CS 6120/CS4120: Natural Language Processing
Instructor: Prof. Lu Wang
College of Computer and Information Science
Northeastern University
Webpage: www.ccs.neu.edu/home/luwang

Projects
• Sample Projects from other NLP courses:

Project Proposal
• Introduction: the problem has to be well-defined. What are the input and output. Why this is an important problem to study.
• Related work: put your work in context. Describe what has been done in previous work on the same or related subject. And why you propose to do here is novel and different.
• Datasets: what data do you want to use? What is the size of it? What information is contained? Why is it suitable for your task?
• Methodology (optional): what models do you want to use? You may change the model as the project goes, but you may want to indicate some type of models that might be suitable for your problem. Is it a supervised learning problem or unsupervised? What classifiers can you start with? Are you making improvements?
• Evaluation: what metrics do you want to use for evaluating your models?

Today’s Outline
• Text Categorization
• Naïve Bayes
• Part-of-speech tagging
• Hidden Markov Models

Is this spam?

From: Information Desk <info@eurowinelottery.com>
Subject: EU/Commonwealth Lottery Promotions

Your email address was selected to claim the sum of $500,000.00 in the 2013 European lottery.
To claim your prize, please contact our agent in Lagos, Nigeria.
Contact person: Mr. Marshall Ellis e-mail: marshall@513@lbe.com
Phone: +2348063654742
Congratulations!
Vincent Kilrens (Coordinator)

[Some slides are modified from lectures of Dan Jurafsky, Chris Manning, and Ray Mooney]
Who wrote which Federalist papers?

- 1787-8: anonymous essays try to convince New York to ratify U.S Constitution: Jay, Madison, Hamilton.
- Authorship of 12 of the letters in dispute
- 1963: solved by Mosteller and Wallace using Bayesian methods

Male or female author?

1. By 1925 present-day Vietnam was divided into three parts under French colonial rule. The southern region embracing Saigon and the Mekong delta was the colony of Cochin-China; the central area with its imperial capital at Hue was the protectorate of Annam...
2. Clara never failed to be astonished by the extraordinary felicity of her own name. She found it hard to trust herself to the mercy of fate, which had managed over the years to convert her greatest shame into one of her greatest assets...

Positive or negative movie review?

- unbelievably disappointing
- Full of zany characters and richly applied satire, and some great plot twists
- this is the greatest screwball comedy ever filmed
- It was pathetic. The worst part about it was the boxing scenes.

What is the subject of this article?

**MEDLINE Article**

**MeSH Subject Category Hierarchy**

- Antagonists and Inhibitors
- Blood Supply
- Chemistry
- Drug Therapy
- Embryology
- Epidemiology
- ...

Text Classification

- Assigning subject categories, topics, or genres
- Spam detection
- Authorship identification
- Age/gender identification
- Language Identification
- Sentiment analysis
- ...

Text Classification: definition

**Input:**
- a document \(d\)
- a fixed set of classes \(C = \{c_1, c_2, ..., c_J\}\)

**Output:** a predicted class \(c \in C\)
Classification Methods: Hand-coded rules
• Rules based on combinations of words or other features
  • spam: black-list address OR ("dollars" AND "have been selected")
• Accuracy can be high
• If rules carefully refined by expert
• But building and maintaining these rules is expensive

Classification Methods: Simple ("Naïve Bayes Intuition"

Hand Classification Methods:
• Any kind of classifier
  • Naïve Bayes
  • Logistic regression
  • Support-vector machines
  • k-Nearest Neighbors
  • ...

Naïve Bayes Classifier

Naïve Bayes Intuition
• Simple ("naive") classification method based on Bayes rule
• Relies on very simple representation of document
• Bag of words

The Bag of Words Representation

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!

Classification Methods: Supervised Machine Learning
• Input:
  • A document $d$
  • a fixed set of classes $C = \{c_1, c_2, ..., c_n\}$
  • A training set of $m$ hand-labeled documents $(d_1, c_1), ..., (d_m, c_m)$
• Output:
  • a learned classifier $y: d \rightarrow c$

Supervised Machine Learning Classification Methods:
• Logistic regression
• Nearest neighbors
• Support-vector machines
• Bag of words

18/9/17
The bag of words representation

\[ \gamma(\ldots) = c \]

Bayes’ Rule Applied to Documents and Classes

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

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

Naïve Bayes Classifier (I)

\[
c_{MAP} = \arg \max_{c \in C} P(c | d)
\]

MAP is “maximum a posteriori” = most likely class

\[
= \arg \max_{c \in C} \frac{P(d | c)P(c)}{P(d)}
\]

Bayes Rule

\[
= \arg \max_{c \in C} P(d | c)P(c)
\]

Dropping the denominator

Naïve Bayes Classifier (II)

\[
c_{MAP} = \arg \max_{c \in C} P(d | c)P(c)
\]

Document \( d \) represented as features \( x_1, x_2, \ldots, x_n \)

\[
= \arg \max_{c \in C} P(x_1, x_2, \ldots, x_n | c)P(c)
\]

Naïve Bayes Classifier (IV)

\[
c_{MAP} = \arg \max_{c \in C} P(x_1, x_2, \ldots, x_n | c)P(c)
\]

Multinomial Naïve Bayes Independence Assumptions

\[
P(x_1, x_2, \ldots, x_n | c)
\]

- Bag of Words assumption: Assume position doesn’t matter
- Conditional Independence: Assume the feature probabilities \( P(x_i | c) \) are independent given the class \( c \).

\[
P(x_1, \ldots, x_n | c) = P(x_1 | c) \cdot P(x_2 | c) \cdot P(x_3 | c) \cdot \ldots \cdot P(x_n | c)
\]
Multinomial Naïve Bayes Classifier

\[ c_{\text{MAP}} = \underset{c \in C}{\arg \max} \ P(x_1, x_2, \ldots, x_n | c) P(c) \]

\[ c_{\text{NB}} = \underset{c \in C}{\arg \max} \ P(c) \prod_{x \in X} P(x | c) \]

Applying Multinomial Naïve Bayes Classifiers to Text Classification

positions ← all word positions in test document

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

Learning for Naïve Bayes Model

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

\[ \hat{P}(c_j) = \frac{\text{doccount}(C = c_j)}{N_{\text{doc}}} \]

\[ \hat{P}(w_i | c_j) = \frac{\text{count}(w_i, c_j)}{\sum_{w \in V} \text{count}(w, c_j)} \]

Learning the Multinomial Naïve Bayes Model

Parameter estimation

\[ \hat{P}(w_i | c_j) = \frac{\text{count}(w_i, c_j)}{\sum_{w \in V} \text{count}(w, c_j)} \]

fraction of times word \( w_i \) appears among all words in documents of topic \( c_j \)

• Create mega-document for topic \( j \) by concatenating all docs in this topic
  • Use frequency of \( w \) in mega-document

Problem with Maximum Likelihood

• What if we have seen no training documents with the word \( \text{fantastic} \) and classified in the topic \( \text{positive (thumbs-up)} \)?

\[ \hat{P}(\text{"fantastic" | positive}) = \frac{\text{count(\"fantastic\", positive)}}{\sum_{w \in \text{positive}} \text{count}(w, \text{positive})} \rightarrow 0 \]

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

\[ c_{\text{MAP}} = \underset{c}{\arg \max} \ \hat{P}(c) \prod_{i} \hat{P}(x_i | c) \]
Laplace (add-1) smoothing for Naïve Bayes

\[ \hat{P}(w_i | c) = \frac{\text{count}(w_i, c) + 1}{\sum_{w \in V} \text{count}(w, c) + 1} \]

Multinomial Naïve Bayes: Learning

- From training corpus, extract vocabulary
- Calculate \(P(c)\) terms
- For each \(c_i\) in \(C\) do
  - \(\text{docs}_i\) ← all docs with class \(= c_i\)
  - \(P(c_i) = \frac{|\text{docs}_i|}{\text{Total # documents}}\)
- Calculate \(P(w_i | c_j)\) terms
- Text \(e\) ← single doc containing all \(\text{docs}_i\)
- For each word \(w_k\) in vocabulary
  - \(n_{ik}\) ← # of occurrences of \(w_k\) in Text
  - \(P(w_i | c_j) = \frac{n_{ik} + \alpha}{n + \alpha |\text{Vocabulary}|}\)

Naïve Bayes and Language Modeling

- Naïve 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
  - Naïve Bayes has an important similarity to language modeling.

Each class = a unigram language model

- Assigning each word: \(P(\text{word} | c)\)
- Assigning each sentence: \(P(s | c) = \prod P(\text{word} | c)\)

Class

<table>
<thead>
<tr>
<th>pos</th>
<th>i</th>
<th>love</th>
<th>this</th>
<th>fun</th>
<th>film</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.1</td>
<td>0.05</td>
<td>0.01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td>0.05</td>
<td>0.001</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>0.1</td>
<td>0.05</td>
<td>0.1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\(P(s | pos) = 0.0000005\)

Naïve Bayes as a Language Model

- Which class assigns the higher probability to \(s\)?

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
\text{Model pos} & i & \text{love} & \text{this} & \text{fun} & \text{film} \\
0.1 & 0.1 & 0.001 & 0.01 & 0.05 & 0.1 \\
0.1 & \text{love} & 0.01 & 0.001 & 0.01 & 0.005 & 0.1 \\
0.05 & \text{fun} & 0.005 & \text{fun} & \text{film} & \\
0.1 & \text{film} & \text{film} & \text{film} & \text{film} & \\
\hline
\end{array}
\]

\(P(s | \text{pos}) > P(s | \text{neg})\)

An Example
Choosing a class:

\[
P(c|d5) = \frac{1}{4} \times (2/9) = 0.0001
\]

\[
P(j|d5) = \frac{3}{4} \times (3/7) = 0.0026
\]

\[
P(c|d5) = \frac{1}{4} \times (2/9) = 0.0001
\]

\[
P(j|d5) = \frac{3}{4} \times (3/7) = 0.0026
\]

\[
P(w|c) = \frac{c}{N} = \frac{1}{7}
\]

\[
P(w|j) = \frac{1}{7}
\]

Summary: Naive Bayes is Not So Naive

- Very Fast, low storage requirements
- Robust to Irrelevant Features
  - Irrelevant Features cancel each other without affecting results
- Very good in domains with many equally important features
- Optimal if the independence assumptions hold: If assumed independence is correct, then it is the Bayes Optimal Classifier for problem
- A good dependable baseline for text classification
  - But we will see other classifiers that give better accuracy

---

### Evaluation

<table>
<thead>
<tr>
<th></th>
<th>correct</th>
<th>not correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>selected</td>
<td>tp</td>
<td>fp</td>
</tr>
<tr>
<td>not selected</td>
<td>fn</td>
<td>tn</td>
</tr>
</tbody>
</table>

### The 2-by-2 contingency table

<table>
<thead>
<tr>
<th></th>
<th>correct</th>
<th>not correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>selected</td>
<td>tp</td>
<td>fp</td>
</tr>
<tr>
<td>not selected</td>
<td>fn</td>
<td>tn</td>
</tr>
</tbody>
</table>

### Precision and recall

- **Precision**: % of selected items that are correct, \( tp/(tp+fp) \)
- **Recall**: % of correct items that are selected, \( tp/(tp+fn) \)

<table>
<thead>
<tr>
<th></th>
<th>correct</th>
<th>not correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>selected</td>
<td>tp</td>
<td>fp</td>
</tr>
<tr>
<td>not selected</td>
<td>fn</td>
<td>tn</td>
</tr>
</tbody>
</table>

### A combined measure: F

- A combined measure that assesses the P/R tradeoff is F measure (weighted harmonic mean):

\[
F = \frac{1}{\frac{1}{P} + (1-\alpha) \frac{1}{R}} = \frac{\beta^P + 1}{\beta^P + \alpha^R}
\]

- The harmonic mean is a very conservative average
- People usually use balanced F1 measure
  - i.e., with \( \beta = 1 \) (that is, \( \alpha = 0 \)):

\[
F = 2PR/(P+R)
\]
More Than Two Classes: Sets of binary classifiers

- One-of or multinomial classification
  - Classes are mutually exclusive: each document in exactly one class
- For each class \( c \in C \)
  - Build a classifier \( y_c \) to distinguish \( c \) from all other classes \( c' \in C \)
- Given test doc \( d \)
  - Evaluate it for membership in each class using \( y_c \)
  - \( d \) belongs to the one class with maximum score

Evaluation:
Classic Reuters-21578 Data Set

- Most (over)used data set, 21,578 docs
- 9,603 training, 3,299 test articles
- 118 categories
  - An article can be in more than one category
  - Learn 118 binary category distinctions
- Average document (with at least one category) has 1.24 classes
- Only about 10 out of 118 categories are large

Common categories

- Finance
- Agriculture
- Trade
- Interest
- Energy
- Weather
- Sports
- Health
- Entertainment

Confusion matrix \( c \)

- For each pair of classes \( c_i, c_j \), how many documents from \( c_i \) were incorrectly assigned to \( c_j \)
- \( \sum_{j=0}^{90} \) 90 what documents incorrectly assigned to poultry

<table>
<thead>
<tr>
<th>True UK</th>
<th>Assigned poultry</th>
<th>Assigned wheat</th>
<th>Assigned coffee</th>
<th>Assigned interest</th>
<th>Assigned trade</th>
</tr>
</thead>
<tbody>
<tr>
<td>95</td>
<td>1</td>
<td>13</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>90</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>34</td>
<td>3</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>-</td>
<td>1</td>
<td>2</td>
<td>13</td>
<td>26</td>
<td>5</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>14</td>
<td>5</td>
<td>10</td>
<td>2</td>
</tr>
</tbody>
</table>
Per class evaluation measures

**Recall:**
Fraction of docs in class \( i \) classified correctly:
\[
\frac{c_{yi}}{\sum_{j} c_{ji}}
\]

**Precision:**
Fraction of docs assigned class \( i \) that are actually about class \( i \):
\[
\frac{c_{yi}}{\sum_{j} c_{ij}}
\]

**Accuracy:** (1 - error rate)
Fraction of docs classified correctly:
\[
\frac{c_{ii}}{\sum_{j} \sum_{i} c_{ij}}
\]

---

Micro- vs. Macro-Averaging

- If we have more than one class, how do we combine multiple performance measures into one quantity?
- **Macroaveraging:** Compute performance for each class, then average.
- **Microaveraging:** Collect decisions for all classes, compute contingency table, evaluate.

---

Micro- vs. Macro-Averaging: Example

<table>
<thead>
<tr>
<th>Class 1</th>
<th>Class 2</th>
<th>Micro Ave. Table</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Truth</td>
<td></td>
</tr>
<tr>
<td></td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Classifier: yes</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Classifier: no</td>
<td>10</td>
<td>170</td>
</tr>
<tr>
<td></td>
<td>Truth</td>
<td></td>
</tr>
<tr>
<td></td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Classifier: yes</td>
<td>90</td>
<td>10</td>
</tr>
<tr>
<td>Classifier: no</td>
<td>10</td>
<td>890</td>
</tr>
</tbody>
</table>

- Macroaveraged precision: \((0.5 + 0.9)/2 = 0.7\)
- Microaveraged precision: \(100/120 = .83\)
- Microaveraged score is dominated by score on common classes

---

Development Test Sets and Cross-validation

- **Metric:** P/R/F1 or Accuracy
- **Unseen test set**
  - avoid overfitting ('tuning to the test set')
  - more conservative estimate of performance
- **Cross-validation over multiple splits**
  - Handle sampling errors from different datasets
  - Pool results over each split
  - Compute pooled dev set performance