Today’s Schedule

- Text classification
- Bag-of-words models
- Multinomial document model
- Bernoulli document model
- Generative models
- Zero Probability Problem

Identifying Spam

- Spam?

Dear Dr. Steve Renals, The proof for your article, Combining Spectral Representations for Large-Vocabulary Continuous Speech Recognition, is ready for your review. Please access your proof via the user ID and password provided below. Kindly log in to the website within 48 HOURS of receiving this message so that we may expedite the publication process.

- How do we represent \( D \)?
  - A sequence of words: \( D = (X_1, X_2, \ldots, X_n) \)
  - A set of words (Bag-of-Words)
    - Ignore the position of the word
    - Ignore the order of the word
    - Consider the words in pre-defined vocabulary \( V (D = |V|) \)
  - Multinomial document model: a document is represented by an integer feature vector, whose elements indicate frequency of corresponding word in the document
    \[ x = (x_1, \ldots, x_K) \quad x_i \in \mathbb{N} \]
  - Bernoulli document model: a document is represented by a binary feature vector, whose elements indicate absence or presence of corresponding word in the document
    \[ b = (b_1, \ldots, b_D) \quad b_i \in \{0, 1\} \]

BoW models: Bernoulli vs. Multinomial

Document \( D \): “Congratulations to you as we bring to your notice, the results of the First Category draws of THE HOLLAND CASINO LOTTO PROMO INT. We are happy to inform you that you have emerged a winner under the First Category, which is part of our promotional draws.”

Notation for document model

- Training documents:
  - Class Documents
    - Multinomial:
      \[
      \text{Term } (w_i \in V) \quad x = (x_1 \in \mathbb{N}_0) \quad \text{Bernoulli } (b_i \in \{0, 1\})
      \]
    - Term
      \[
      \begin{array}{c|cc}
      \text{Term } (w_i \in V) & \text{Multinomial } (x_1 \in \mathbb{N}_0) & \text{Bernoulli } (b_i \in \{0, 1\}) \\
      \hline
      \text{bring} & 1 & 1 \\
      \text{can} & 0 & 0 \\
      \text{casino} & 1 & 1 \\
      \text{category} & 2 & 1 \\
      \text{congratulations} & 1 & 1 \\
      \text{draws} & 2 & 1 \\
      \text{first} & 1 & 1 \\
      \text{lotto} & 4 & 1 \\
      \text{the} & 1 & 1 \\
      \text{true} & 0 & 0 \\
      \text{winner} & 3 & 1 \\
      \end{array}
      \]
    - \( D = 12 \) \( x = (1, 0, 1, 1, 2, 3) \) \( b = (1, 0, 1, 1, 1, 1, 1) \)
  - Flattened representation of training data:
    - Documents:
      \[
      D_1: D_1^{(1)} \ldots D_1^{(K)} \quad D_2: D_2^{(1)} \ldots D_2^{(K)} \ldots D_N: D_N^{(1)} \ldots D_N^{(K)}
      \]
    - Class indicator:
      \[
      z = \begin{cases} 
      1 & \text{if } D_i \text{ belongs to class } C_k \\
      0 & \text{otherwise}
      \end{cases}
      \]
    - Test document:
      \[
      d = D \quad z = (z_1, \ldots, z_N)
      \]

Identifying Spam

- Spam?

I got your contact information from your country’s information directory during my desperate search for someone who can assist me secretly and confidentially in relocating and managing some family fortunes.
Classification with multinomial document model

Assume a test document \( D \) is given as a sequence of words:

\( (a_1, a_2, \ldots, a_n), \quad a_i \in V \)

Feature vector: \( x = (x_1, x_2, \ldots), \quad \text{word frequencies}, \sum_{i=1}^{n} x_i = n \)

Document likelihood with multinomial distribution:

\[
P(x | C_k) = \frac{n_1! n_2! \cdots n_k!}{n!} P(w_1 | C_k)^{x_1} P(w_2 | C_k)^{x_2} \cdots P(w_k | C_k)^{x_k}
\]

For classification, we can omit irrelevant term, so that:

\[
P(x | C_k) \propto \prod_{i=1}^{k} P(w_i | C_k)^{x_i}
\]

Class \( C_k \) is associated with the document likelihood

\[
P(C_k | x) \propto P(C_k) \prod_{i=1}^{k} P(w_i | C_k)^{x_i}
\]

Training of multinomial document model

Features: \( x = (x_1, x_2, \ldots) \) : word frequencies in a doc.

Training data set

<table>
<thead>
<tr>
<th>Class</th>
<th>Docs</th>
<th>Feature vectors</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_1 )</td>
<td>( x_1^{(1)} )</td>
<td>( x_1^{(1)} ) ( x_1^{(2)} ) ( x_1^{(3)} ) ( x_1^{(4)} )</td>
</tr>
<tr>
<td>( \vdots )</td>
<td>( \vdots )</td>
<td>( \vdots )</td>
</tr>
<tr>
<td>( P(C_k) ) ( = N_k / N )</td>
<td>( P(w_i</td>
<td>C_k) : n_i(w_i</td>
</tr>
</tbody>
</table>

Multinomial doc. model – example

Classify documents as \( S \) or \( S' \)

\[
V: \quad w_1 = \text{goal}, \quad w_2 = \text{tutor}, \quad w_3 = \text{variance}, \quad w_4 = \text{speed}, \quad w_5 = \text{drink}, \quad w_6 = \text{defence}, \quad w_7 = \text{performance}, \quad w_8 = \text{field}
\]

\[
D = |V| = 8
\]

Bernoulli doc. model – example

\[
\text{Classify as } S \quad \text{or} \quad S'.
\]

Classification with Bernoulli document model

A test document \( D \) with feature vector \( b = (b_1, b_2, \ldots, b_k) \)

Document likelihood with Bernoulli distribution:

\[
P(b | C_k) = \prod_{i=1}^{k} P(b_i | C_k) = \prod_{i=1}^{k} (b_i P(w_i | C_k) + (1-b_i)(1-P(w_i | C_k)))
\]

\[
P(w_i | C_k) = \frac{n_{i}(w_i)}{N_k}
\]

\[
P(C_k | b) \propto P(C_k) \prod_{i=1}^{k} P(w_i | C_k)^{b_i}
\]

In Classification,

\[
P(C_k | b) \propto P(C_k) \prod_{i=1}^{k} P(w_i | C_k)^{b_i}
\]

Training of Bernoulli document model

Features: \( b = (b_1, b_2, \ldots, b_k) \) : \( D = |V| \), i.e. vocabulary binary vector of word occurrences in a document

Training data set

<table>
<thead>
<tr>
<th>Class</th>
<th>Docs</th>
<th>Feature vectors</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_1 )</td>
<td>( b_1^{(1)} )</td>
<td>( b_1^{(1)} ) ( b_1^{(2)} ) ( b_1^{(3)} ) ( b_1^{(4)} )</td>
</tr>
<tr>
<td>( \vdots )</td>
<td>( \vdots )</td>
<td>( \vdots )</td>
</tr>
<tr>
<td>( P(C_k) ) ( = N_k / N )</td>
<td>( P(w_i</td>
<td>C_k) : n_i(w_i</td>
</tr>
</tbody>
</table>

Bernoulli doc. model – example (cont.)

Test documents: \( b = \{ 0 1 0 0 1 1 1 1 0 1 \} \)

Priors, Likelihoods:

\[
P(S) = 6/11, \quad P(T) = 5/11
\]

\[
P(w_i | S) = (3/6 \times 1/6 \times 2/6 \times 2/3 \times 3/6 \times 4/6 \times 4/6 \times 6/6)
\]

\[
P(w_i | D) = (1/6 \times 3/5 \times 3/5 \times 1/5 \times 1/5 \times 1/5 \times 1/5 \times 1/5)
\]

Posterior probabilities:

\[
P(S | b) \propto P(S) \prod_{i=1}^{k} [b_i P(w_i | S) + (1-b_i)(1-P(w_i | S))]
\]

\[
\alpha = P(S | b) \propto P(S) \prod_{i=1}^{k} [b_i P(w_i | S) + (1-b_i)(1-P(w_i | S))]
\]

Classify this document as \( S \).

Summary of the document models

<table>
<thead>
<tr>
<th>Class</th>
<th>Multinomial doc. model</th>
<th>Bernoulli doc. model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Docs</td>
<td>Feature vectors</td>
<td>Feature vectors</td>
</tr>
<tr>
<td>( P(C_k) = N_k / N )</td>
<td>( n_i(w_i</td>
<td>S) / n_i(w_i</td>
</tr>
</tbody>
</table>

\[
P(X | C_k) \propto \prod_{i=1}^{D} P(w_i | C_k)^{x_i} = \prod_{i=1}^{D} P(w_i | C_k)
\]

\[
P(b | C_k) = \prod_{i=1}^{D} P(b_i | C_k) = \prod_{i=1}^{D} P(b_i | C_k)
\]

\[
P(b | C_k) = \prod_{i=1}^{D} [P(b_i | C_k) + (1-b_i)(1-P(w_i | C_k))]
\]

\[
P(X | C_k) \propto \prod_{i=1}^{D} P(w_i | C_k)^{x_i} = \prod_{i=1}^{D} P(w_i | C_k)
\]

\[
P(b | C_k) = \prod_{i=1}^{D} [P(b_i | C_k) + (1-b_i)(1-P(w_i | C_k))]
\]
# Text Classification using Naive Bayes

## Generative models

- **Models that generate observable data randomly based on a distribution**
- **Examples**
  - Coin tossing models
  - Dice throwing models

<table>
<thead>
<tr>
<th>Coin</th>
<th>Generated data sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fair coin (P((H))= (P(T))=0.5)</td>
<td>(H, T, T, H, T, H, T, T, H, T, H, T, H, T, H, H, T, H, H, T, H, T, H, T)</td>
</tr>
<tr>
<td>Unfair coin (P((H))=0.7, P((T))=0.3)</td>
<td>(T, H, H, H, H, T, H, T, H, T, H, T, H, T, H, T, H, T, H, T, H, T, H, T)</td>
</tr>
</tbody>
</table>

## Dice throwing models

- Unbiased dice \(P(X=1/6, X \in \{1, \ldots, 6\})\):
  - 2, 4, 3, 5, 3, 6, 5, 4, 6, 5, 5, 6, 1, 2, 6, 6, 6, 6
- Biased dice \(P(X=1)\):
  - \(0.1, 0.1, 0.1, 0.1, 0.2, 0.4\)
  - 6, 6, 5, 5, 6, 1, 2, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6

## Generative model — Multinomial document model

- \(o_1, o_2, o_3, \ldots, o_L | \mathcal{C}_k\)

## Generative model — Bernoulli document model

- Terms in Document \(D\) in \(\mathcal{C}_k\)

<table>
<thead>
<tr>
<th>(b_1)</th>
<th>(b_2)</th>
<th>(\ldots)</th>
<th>(b_D)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>(\ldots)</td>
<td>0</td>
</tr>
</tbody>
</table>

## Word relative-frequencies of spam emails

- **# of spam emails:** 169

<table>
<thead>
<tr>
<th><strong>to</strong></th>
<th><strong>from</strong></th>
<th><strong>subject</strong></th>
<th><strong>body</strong></th>
<th><strong>sent</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>you</td>
<td>0.00377344</td>
<td>0.00385205</td>
<td>0.00377344</td>
<td>0.00377344</td>
</tr>
<tr>
<td>us</td>
<td>0.00244512</td>
<td>0.00244512</td>
<td>0.00244512</td>
<td>0.00244512</td>
</tr>
<tr>
<td>me</td>
<td>0.00244512</td>
<td>0.00244512</td>
<td>0.00244512</td>
<td>0.00244512</td>
</tr>
<tr>
<td>bank</td>
<td>0.00244512</td>
<td>0.00244512</td>
<td>0.00244512</td>
<td>0.00244512</td>
</tr>
<tr>
<td>usd</td>
<td>0.00244512</td>
<td>0.00244512</td>
<td>0.00244512</td>
<td>0.00244512</td>
</tr>
<tr>
<td>atm</td>
<td>0.00244512</td>
<td>0.00244512</td>
<td>0.00244512</td>
<td>0.00244512</td>
</tr>
<tr>
<td>good</td>
<td>0.00244512</td>
<td>0.00244512</td>
<td>0.00244512</td>
<td>0.00244512</td>
</tr>
<tr>
<td>part</td>
<td>0.00244512</td>
<td>0.00244512</td>
<td>0.00244512</td>
<td>0.00244512</td>
</tr>
<tr>
<td>new</td>
<td>0.00244512</td>
<td>0.00244512</td>
<td>0.00244512</td>
<td>0.00244512</td>
</tr>
<tr>
<td>from</td>
<td>0.00244512</td>
<td>0.00244512</td>
<td>0.00244512</td>
<td>0.00244512</td>
</tr>
</tbody>
</table>

## Word sequence example

- **Generated word sequence example**
  - of kin good your the part of with and atm to new from which projects has the transfer while the by for your willing

## Model for classification

\[ P(\mathcal{C}_i | x) = \frac{P(x | \mathcal{C}_i) P(\mathcal{C}_i)}{P(x)} \propto P(x | \mathcal{C}_i) P(\mathcal{C}_i) \]

## Model for observation

- **generative model**

\[ P(x) = \sum_{i=1}^{N} P(x | \mathcal{C}_i) P(\mathcal{C}_i) \]

## Smoothing in multinomial document model

- **Zero probability problem**

\[ \Pr(x | \mathcal{C}_i) = \prod_{j=1}^{D} P(w_j | \mathcal{C}_i) = 0 \text{ if } \exists j : P(w_j | \mathcal{C}_i) = 0 \]

\[ P(w_j | \mathcal{C}_i) = \frac{\sum_{k=1}^{N} n_{k,w_j} \frac{1}{d_k}}{\sum_{k=1}^{N} \frac{1}{d_k}} = \frac{n_{k,w_j}}{d_k} \]

- **Smoothing — a ‘trick’ to avoid zero counts:**

\[ P(w_j | \mathcal{C}_i) = \frac{1 + \sum_{k=1}^{N} n_{k,w_j} \frac{1}{d_k}}{V + \sum_{k=1}^{N} \frac{1}{d_k}} = \frac{1 + n_{k,w_j}}{D + \sum_{k=1}^{N} n_{k,w_j}} \]

Known as Laplace’s rule of succession or add one smoothing.
**Exercise 1**

Use the Bernoulli model and the Naive Bayes assumption for the following. Consider the vocabulary \( V = \{\text{apple, banana, computer}\} \). We have two classes of documents \( F \) (fruit) and \( E \) (electronics). There are four training documents in class \( F \); they are listed below in terms of the number of occurrences of each word from \( V \) in each document:

- apple(2); banana(1); computer(0)
- apple(0); banana(1); computer(0)
- apple(3); banana(2); computer(1)
- apple(1); banana(0); computer(0)

There are also four training documents in class \( E \):

- apple(2); banana(0); computer(0)
- apple(0); banana(0); computer(1)
- apple(3); banana(1); computer(2)
- apple(0); banana(0); computer(1)

1. Write the training data as a matrix for each class, where each row corresponds to a training document.
2. Estimate the prior probabilities from the training data.
3. For each class \( F \) and \( E \) and for each word (apple, banana and computer) estimate the likelihood of the word given the class.
4. Consider two test documents:
   - apple(1); banana(0); computer(0)
   - apple(1); banana(1); computer(0)

   For each test document, estimate the posterior probabilities of each class given the document, and hence classify the document.

**Exercise 2**

Use the Multinomial model and the Naive Bayes assumption for the following. Consider the vocabulary \( V = \{\text{fish, chip, circuit}\} \). We have two classes of documents \( F \) (food) and \( E \) (electronics). There are four training documents in class \( F \); they are listed below:

- fish(0); chip(1)
- circuit(1)
- fish(1); chip(1)
- chip(1)

There are also four training documents in class \( E \):

- fish(0); chip(0)
- circuit(1)
- circuit(2)
- chip(0)

1. Write the training data as a matrix for each class, where each row corresponds to a training document.
2. Estimate the prior probabilities from the training data.
3. For each class \( F \) and \( E \) and for each word (apple, banana and computer) estimate the likelihood of the word given the class.
4. Consider two test documents:
   - apple(1); banana(0); computer(0)
   - apple(1); banana(1); computer(0)

   For each test document, estimate the posterior probabilities of each class given the document, and hence classify the document.

**Exercise 3**

Consider two writers, Baker and Clark, who were twins, and who published four and six childrens books, respectively. The following table shows the frequencies of four words, wizard, river, star, and warp, with respect to the first page of each book, and the information whether the book was a bestseller or not.

<table>
<thead>
<tr>
<th>Author</th>
<th>wizard</th>
<th>river</th>
<th>star</th>
<th>warp</th>
<th>Bestseller</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baker</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>No</td>
</tr>
<tr>
<td>Baker</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Yes</td>
</tr>
<tr>
<td>Baker</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>No</td>
</tr>
<tr>
<td>Clark</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>No</td>
</tr>
<tr>
<td>Clark</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>Yes</td>
</tr>
<tr>
<td>Clark</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Two unpublished book drafts, Doc 1 and Doc 2, were found after the death of the writers, but its not clear which of them wrote the documents.
Summary

- Our first 'real' application of Naive Bayes
- Two BoW models for documents: Multinomial and Bernoulli
- Generative models
- Smoothing (Add-one/Laplace smoothing)

- As always: be able to implement, describe, compare and contrast (see Lecture Note)