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

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 \( C = \{c_1, c_2, ..., c_l\} \)

- **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 \(\gamma : d \rightarrow c\)

Outline
- Text Categorization/Classification
  - Naïve Bayes
  - Evaluation

Naïve Bayes Classifier

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

Naïve Bayes Classifier (I)

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

Naïve Bayes Classifier (II)

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

Bayes’ Rule Applied to Documents and Classes

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

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

Why we can do this?
Naïve Bayes Classifier (IV)

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

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

\[ P(x_1, x_2, \ldots, x_n \mid c) = P(x_1 \mid c) \cdot P(x_2 \mid c) \cdot P(x_3 \mid c) \cdot \ldots \cdot P(x_n \mid c) \]

Multinomial Naïve Bayes Classifier

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

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

Learning for Naïve Bayes Model

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

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

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

Learning the Multinomial Naïve Bayes Model
Parameter estimation

\[ \hat{P}(w_j \mid c_i) = \frac{\text{count}(w_j, c_i)}{\sum_{w \in V} \text{count}(w, c_i)} \]

fraction of times word \( w \) appears among all words in documents of class \( c_i \)

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)?
  \[ \hat{P}(\text{fantastic} \mid \text{positive}) = \frac{\text{count}(\text{fantastic}, \text{positive})}{\sum_{w \in V} \text{count}(w, \text{positive})} = 0 \]

- Zero probabilities cannot be conditioned away, no matter the other evidence!
  \[ c_{MAP} = \arg \max_c \hat{P}(c) \prod \hat{P}(x_j \mid c) \]

Laplace (add-1) smoothing for Naïve Bayes

\[ \hat{P}(w_j \mid c_i) = \frac{\text{count}(w_j, c_i) + 1}{\sum_{w \in V} \text{count}(w, c_i) + 1} \]

\[ = \frac{\text{count}(w_j, c_i) + 1}{\sum_{w \in V} \text{count}(w, c_i) + |V|} \]

Multinomial Naïve Bayes: Learning

- From training corpus, extract Vocabulary

  - Calculate \( P(c) \) terms
    - For each \( c \in C \) do
      - \( \text{docs}_c \leftarrow \) all docs with class \( c \)

    \[ P(c) = \frac{|\text{docs}_c|}{\text{total # documents}} \]

  - Text, \( e \) — single doc containing all \( c \)

    \[ P(c) = \frac{|\text{docs}_c|}{\text{total # documents}} \]

    \[ P(w_j \mid c) = \frac{n_j + \alpha}{n + \alpha |Vocabulary|} \]

    A more general form: add-\( \alpha \) smoothing!

- Calculate \( P(w_j \mid c) \) terms

  - For each word \( w_j \) in Vocabulary

    \[ n_j \leftarrow \text{# of occurrences of } w_j \text{ in Text} \]

    \[ P(c) = \frac{|\text{docs}_c|}{\text{total # documents}} \]

    \[ P(w_j \mid c) = \frac{n_j + \alpha}{n + \alpha |Vocabulary|} \]

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

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

\[ P(\text{sentence} \mid \text{pos}) = 0.0000005 \]

Naive Bayes as a Language Model

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

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

<table>
<thead>
<tr>
<th>Model neg</th>
<th>0.2</th>
<th>love</th>
<th>0.001</th>
<th>this</th>
<th>0.001</th>
<th>fun</th>
<th>film</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.1</td>
<td>0.1</td>
<td>0.01</td>
<td>0.05</td>
<td>0.1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ P(s \mid \text{pos}) > P(s \mid \text{neg}) \]

An Example

\[ P(c \mid d_5) = \frac{1}{4} \times \frac{2}{9} \times \frac{2}{9} \approx 0.0001 \]

Choosing a class:

\[ P(c \mid d_5) = \frac{3}{4} \times \frac{3}{7} \times \frac{1}{14} \times \frac{1}{14} \approx 0.0003 \]

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
- Naïve 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 ( (\text{true positive}) )</td>
<td>fp ( (\text{false positive}) )</td>
</tr>
<tr>
<td>not selected</td>
<td>fn ( (\text{false negative}) )</td>
<td>tn ( (\text{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, \( \frac{tp}{tp+fp} \)
- **Recall**: \( \% \) of correct items that are selected, \( \frac{tp}{tp+fn} \)

<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-measure or F-score

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

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

- 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 \(c \in \mathcal{C}\)
  - Build a classifier \(\gamma_c\) to distinguish \(c\) from all other classes \(c' \in \mathcal{C}\)

- 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

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 \(c \in \mathcal{C}\)
  - Build a classifier \(\gamma_c\) to distinguish \(c\) from all other classes \(c' \in \mathcal{C}\)

- Given test doc \(d\),
  - Evaluate it for membership in each class using each \(\gamma_c\)
  - \(d\) belongs to the one class with maximum score

Confusion matrix \(c\)

- For each pair of classes \(<c_i, c_j>\) how many documents from \(c_i\) were incorrectly assigned to \(c_j\)?

<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>34</td>
<td>3</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>True interest</td>
<td>0</td>
<td>1</td>
<td>13</td>
<td>26</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>True trade</td>
<td>0</td>
<td>0</td>
<td>14</td>
<td>5</td>
<td>10</td>
<td>0</td>
</tr>
</tbody>
</table>

Per class evaluation measures

- **Recall**: Fraction of docs in class \(i\) classified correctly: \(\frac{c_{ii}}{\sum_j c_{ij}}\)
- **Precision**: Fraction of docs assigned class \(i\) that are actually about class \(i\): \(\frac{\sum_i c_{ii}}{\sum_j c_{ij}}\)
- **Accuracy**: (1 - error rate) Fraction of docs classified correctly: \(\frac{\sum_i \sum_j c_{ij}}{\sum_j c_{ij}}\)

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</td>
<td></td>
<td></td>
</tr>
<tr>
<td>yes</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>no</td>
<td>50</td>
<td>0</td>
</tr>
<tr>
<td>yes</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>no</td>
<td>90</td>
<td>10</td>
</tr>
<tr>
<td>yes</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>no</td>
<td>50</td>
<td>0</td>
</tr>
</tbody>
</table>

- 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></td>
<td></td>
</tr>
<tr>
<td>Truth: yes</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Truth: yes</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Truth: no</td>
<td>20</td>
<td>180</td>
</tr>
<tr>
<td>Truth: no</td>
<td>20</td>
<td>180</td>
</tr>
</tbody>
</table>

- Macroaveraged precision: \((0.5 + 0.9)/2 = 0.7\)
- Microaveraged precision: \(100/120 = .83\)

### Development Test Sets and Cross-validation

<table>
<thead>
<tr>
<th>Metric: P/R/F1 or Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Unseen test set</strong></td>
</tr>
<tr>
<td>- avoid overfitting ('tuning to the test set')</td>
</tr>
<tr>
<td>- more conservative estimate of performance</td>
</tr>
</tbody>
</table>

**Cross-validation over multiple splits**

- Handle sampling errors from different datasets
- Pool results over each split
- Compute pooled dev set performance