Information Retrieval (5LN712)

Text classification and naïve Bayes

2021-04-29

Ali Basirat

Department of Computer and Information Science (IDA)
Linköping University

Department of Linguistics and Philology
Uppsala University
Today

• The text classification problem
• The naïve Bayesian text classification
• Multinomial model
• The Bernoulli model
• A comparison between the two models
• Summary
The text classification problem

- Queries are generated by probabilistic models from some topics (or document classes)
- For example: the terms Taipei and Beijing are generated from a broader document class, China
- In order to classify a document, we should find a class that is more likely to generate the document
The text classification problem

- For a given document space $X$ and a set $f$ of classes $C$, a document classifier maps each document description in $X$ to a class in $C$

$$\gamma : X \rightarrow C$$

- The classification function $\gamma$ is learnt from a training set of labelled documents $(d, c) \in X \times C$
The Naïve Bayesian Approach

• The document-class probabilities can be estimated by the Bayesin equation:

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

• The denominator \( P(d) \) is a constant for a document and can be ignored
The Naïve Bayesian Approach

- The prior probability $P(c)$ can be estimated from the data and other information about the classes.
- We can make a naïve assumption about the independence occurrence of words in a document.
- This assumption is not correct but helps us to have an estimation of the document probabilities.
The Naïve Bayesian Approach

- If we consider a **multinominal** probability distribution over positions

\[
P(d|c) \approx \prod_{t \in d} P(t|c)
\]

- If we consider a **Bernoulli** distribution over the occurrence of words

\[
P(d|c) \approx \prod_{t \in V} P(t|c)
\]
Multinomial model

- The probability of a document $d$ being in class $c$

$$P(c|d) \propto p(c) \prod_{1 \leq k \leq n_d} P(t_k|c)$$

- The best class for a document in NB is the most likely one (maximum a posteriori):

$$c_{map} = \arg \max_{c \in C} \hat{P}(c|d)$$
• If we expand $\hat{P}(c|d)$, we will have

$$c_{map} = \arg \max_{c \in C} \hat{P}(c) \prod_{1 \leq k \leq n_d} \hat{P}(t_k|c)$$

• We can also take a logarithm from the right side

$$c_{map} = \arg \max_{c \in C} \left[ \log \hat{P}(c) + \sum_{1 \leq k \leq n_d} \log \hat{P}(t_k|c) \right]$$
Multinomial model

• We may have zero probabilities for some terms
• It can be resolved by an add-one or Laplace smoothing

\[ \hat{P}(t|c) = \frac{T_{ct} + 1}{\sum_{t \in V} (T_{ct'} + 1)} = \frac{T_{ct} + 1}{|c| + |V|} \]

• \(|c|\) is the total number of terms in class \(c\)
• \(|V|\) is the vocabulary size of the collection
## Multinomial model

<table>
<thead>
<tr>
<th>Doc ID</th>
<th>Words in documents</th>
<th>class</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Training set</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Chinese Beijing Chinese</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Chinese Chinese Shanghai</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Chinese Macao</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>Tokyo Japan Chinese</td>
<td>0</td>
</tr>
<tr>
<td><strong>Test set</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Chinese Chinese Chinese Tokyo Japan</td>
<td>?</td>
</tr>
</tbody>
</table>
Multinomial model

- \( V = \{ \text{Chinese, Beijing, Shanghai, Macao, Tokyo, Japan} \} \) (|\( V \)| = 6)
- \( |C_1| = 8, |C_0| = 3 \)

- \( P(C_1) = 0.75, P(C_0) = 0.25 \)

- \( P(\text{Chinese}|C_1) = 5 + 1/8 + 6 = 3/7 \)
- \( P(\text{Chinese}|C_0) = 1 + 1/3 + 6 = 2/9 \)
- \( P(\text{Tokyo}|C_1) = 1/14, P(\text{Tokyo}|C_0) = 2/9 \)
- \( P(\text{Japan}|C_1) = 1/14, P(\text{Japan}|C_0) = 2/9 \)

- \( P(C_1|d_5) \approx (3/4) \times (3/7)^3 \times (1/14) \times (1/14) = 0.0003 \)
- \( P(C_0|d_5) \approx (1/4) \times (2/9)^3 \times (2/9) \times (2/9) = 0.0001 \)
Multinomial model

**TRAINMULTINOMIALNB(C, D)**

1. $V \leftarrow \text{EXTRACTVOCABULARY}(D)$
2. $N \leftarrow \text{COUNTDOCS}(D)$
3. for each $c \in C$
4. do $N_c \leftarrow \text{COUNTDOCSINCLASS}(D, c)$
5. $\text{prior}[c] \leftarrow N_c / N$
6. $text_c \leftarrow \text{CONCATENATETEXTOFALLDOCSINCLASS}(D, c)$
7. for each $t \in V$
8. do $T_{ct} \leftarrow \text{COUNTTOKENSOFTERM}(text_c, t)$
9. for each $t \in V$
10. do $\text{condprob}[t][c] \leftarrow \frac{T_{ct} + 1}{\sum_{t'} (T_{ct'} + 1)}$
11. return $V, \text{prior}, \text{condprob}$

**APPLYMULTINOMIALNB(C, V, prior, condprob, d)**

1. $W \leftarrow \text{EXTRACTTOKENSFROMDOC}(V, d)$
2. for each $c \in C$
3. do $\text{score}[c] \leftarrow \log \text{prior}[c]$
4. for each $t \in W$
5. do $\text{score}[c] += \log \text{condprob}[t][c]$
6. return arg max$_{c \in C} \text{score}[c]$

▶ Figure 13.2 Naive Bayes algorithm (multinomial model): Training and testing.
Multinomial model

- The multinomial NB (A) is identical to the multinomial unigram language model (B)
  - A: \( P(c|d) \propto p(c) \prod_{1 \leq k \leq n_d} P(t_k|c) \)
  - B: \( P(d|q) \propto P(d) \prod_{t \in q} P(t|M_d) \)
- The document in A takes the role of a query in B
- The class in A takes the role of a document in B
The Bernoulli model

- A Bernoulli random variable takes one of two values.
- A multivariate Bernoulli variable is a vector of Bernoulli variables which is used as an indicator.
- When using in text classification, a Bernoulli model only considers the presence of a term in a class.
- It ignores the number of occurrences.
- It is equivalent to the binary independence model.
Bernoulli model

\[
P(c|d) \propto P(c)P(d|c)
\]

\[
P(d = [t_1, \ldots, t_M]|c; V) = \prod_{t \in V} P(t|c) = \prod_{t \in V_d} P(t|c) \prod_{t \notin V_t} (1 - P(t|c))
\]

\[
P(c) = \frac{N_c}{N}
\]

\[
P(t|c) = \frac{N_{ct} + 1}{N_c + 2}
\]

- \(N_c\) is the number of documents in class c
- \(N_{ct}\) is the number of document in class c containing term t
- \(N\) is the total number of documents
## Bernoulli model

<table>
<thead>
<tr>
<th>Training set</th>
<th>Doc ID</th>
<th>Words in documents</th>
<th>class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>Chinese Beijing Chinese</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Chinese Chinese Shanghai</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Chinese Macao</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>Tokyo Japan Chinese</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test set</th>
<th>Doc ID</th>
<th>Words in documents</th>
<th>class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5</td>
<td>Chinese Chinese Chinese Tokyo Japan</td>
<td>?</td>
</tr>
</tbody>
</table>
Multinomial model

- \( V = \{ \text{Chinese, Beijing, Shanghai, Macao, Tokyo, Japan} \} \ (|V| = 6) \)
- \( N_1 = 3, N_0 = 1, N = 4 \)

- \( P(C_1) = 0.75, P(C_0) = 0.25 \)

- \( P(\text{Chinese}|C_1) = 3 + 1/3 + 2 = 4/5 \)
- \( P(\text{Chinese}|C_0) = 1 + 1/1 + 2 = 2/3 \)
- \( P(\text{Tokyo}|C_1) = 1/5, P(\text{Tokyo}|C_0) = 2/3 \)
- \( P(\text{Japan}|C_1) = 1/5, P(\text{Japan}|C_0) = 2/3 \)

- \( P(\text{Beijing}|C_1) = P(\text{Macao}|C_1) = P(\text{Shanghai}|C_1) = 2/5 \)
- \( P(\text{Beijing}|C_0) = P(\text{Macao}|C_0) = P(\text{Shanghai}|C_0) = 1/3 \)

- \( P(C_1|d_5) \approx 0.75 \times \frac{4}{5} \times \frac{1}{5} \times \frac{1}{5} \times \frac{3}{5} \times \frac{3}{5} \times \frac{3}{5} = 0.0051 \)
- \( P(C_0|d_5) \approx 0.25 \times \frac{2}{3} \times \frac{2}{3} \times \frac{2}{3} \times \frac{2}{3} \times \frac{2}{3} \times \frac{2}{3} = 0.0219 \)
Bernoulli model

**TrainBernoulliNB**\((C, D)\)
1. \(V \leftarrow \text{ExtractVocabulary}(D)\)
2. \(N \leftarrow \text{CountDocs}(D)\)
3. for each \(c \in C\) do
4.   \(N_c \leftarrow \text{CountDocsInClass}(D, c)\)
5.   \(\text{prior}[c] \leftarrow N_c / N\)
6.   for each \(t \in V\) do
7.     \(N_{ct} \leftarrow \text{CountDocsInClassContainingTerm}(D, c, t)\)
8.     \(\text{condprob}[t][c] \leftarrow (N_{ct} + 1) / (N_c + 2)\)
9. return \(V, \text{prior, condprob}\)

**ApplyBernoulliNB**\((C, V, \text{prior, condprob}, d)\)
1. \(V_d \leftarrow \text{ExtractTermsFromDoc}(V, d)\)
2. for each \(c \in C\) do
3.   \(\text{score}[c] \leftarrow \log \text{prior}[c]\)
4.   for each \(t \in V\) do
5.     if \(t \in V_d\) then
6.       \(\text{score}[c] += \log \text{condprob}[t][c]\)
7.     else \(\text{score}[c] += \log (1 - \text{condprob}[t][c])\)
8. return \(\arg \max_{c \in C} \text{score}[c]\)
## Multinomial vs Bernoulli model

<table>
<thead>
<tr>
<th></th>
<th>Multinomial model</th>
<th>Bernoulli model</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Event model</strong></td>
<td>Generation of token</td>
<td>Generation of document</td>
</tr>
<tr>
<td><strong>Random variable</strong></td>
<td>$X = t \text{ iff } t$ occurs at given position</td>
<td>$U_t = 1 \text{ iff } t$ occurs in the document</td>
</tr>
<tr>
<td><strong>Document representation</strong></td>
<td>$d = \langle t_1, ..., t_k, ..., t_{nd} \rangle, t_k \in V$</td>
<td>$d = \langle e, ..., e_i, ..., e_M \rangle, e_i \in {0,1}$</td>
</tr>
<tr>
<td><strong>Parameter estimation</strong></td>
<td>$\hat{P}(X = t</td>
<td>c)$</td>
</tr>
<tr>
<td><strong>Multiple occurrences</strong></td>
<td>Taken into account</td>
<td>Ignored</td>
</tr>
<tr>
<td><strong>Length of documents</strong></td>
<td>Can handle longer documents</td>
<td>Works best for short documents</td>
</tr>
<tr>
<td><strong>Number of features</strong></td>
<td>Can handle more</td>
<td>Works best with fewer</td>
</tr>
<tr>
<td><strong>Estimate for term ‘the’</strong></td>
<td>$\hat{P}(X = \text{‘the’}</td>
<td>c) \approx 0.05$</td>
</tr>
</tbody>
</table>
Summary

- Documents can be classified into multiple classes
- A text classifier is a function from document space to a label set
- A naïve Bayes classifier can be used for document classification
- It is similar to a language model information retrieval
- A Bernoulli model can also be used for document classification