CS145: INTRODUCTION TO DATA MINING

Text Data: Naïve Bayes

Instructor: Yizhou Sun
yzsun@cs.ucla.edu

December 7, 2017
Methods to be Learnt

<table>
<thead>
<tr>
<th></th>
<th>Vector Data</th>
<th>Set Data</th>
<th>Sequence Data</th>
<th>Text Data</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Classification</strong></td>
<td>Logistic Regression; Decision Tree; KNN; SVM; NN</td>
<td></td>
<td></td>
<td>Naïve Bayes for Text</td>
</tr>
<tr>
<td><strong>Clustering</strong></td>
<td>K-means; hierarchical clustering; DBSCAN; DBSCAN; Mixture Models</td>
<td></td>
<td></td>
<td>PLSA</td>
</tr>
<tr>
<td><strong>Prediction</strong></td>
<td>Linear Regression GLM*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Frequent Pattern Mining</strong></td>
<td>Apriori; FP growth</td>
<td></td>
<td>GSP; PrefixSpan</td>
<td></td>
</tr>
<tr>
<td><strong>Similarity Search</strong></td>
<td></td>
<td></td>
<td></td>
<td>DTW</td>
</tr>
</tbody>
</table>
Naïve Bayes for Text

• Text Data

• Revisit of Multinomial Distribution

• Multinomial Naïve Bayes

• Summary
Text Data

• Word/term
• Document
  • A sequence of words
• Corpus
  • A collection of documents
Text Classification Applications

• Spam detection
From: airak@medicana.com.tr
Subject: Loan Offer
Do you need a personal or business loan urgent that can be process within 2 to 3 working days? Have you been frustrated so many times by your banks and other loan firm and you don't know what to do? Here comes the Good news Deutsche Bank Financial Business and Home Loan is here to offer you any kind of loan you need at an affordable interest rate of 3% If you are interested let us know.

• Sentiment analysis

The Lion King, complete with jaunty songs by Elton John and Tim Rice, is undeniably and fully worthy of its glorious Disney heritage. It is a gorgeous triumph -- one lion in which the studio can take justified pride.

Between traumas, the movie serves up soothingly banal musical numbers (composed by Elton John and Tim Rice) and silly, rambunctious comedy.

July 31, 2013 | Full Review…
Represent a Document

- **Most common way:** Bag-of-Words
  - Ignore the order of words
  - keep the count

For document $d$, $\mathbf{x}_d = (x_{d1}, x_{d2}, ..., x_{dN})$, where $x_{dn}$ is the number of words for $n$th word in the vocabulary

<table>
<thead>
<tr>
<th>Word</th>
<th>c1</th>
<th>c2</th>
<th>c3</th>
<th>c4</th>
<th>c5</th>
<th>m1</th>
<th>m2</th>
<th>m3</th>
<th>m4</th>
</tr>
</thead>
<tbody>
<tr>
<td>human</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>interface</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>computer</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>user</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>system</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>response</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>time</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>EPS</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>survey</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>trees</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>graph</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>minors</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**Vector space model**

m1: The generation of random, binary, unordered trees
m2: The intersection graph of paths in trees
m3: Graph minors IV: Widths of trees and well-quasi-ordering
m4: Graph minors: A survey
More Details

• Represent the doc as a vector where each entry corresponds to a different word and the number at that entry corresponds to how many times that word was present in the document (or some function of it)

  • Number of words is huge
  • Select and use a smaller set of words that are of interest
  • E.g. uninteresting words: ‘and’, ‘the’ ‘at’, ‘is’, etc. These are called stop-words
  • Stemming: remove endings. E.g. ‘learn’, ‘learning’, ‘learnable’, ‘learned’ could be substituted by the single stem ‘learn’
  • Other simplifications can also be invented and used
  • The set of different remaining words is called dictionary or vocabulary. Fix an ordering of the terms in the dictionary so that you can operate them by their index.
  • Can be extended to bi-gram, tri-gram, or so
Limitations of Vector Space Model

• Dimensionality
  • High dimensionality

• Sparseness
  • Most of the entries are zero

• Shallow representation
  • The vector representation does not capture semantic relations between words
Naïve Bayes for Text

- Text Data
- Revisit of Multinomial Distribution
- Multinomial Naïve Bayes
- Summary
Bernoulli and Categorical Distribution

• Bernoulli distribution
  • Discrete distribution that takes two values \{0,1\}
    • \( P(X = 1) = p \) and \( P(X = 0) = 1 - p \)
    • E.g., toss a coin with head and tail

• Categorical distribution
  • Discrete distribution that takes more than two values, i.e., \( x \in \{1, \ldots, K\} \)
    • Also called generalized Bernoulli distribution, multinoulli distribution
    • \( P(X = k) = p_k \) and \( \sum_k p_k = 1 \)
    • E.g., get 1-6 from a dice with 1/6
Binomial and Multinomial Distribution

- **Binomial distribution**
  - Number of successes (i.e., total number of 1’s) by repeating n trials of independent Bernoulli distribution with probability $p$
    - $x$: number of successes
    - $P(X = x) = \binom{n}{x} p^x (1 - p)^{n-x}$

- **Multinomial distribution (multivariate random variable)**
  - Repeat n trials of independent categorical distribution
    - Let $x_k$ be the number of times value $k$ has been observed, note $\sum_k x_k = n$
    - $P(X_1 = x_1, X_2 = x_2, ..., X_K = x_K) = \frac{n!}{x_1!x_2!...x_K!} \prod_k p_k^{x_k}$
Naïve Bayes for Text

• Text Data

• Revisit of Multinomial Distribution

• Multinomial Naïve Bayes

• Summary
Bayes’ Theorem: Basics

- **Bayes’ Theorem:** \( P(h|X) = \frac{P(X|h)P(h)}{P(X)} \)

- **Let** \( X \) **be a data sample (“evidence”)**
- **Let** \( h \) **be a hypothesis** that \( X \) belongs to class \( C \)
- **\( P(h) \) (prior probability):** the probability of hypothesis \( h \)
  - E.g., the probability of “spam” class
- **\( P(X|h) \) (likelihood):** the probability of observing the sample \( X \), given that the hypothesis holds
  - E.g., the probability of an email given it’s a spam
- **\( P(X) \):** marginal probability that sample data is observed
  - \( P(X) = \sum_h P(X|h) P(h) \)
- **\( P(h|X) \), (i.e., posterior probability):** the probability that the hypothesis holds given the observed data sample \( X \)
Classification: Choosing Hypotheses

- **Maximum Likelihood** (maximize the likelihood):
  \[
  h_{ML} = \arg \max_{h \in H} P(X | h)
  \]

- **Maximum a posteriori** (maximize the posterior):
  - Useful observation: it does not depend on the denominator \(P(X)\)
  \[
  h_{MAP} = \arg \max_{h \in H} P(h | X) = \arg \max_{h \in H} P(X | h)P(h)
  \]
Classification by Maximum A Posteriori

• Let D be a training set of tuples and their associated class labels, and each tuple is represented by an p-D attribute vector \( x = (x_1, x_2, ..., x_p) \)

• Suppose there are \( m \) classes \( y \in \{1, 2, ..., m\} \)

• Classification is to derive the maximum posteriori, i.e., the maximal \( P(y=j|x) \)

• This can be derived from Bayes’ theorem

\[
p(y = j|x) = \frac{p(x|y = j)p(y = j)}{p(x)}
\]

• Since \( p(x) \) is constant for all classes, only \( p(x|y)p(y) \) needs to be maximized
Now Come to Text Setting

- A document is represented as a bag of words
  - \( x_d = (x_{d1}, x_{d2}, ..., x_{dN}) \), where \( x_{dn} \) is the number of words for nth word in the vocabulary
- Model \( p(x_d | y) \) for class \( y \)
  - Follow multinomial distribution with parameter vector \( \beta_y = (\beta_{y1}, \beta_{y2}, ..., \beta_{yN}) \), i.e.,
    - \( p(x_d | y) = \frac{(\Sigma_n x_{dn})!}{x_{d1}!x_{d2}!...x_{dN}!} \prod_n \beta_{yn}^{x_{dn}} \)
- Model \( p(y = j) \)
  - Follow categorical distribution with parameter vector \( \pi = (\pi_1, \pi_2, ..., \pi_m) \), i.e.,
    - \( p(y = j) = \pi_j \)
Classification Process Assuming Parameters are Given

- Find $y$ that maximizes $p(y|x_d)$, which is equivalently to maximize

\[
y^* = \arg\max_y p(x_d, y) \\
= \arg\max_y p(x_d|y)p(y)
\]

\[
= \arg\max_y \frac{(\sum_n x_{dn})!}{x_{d1}! x_{d2}! \ldots x_{dN}!} \prod_n \beta_{yn}^{x_{dn}} \times \pi_y
\]

Constant for every class, denoted as $c_d$

\[
= \arg\max_y \prod_n \beta_{yn}^{x_{dn}} \times \pi_y
\]

\[
= \arg\max_y \sum_n x_{dn} \log \beta_{yn} + \log \pi_y
\]
Given a corpus and labels for each document

- $D = \{(x_d, y_d)\}$
- Find the MLE estimators for $\theta = (\beta_1, \beta_2, \ldots, \beta_m, \pi)$

The log likelihood function for the training dataset

$$\log L = \log \prod_d p(x_d, y_d | \theta) = \sum_d \log p(x_d, y_d | \theta)$$

$$= \sum_d \log p(x_d | y_d)p(y_d) = \sum_d x_{dn} \log \beta_{yn} + \log \pi_{y_d} + \log c_d$$

The optimization problem

$$\max_{\theta} \log L$$

s.t.

$$\pi_j \geq 0 \text{ and } \sum_j \pi_j = 1$$

$$\beta_{jn} \geq 0 \text{ and } \sum_n \beta_{jn} = 1 \text{ for all } j$$

Does not involve parameters, can be dropped for optimization purpose
Solve the Optimization Problem

• Use the Lagrange multiplier method

• Solution

\[ \hat{\beta}_{jn} = \frac{\sum_{d:y_d=j} x_{dn}}{\sum_{d:y_d=j} \sum_{n'} x_{dn'}} \]

• \( \sum_{d:y_d=j} x_{dn} \): total count of word n in class j

• \( \sum_{d:y_d=j} \sum_{n'} x_{dn'} \): total count of words in class j

\[ \hat{\pi}_j = \frac{\sum_d 1(y_d=j)}{|D|} \]

• \( 1(y_d=j) \) is the indicator function, which equals to 1 if \( y_d = j \) holds

• \( |D| \): total number of documents
Smoothing

- What if some word n does not appear in some class j in training dataset?

\[ \hat{\beta}_{jn} = \frac{\sum_{d:y_d=j} x_{dn}}{\sum_{d:y_d=j} \sum_{n'} x_{dn'}} = 0 \]

\[ \Rightarrow p(x_d | y = j) \propto \prod_n \beta_{yn}^{x_{dn}} = 0 \]

- But other words may have a strong indication the document belongs to class j

- Solution: add-1 smoothing or Laplacian smoothing

\[ \hat{\beta}_{jn} = \frac{\sum_{d:y_d=j} x_{dn} + 1}{\sum_{d:y_d=j} \sum_{n'} x_{dn'} + N} \]

- \( N \): total number of words in the vocabulary
- Check: \( \sum_n \hat{\beta}_{jn} = 1? \)
Example

• Data:

<table>
<thead>
<tr>
<th>Doc</th>
<th>Words</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Chinese Beijing Chinese</td>
<td>c</td>
</tr>
<tr>
<td>2</td>
<td>Chinese Chinese Shanghai</td>
<td>c</td>
</tr>
<tr>
<td>3</td>
<td>Chinese Macao</td>
<td>c</td>
</tr>
<tr>
<td>4</td>
<td>Tokyo Japan Chinese</td>
<td>j</td>
</tr>
<tr>
<td>Test</td>
<td>Chinese Chinese Chinese Tokyo Japan</td>
<td>?</td>
</tr>
</tbody>
</table>

• Vocabulary:

<table>
<thead>
<tr>
<th>Index</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word</td>
<td>Chinese</td>
<td>Beijing</td>
<td>Shanghai</td>
<td>Macao</td>
<td>Tokyo</td>
<td>Japan</td>
</tr>
</tbody>
</table>

• Learned parameters (with smoothing):

\[ \hat{\beta}_{c1} = \frac{5 + 1}{8 + 6} = \frac{3}{7} \]
\[ \hat{\beta}_{c2} = \frac{8 + 6}{1 + 1} = \frac{1}{7} \]
\[ \hat{\beta}_{c3} = \frac{8 + 6}{1 + 1} = \frac{1}{7} \]
\[ \hat{\beta}_{c4} = \frac{8 + 6}{0 + 1} = \frac{1}{7} \]
\[ \hat{\beta}_{c5} = \frac{8 + 6}{0 + 1} = \frac{14}{1} \]
\[ \hat{\beta}_{c6} = \frac{8 + 6}{0 + 1} = \frac{14}{1} \]
\[ \hat{\beta}_{j1} = \frac{1 + 1}{3 + 6} = \frac{2}{9} \]
\[ \hat{\beta}_{j2} = \frac{3 + 6}{0 + 1} = \frac{1}{9} \]
\[ \hat{\beta}_{j3} = \frac{3 + 6}{0 + 1} = \frac{1}{9} \]
\[ \hat{\beta}_{j4} = \frac{3 + 6}{1 + 1} = \frac{2}{9} \]
\[ \hat{\beta}_{j5} = \frac{3 + 6}{1 + 1} = \frac{2}{9} \]
\[ \hat{\beta}_{j6} = \frac{3 + 6}{3 + 6} = \frac{2}{9} \]

\[ \hat{\pi}_c = \frac{3}{4} \]
\[ \hat{\pi}_j = \frac{1}{4} \]
Example (Continued)

• **Classification stage**
  
  • For the test document \( d=5 \), compute
    
    \[
    p(y = c | x_5) \propto p(y = c) \times \prod_n \beta_{cn}^{x_{5n}} = \frac{3}{4} \times \left(\frac{3}{7}\right)^3 \times \left(\frac{1}{14}\right) \times \left(\frac{1}{14}\right) \approx 0.0003
    \]

    \[
    p(y = j | x_5) \propto p(y = j) \times \prod_n \beta_{jn}^{x_{5n}} = \frac{1}{4} \times \left(\frac{2}{9}\right)^3 \times \left(\frac{2}{9}\right) \times \left(\frac{2}{9}\right) \approx 0.0001
    \]

  • **Conclusion:** \( x_5 \) should be classified into \( c \) class
A More General Naïve Bayes Framework

• Let D be a training set of tuples and their associated class labels, and each tuple is represented by an p-D attribute vector \( \mathbf{x} = (x_1, x_2, \ldots, x_p) \)

• Suppose there are \( m \) classes \( y \in \{1, 2, \ldots, m\} \)

• Goal: Find \( y \)

\[
\max_y P(y|x) = P(y, x)/P(x) \propto P(x|y)P(y)
\]

• A simplified assumption: attributes are conditionally independent given the class (class conditional independency):

\[
p(x|y) = \prod_k p(x_k|y)
\]

\( p(x_k|y) \) can follow any distribution, e.g., Gaussian, Bernoulli, categorical, ...
Generative Model vs. Discriminative Model

• Generative model
  • \textit{model joint probability} $p(x, y)$
  • E.g., naïve Bayes

• Discriminative model
  • \textit{model conditional probability} $p(y|x)$
  • E.g., logistic regression
Naïve Bayes for Text

- Text Data

- Revisit of Multinomial Distribution

- Multinomial Naïve Bayes

- Summary
Summary

- Text data
  - Bag of words representation
- Naïve Bayes for Text
  - Multinomial naïve Bayes