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

Project Proposal
• Length: 1 page (or more if necessary).
  • Single space if MS word is used. Or you can choose latex templates, e.g. https://www.acm.org/publications/proceedings/template or http://icml.cc/2015/?page_id=151.
• 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 what 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: 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? You don’t have to be crystal clear on this section, but it can be used to indicate the direction that your project goes to.
• Evaluation: what metrics do you want to use for evaluating your models?

Sample proposal and reports
• http://www.ccs.neu.edu/home/luwang/courses/cs6120_sp2019/cs6120_sp2019.html
• Sample projects from Stanford NLP course
  • http://web.stanford.edu/class/cs224n
• Finding teammates on Piazza!

Outline
• Text Categorization/Classification
• Naïve Bayes
• Evaluation

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.

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...

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 $C = \{c_1, c_2, \ldots, c_J\}$
- 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:

Supervised Machine Learning

- Input:
  - a document $d$
  - a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$
  - A training set of $m$ hand-labeled documents $(d_1, y_1), \ldots, (d_m, y_m)$, $y_i$ is in $C$
- Output:
  - a learned classifier $\gamma : d \rightarrow c$

Classification Methods:

Supervised Machine Learning

- Any kind of classifier
  - Naive Bayes
  - Logistic regression
  - Support-vector machines
  - k-Nearest Neighbors
  - ...

Outline

- Text Categorization/Classification
  - Naive Bayes
  - Evaluation
Naïve Bayes Classifier

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

The Bag of Words Representation

The bag of words representation

\[ Y(d) = \begin{array}{c|c}
\text{seen} & 2 \\
\text{sweet} & 1 \\
\text{whimsical} & 1 \\
\text{recommend} & 1 \\
\text{happy} & 1 \\
... & ...
\end{array} \]

Bayes’ Rule Applied to Documents and Classes

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

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

Naïve Bayes Classifier (I)

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

MAP is "maximum a posteriori" = most likely class

Bayes Rule

Dropping the denominator

...
Naïve Bayes Classifier (I)

$$c_{MAP} = \arg \max_{c \in C} P(c \mid d)$$

MAP is "maximum a posteriori" = most likely class

$$= \arg \max_{c \in C} \frac{P(d \mid c)P(c)}{P(d)}$$

Bayes Rule

$$= \arg \max_{c \in C} P(d \mid c)P(c)$$

Dropping the denominator

Why we can do this?

Naïve Bayes Classifier (II)

$$c_{MAP} = \arg \max_{c \in C} P(d \mid c)P(c)$$

MAP = argmax

Document d represented as features x1..xn

Naïve Bayes Classifier (IV)

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

O(|X|*|C|) parameters $|X|$ represents the maximum number of possible values for xi

P(x1, x2, ..., xn | 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.

$$P(x_1, x_2, \ldots, x_n | c) = P(x_1 | c) \cdot P(x_2 | c) \cdot P(x_3 | c) \cdot \ldots \cdot P(x_n | c)$$

Multinomial Naïve Bayes Classifier

Naïve Bayes Classifiers to Text Classification

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

$$c_{NB} = \arg \max_{c \in C} P(c) \prod_{x \in X} P(x \mid c)$$

positions ← all word positions in test document

Applying Multinomial Naïve Bayes Classifiers to Text Classification

$$c_{NB} = \arg \max_{c_i \in C} P(c_j) \prod_{i \in \text{positions}} P(x_i \mid c_j)$$
Learning for Naive Bayes Model

Learning the Multinomial Naive Bayes Model

- First attempt: maximum likelihood estimates
  - simply use the frequencies in the data
  \[ \hat{P}(c_j) = \frac{\text{doc}\text{count}(C = c_j)}{N_{\text{doc}}} \]
  \[ \hat{P}(w_i | c_j) = \frac{\text{count}(w_i, c_j)}{\sum_{w \in V} \text{count}(w, c_j)} \]

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 class \( c_j \)

Problem with Maximum Likelihood

- What if we have seen no training documents with the word \textit{fantastic} and classified in the topic \textit{positive (thumbs-up)}?
  \[ \hat{P}(\text{"fantastic" | positive}) = \frac{\text{count}(\text{"fantastic", positive})}{\sum_{w \in V} \text{count}(w, \text{positive})} = 0 \]

- Zero probabilities cannot be conditioned away, no matter the other evidence!
  \[ c_{\text{max}} = \arg \max \hat{P}(c) \prod_{X \in \text{Text}} \hat{P}(X_i | c) \]

Laplace (add-1) smoothing for Naive Bayes

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

Multinomial Naive Bayes: Learning

- From training corpus, extract Vocabulary
- Calculate \( P(c_j) \) terms
  - For each \( c_j \) in \( C \) do
    - \( \text{doc}_j \) ← all docs with class \( = c_j \)
    - \( P(c_j) \) ← \( \frac{| \text{doc}_j |}{| \text{total # documents} |} \)
- Calculate \( P(w_i | c_j) \) terms
  - For each word \( w_i \) in Vocabulary
    - \( n_i \) ← # of occurrences of \( w_i \) in \( \text{Text} \),
    - \( P(w_i | c_j) \) ← \( \frac{n_i + \alpha}{n + \alpha | \text{Vocabulary} |} \)
**Multinomial Naive Bayes: Learning**

- From training corpus, extract Vocabulary
- Calculate $P(c)$ terms
  - For each $c_j$ in $C$ do
    - $P(c)_j \leftarrow \frac{\text{docs}_j}{\text{total \# \ documents}}$
- Calculate $P(w_k | c_j)$ terms
  - For each word $w_k$ in Vocabulary
    - $n_w \leftarrow \# \ of \ occurrences \ of \ w_k \ in \ Text$
    - $P(w_k | c_j) \leftarrow \frac{n_w + \alpha}{n_j + \alpha | \text{Vocabulary}}$

  A more general form: add-$\alpha$ smoothing!

**Naive 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: $P(\text{word} \mid c)$
- Assigning each sentence: $P(s \mid c) = \prod P(\text{word} \mid c)$

**Naïve Bayes as a Language Model**

- Which class assigns the higher probability to $s$?

<table>
<thead>
<tr>
<th>Class</th>
<th>Model pos</th>
<th>Model neg</th>
</tr>
</thead>
<tbody>
<tr>
<td>pos</td>
<td>0.1</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>love</td>
<td>love</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>0.001</td>
</tr>
<tr>
<td></td>
<td>this</td>
<td>this</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td></td>
<td>film</td>
<td>film</td>
</tr>
<tr>
<td></td>
<td>P(s \mid pos) = 0.0000005</td>
<td>P(s \mid neg) &gt; P(s \mid pos)</td>
</tr>
</tbody>
</table>

**An Example**

<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>c</td>
</tr>
<tr>
<td>5</td>
<td>Chinese Chinese Tokyo Japanese</td>
<td>?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Training</th>
<th>Chinese Beijing Chinese</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Chinese Chinese Shanghai</td>
<td>c</td>
</tr>
<tr>
<td>2</td>
<td>Chinese Macao</td>
<td>c</td>
</tr>
<tr>
<td>3</td>
<td>Tokyo Japan Chinese</td>
<td>c</td>
</tr>
<tr>
<td>4</td>
<td>Chinese Chinese Tokyo Japanese</td>
<td>?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Priors:</th>
<th>$P(c)$</th>
<th>$P(w \mid c)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P(c) = \frac{1}{4}$</td>
<td>$P(w \mid c) = \frac{1}{4}$</td>
<td></td>
</tr>
</tbody>
</table>

Choosing a class: $P(c \mid \text{docs}) = \frac{3}{4} \times \frac{3}{7} \times \frac{1}{14} \times \frac{1}{14} = 0.0003$

Conditional Probabilities:

- $P(\text{Chinese} \mid c) = \frac{1}{4}$
- $P(\text{Tokyo} \mid c) = \frac{1}{4}$
- $P(\text{Japan} \mid c) = \frac{1}{4}$
- $P(\text{Chinese} \mid \text{docs}) = \frac{1}{14} \times \frac{1}{14} = 0.0002$
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

Outline

- Text Categorization/Classification
- Naive Bayes
  - Evaluation

Evaluation

The 2-by-2 contingency table (or confusion matrix)

<table>
<thead>
<tr>
<th></th>
<th>correct</th>
<th>not correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>selected</td>
<td>tp (true positive)</td>
<td>fp (false positive)</td>
</tr>
<tr>
<td>not selected</td>
<td>fn (false negative)</td>
<td>tn (true negative)</td>
</tr>
</tbody>
</table>

For example,
- Which set of documents are related to the topic of NLP?
- Which set of documents are written by Shakespeare?

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)
A combined measure: F-measure or F-score

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

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

- People usually use balanced F1 measure
  - i.e., \( \alpha = \frac{1}{2} \), \( F = 2PR/(P+R) \)

Text Classification Evaluation

More Than Two Classes:
Sets of binary classifiers

- Dealing with any-of or multivalue classification
  - A document can belong to 0, 1, or >1 classes.

- For each class \( c \in C \)
  - Build a classifier \( \gamma_c \) to distinguish \( c \) from all other classes \( c' \in C \)
  - Given test doc \( d \)
    - Evaluate it for membership in each class using each \( \gamma_c \)
      - \( d \) belongs to any class for which \( \gamma_c \) returns true

Confusion matrix \( c \)

- For each pair of classes \( <c_1, c_2> \) how many documents from \( c_1 \) were incorrectly assigned to \( c_2 \)?
  - \( c_{12} \): 90 wheat documents incorrectly assigned to poultry

<table>
<thead>
<tr>
<th>Docs in test set</th>
<th>Assigned 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>True UK</td>
<td>95</td>
<td>1</td>
<td>13</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>True poultry</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>True wheat</td>
<td>10</td>
<td>90</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>True coffee</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>34</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>True interest</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>13</td>
<td>26</td>
<td>5</td>
</tr>
<tr>
<td>True trade</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>14</td>
<td>5</td>
<td>10</td>
</tr>
</tbody>
</table>

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 \( \gamma_c \) to distinguish \( c \) from all other classes \( c' \in C \)
  - Given test doc \( d \)
    - Evaluate it for membership in each class using each \( \gamma_c \)
    - \( d \) belongs to the one class with maximum score

Per class evaluation measures

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

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

- **Accuracy:** (1 - error rate)
  - Fraction of docs classified correctly:
    \[ \sum_{j} c_{ij} \]
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>Truth: yes</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Truth: no</td>
<td>10</td>
<td>970</td>
</tr>
</tbody>
</table>

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

- Macroaveraged precision: \( \frac{0.5 + 0.9}{2} = 0.7 \)
- Microaveraged precision: \( \frac{100}{120} = 0.83 \)

### 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

---

**Training set**

<table>
<thead>
<tr>
<th>Training</th>
<th>Development/tuning/held-out Set</th>
<th>Test Set</th>
</tr>
</thead>
</table>

---

**Test Set**

- Training set
  - Dev Test
- Training Set
  - Dev Test
- Test Set