all words in dictionary
  - Value of $X_i$ is the word in position $i$
  - 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 for all positions $i,j$, word $w$, and class $c$

Parameter estimation

- Multivariate Bernoulli 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:
  \[ \hat{P}(X_i = w \mid c_j) = \text{fraction of times in which word } w \text{ appears among all words in documents of topic } c_j \]
  - Can create a mega-document for topic $j$ by concatenating all documents in this topic
  - Use frequency of $w$ in mega-document

Which to use for classification?

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

Feature Selection: Why?

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

Feature Selection: How?

- Two ideas:
  - Hypothesis testing statistics:
    - Are we confident that the value of one categorical variable is associated with the value of another
    - $\chi^2$ test
  - 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 $\chi^2$ measures confidence in association, (based on available statistics), while MI measures extent of association (assuming perfect knowledge of probabilities)
Feature Selection: Frequency

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

Feature selection for NB

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

Evaluating Categorization

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

WebKB Experiment (1998)

- Classify webpages from CS departments into:
  - student, faculty, course, project
- Train on ~5,000 hand-labeled webpages
  - 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>Department</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extracted</td>
<td>180</td>
<td>66</td>
<td>246</td>
<td>99</td>
<td>26</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>

NB Model Comparison: WebKB

![NB Model Comparison Graph](image-url)
SpamAssassin

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

Naïve Bayes in Spam Filtering

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

Naïve Bayes Posterior Probabilities

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

Naïve Bayes is Not So Naive

- Very Fast Learning and Testing (basically just count the data)
- Low Storage requirements
- Very good in domains with many equally important features
- More robust to irrelevant features than many learning methods
  - Irrelevant Features cancel each other without affecting results
  - More robust to concept drift (changing class definition over time)
  - Naïve Bayes won 1st and 2nd place in KDD–CUP 97 competition out of 16 systems
  - Goal: Financial services industry direct mail response prediction: Predict if the recipient of mail will actually respond

Resources for today’s lecture

- IIR 13
  - Clear simple explanation of Naïve Bayes
- Open Calais: Automatic Semantic Tagging
  - Free: Use it, they can keep your data, provided by Thompson/Reuters (ex-ClearForest)
  - Weka: A data mining software package that includes an implementation of Naïve Bayes
- Reuters-21578 – the most famous text classification evaluation set
  - Still widely used by lazy people (but now it’s too small for