Chapter 2

Supervised Bernoulli Text Topic Identification Model using Naïve Bayes

2.1 Introduction

The primary objective of this chapter is to identify uncategorized documents with either of the given categories or labels based on the Bernoulli distribution of the terms present in the labelled documents (also known as the training dataset). The task of classifying a piece of text algorithmically does not often require the algorithm to have a deep understanding of the language. This is primarily due to the fact that any text document, irrespective of language, finds a bag-of-words representation whatever be the setting of the algorithm in the background. Here the bag resembles a collection having elements where repetition is allowed. This representation is very simple and intuitive and simply focuses on the words occurring in the text document and the frequency. As a result, the notion of ordering of the words or the arrangement in which it existed in the original document, gets ignored. Let there be a document $D$ having class/topic/category denoted by $C$. Let the various realizations of $C$ be denoted as $C_1, C_2$ etc. The posterior probability $P(C | D)$ is given by

$$P(C | D) = \frac{P(D | C) \times P(C)}{P(D)} \propto P(D | C)P(C)$$  \hspace{1cm} (2.1)

Under the assumption of Naïve Bayes (Qin, Tang and Chen, 2012), Bernoulli document model will be proposed in this chapter which will follow a bag-of-words representation for the documents. The documents are represented in the model with the help of feature vectors, the components of which are types of words occurring in the documents. It is assumed that there is a vocabulary $V$ over the given documents, which contains $|V|$ types of words. In Bernoulli document model, documents find representation using feature vectors with Boolean or binary components (Manning, Raghavan and Schutze, 2008) assuming realization 1 if the equivalent word occurs in the document and 0 if the word is absent. Similarly, for Multinomial document model, documents find representation using feature vectors with integer components having values denoting the frequencies of the associated words in the concerning documents.
In Section 2.2, we describe the Bernoulli model setup in detail and give an example in Section 2.3. In Section 2.4, we show the application of Bernoulli document model on a real-life data and its implementation in R. Implementation in Python is provided in same section which is followed by conclusions in Section 2.5.

### 2.2 Bernoulli Model

A document, in a Bernoulli set up, finds representation using a vector of binary type, which in turn is a representation of a point in the word space. Let there be a vocabulary $V$ having $|V|$ words. The $t^{th}$ item of a document vector will correspond to word $w_t$ of the vocabulary.

If $b$ represents document $D$’s feature vector; then $b_t$, $t^{th}$ item of $b$, assumes the value 0 or 1 depending upon the non-occurrence/occurrence of word $w_t$ within the document text.

As an example, let there be a vocabulary:

$$V = \{\text{blue, green, dog, tiger, biscuit, banana}\}$$

Cardinality of $V$ or $|V|$= dimension of feature vector $D = 6$.

To illustrate further, consider a document “the blue dog ate a blue biscuit”. Let $d^B$ be the associated Bernoulli feature vector, and $d^M$ the multinomial feature vector. Thus

$$d^B = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 2 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Hence, for classifying the document, (2.1) can be used. Now that requires estimation of the document’s likelihood conditional on the class/topic/category $C$, $P(D|C)$ and the associated prior probability $P(C)$ of the classes. The Naive Bayes assumptions are applicable to either of the two document models that would be used, while estimating the likelihood $P(D|C)$.  

33
Under the assumption of Naive Bayes which suggests that event of any word being present in the document is not dependent on presence of any other word, enables us to express the likelihood $P(D|C_k)$ of the document in terms of the distinct solitary word likelihoods $P(w_i|C_k)$ as

$$P(D|C_k) = P(b|C_k) = \prod_{t=1}^{V_k} \left[ b_t \times P(w_t|C_k) + (1-b_t) \times (1-P(w_t|C_k)) \right]$$  \hspace{1cm} (2.2)

where $P(w_t|C_k)$ is the probability that word $w_t$ occurs in a document belonging to $k^{th}$ category or topic and $(1 - P(w_t|C_k))$ is the probability that $w_t$ does not occur in some document belonging to this category. (2.2) iterates over every word present in the vocabulary.

If word $w_t$ occurs, then $b_t$ equals 1 and associated probability is $P(w_t|C_k)$. Otherwise, $b_t$ equals zero with associated probability as $(1 - P(w_t|C_k))$. This can be looked upon as a model to generate feature vectors of documents belonging to class $k$, where the feature vector of the document is exhibited as a collection of coin tosses with $|V|$ weights, where $t^{th}$ toss has success probability as $P(w_t|C_k)$.

Let $n_k(w_t)$ be the count of documents of category $k$ where $w_t$ is present and let $N_k$ be the total count of documents belonging to the $k^{th}$ class. Then

$$\hat{P}(w_t|C_k) = \frac{n_k(w_t)}{N_k}$$  \hspace{1cm} (2.3)

represents relative frequency of documents belonging to category $k$ and containing $w_t$.

For a given training set having $N$ documents, the prior probability for category $k$ is written as

$$\hat{P}(C_k) = \frac{N_k}{N}$$  \hspace{1cm} (2.4)

Therefore, for a given training dataset consisting of labeled documents, each associated to either of the $k$ categories, a Bernoulli classification model can be estimated in the following manner:

1. Define the vocabulary $V$ where the count of words in it provides the feature vector dimensionality;
2. In the training dataset, enumerate
   - \( N \), the total count of documents in training dataset
   - \( N_k \), the count of documents with category labels \( k \), where \( k \) ranges from 1 to \( K \)
   - \( n_k(w_t) \), the count of documents belonging to category \( k \), and having word \( w_t \) where \( k \)
     ranges from 1 to \( K \) and \( t \) ranges from 1 to \( |V| \)
3. Estimation of the likelihood \( P(w_t|C_k) \) using (2.3);
4. Estimation of the prior probabilities \( P(C_k) \) using (2.4).

Finally, for classifying an unknown and unseen document \( D \), the posterior probability needs to be estimated for each category \( k \) using the combination of (2.1) with the Bernoulli model document likelihood equation in the following manner

\[
P(C_k | b) \propto P(b | C_k) \times P(C_k)
\]

\[
\propto P(C_k) \left[ \prod_{t=1}^{\lfloor |V| \rfloor} [b_t \times P(w_t | C_k) + (1 - b_t) \times (1 - P(w_t | C_k))] \right]
\]

\[ (2.5) \]

### 2.3 Example of Bernoulli Model

Let there be a collection of documents, every one of which belongs to one of the two topics, that is, Sports or Informatics, denoted by S and I respectively. Now, for a given training dataset having eleven documents, the objective is establishing an estimation for a Bernoulli document classifier, to label unseen documents pertaining to Sports or Informatics.

Let the vocabulary \( V \) consist of 8 words as

\[
V = \begin{bmatrix}
w_1 = \text{goal} \\
w_2 = \text{tutor} \\
w_3 = \text{variance} \\
w_4 = \text{speed} \\
w_5 = \text{drink} \\
w_6 = \text{defence} \\
w_7 = \text{performance} \\
w_8 = \text{field}
\end{bmatrix}
\]
Hence, all the documents can have a representation in form of a vector of 8 dimensions. A document $D_i$ can now be denoted as a row vector $m_i$ where $m_{it}$ denotes the count of word $w_t$ in $D_i$.

The training dataset is shown below in the form of a matrix corresponding to each category or topic where each row signifies a document vector of 8 dimensions.

$$
B^{\text{Sport}} = \begin{bmatrix}
1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
\end{bmatrix}
$$

$$
B^{\text{Inf}} = \begin{bmatrix}
0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
\end{bmatrix}
$$

Now, the objective is classifying the following vectors into either of the topics with the help of an Naïve Bayes (NB) classifier.

1. $b_1 = (1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1)^T$
2. $b_2 = (0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0)^T$

Enumerations in the training dataset are made as

$N = 11, N_S = 6, N_I = 5.$

The prior probabilities can be estimated from the training dataset using (2.4) and are given by

$$
\hat{P}(S) = \frac{6}{11}; \quad \hat{P}(I) = \frac{5}{11}.
$$
The count of documents $n_k(w)$ in the training dataset, and estimates of word likelihoods are given in the table 2.1.

**Table 2.1: Count of Documents and Estimates of Word Likelihood**

<table>
<thead>
<tr>
<th>Words</th>
<th>$n_k(w)$</th>
<th>$\hat{P}(w \mid S)$</th>
<th>$n_k(w)$</th>
<th>$\hat{P}(w \mid I)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$W_1$</td>
<td>3</td>
<td>$\frac{3}{6}$</td>
<td>1</td>
<td>$\frac{1}{5}$</td>
</tr>
<tr>
<td>$W_2$</td>
<td>1</td>
<td>$\frac{1}{6}$</td>
<td>3</td>
<td>$\frac{4}{5}$</td>
</tr>
<tr>
<td>$W_3$</td>
<td>2</td>
<td>$\frac{2}{6}$</td>
<td>3</td>
<td>$\frac{3}{5}$</td>
</tr>
<tr>
<td>$W_4$</td>
<td>3</td>
<td>$\frac{3}{6}$</td>
<td>1</td>
<td>$\frac{1}{5}$</td>
</tr>
<tr>
<td>$W_5$</td>
<td>3</td>
<td>$\frac{3}{6}$</td>
<td>1</td>
<td>$\frac{1}{5}$</td>
</tr>
<tr>
<td>$W_6$</td>
<td>4</td>
<td>$\frac{3}{6}$</td>
<td>1</td>
<td>$\frac{2}{5}$</td>
</tr>
<tr>
<td>$W_7$</td>
<td>4</td>
<td>$\frac{4}{6}$</td>
<td>3</td>
<td>$\frac{3}{5}$</td>
</tr>
<tr>
<td>$W_8$</td>
<td>4</td>
<td>$\frac{4}{6}$</td>
<td>1</td>
<td>$\frac{1}{5}$</td>
</tr>
</tbody>
</table>

Now the posterior probabilities of two test data points are computed for purpose of classification.

$$
\hat{P}(S \mid b_i) \propto \hat{P}(S) \times \prod_{i=1}^{8} [b_{ii} \times \hat{P}(w_i \mid S) + (1 - b_{ii}) \times (1 - \hat{P}(w_i \mid S))]
$$

$$
\approx \frac{6}{11} \left( \frac{1}{2} \times \frac{5}{6} \times \frac{2}{3} \times \frac{1}{2} \times \frac{1}{3} \times \frac{2}{3} \times \frac{1}{2} \times \frac{2}{3} \times \frac{1}{3} \right) = \frac{5}{891}
$$

$$
\approx 5.6 \times 10^{-3}
$$
\[ \hat{P}(I \mid b_1) \propto \hat{P}(I) \times \prod_{i=1}^{8} \left[ b_{1i} \times \hat{P}(w_i \mid I) + (1-b_{1i}) \times (1-\hat{P}(w_i \mid I)) \right] \]
\[ \propto \frac{5}{11} \left( \frac{1}{5} \times \frac{2}{5} \times \frac{2}{5} \times \frac{1}{5} \times \frac{1}{5} \times \frac{2}{5} \times \frac{1}{5} \right) = \frac{8}{859375} \]
\[ \approx 9.3 \times 10^{-6}. \]

Hence, \( b_1 \) can be categorized as belonging to \( S \).

\[ \hat{P}(S \mid b_2) \propto \hat{P}(S) \times \prod_{i=1}^{8} \left[ b_{2i} \times \hat{P}(w_i \mid S) + (1-b_{2i}) \times (1-\hat{P}(w_i \mid S)) \right] \]
\[ \propto \frac{6}{11} \left( \frac{1}{2} \times \frac{1}{6} \times \frac{1}{2} \times \frac{1}{3} \times \frac{2}{3} \times \frac{2}{3} \times \frac{3}{3} \right) = \frac{12}{42768} \]
\[ \approx 2.81 \times 10^{-4}. \]

\[ \hat{P}(I \mid b_2) \propto \hat{P}(I) \times \prod_{i=1}^{8} \left[ b_{2i} \times \hat{P}(w_i \mid I) + (1-b_{2i}) \times (1-\hat{P}(w_i \mid I)) \right] \]
\[ \propto \frac{5}{11} \left( \frac{4}{5} \times \frac{3}{5} \times \frac{3}{5} \times \frac{4}{5} \times \frac{1}{5} \times \frac{3}{5} \times \frac{4}{5} \right) = \frac{34560}{4296875} \]
\[ \approx 8.1 \times 10^{-3}. \]

This implies that \( b_2 \) can be categorized as belonging to \( I \).

### 2.4 Real-life Implementation in R

In order to implement Bernoulli Naïve Bayes on text data in R, the following line of action can be adopted. First, the required packages are imported into the R environment as shown in fig 2.1.

![Fig 2.1: Code to Import Required Packages](image-url)
Next, the dataset is imported into the environment using the code shown in Fig. 2.2. The used dataset is the Cornell IMDB movie reviews dataset. There are reviews of 2000 movies and an associated positive or negative label based on the sentiment.

![Fig 2.2: Importing Dataset into R](image)

The imported dataset looks as in Fig 2.3.

![Fig 2.3: Snapshot of Dataset](image)

Then the ordering of the dataset is randomized twice to ensure that any kind of patterns in labels are not present while getting the train and test split.
Fig 2.4: Randomizing the Entire Dataset

The dataset now looks like in Fig 2.5 and compared to the earlier data snapshot (Fig 2.3), it is different in ordering.

Fig 2.5: Randomized Dataset

The `class` variable is converted to type `factor` and corpus of the `text` variable is created.
Fig 2.6: Code for Conversion of ‘class’ to Factor and Creation of Corpus of ‘text’

Once the corpus is created, it is cleaned by data pre-processing such as conversion to lower case, removing punctuations, removing numbers, removing stopwords and clearing leading whitespaces.

Fig 2.7: Preprocessing Step

Once the pre-processing step is completed, a Document Term Matrix on the clean corpus is created.

Fig 2.8: DTM Creation on Cleaned Corpus
On inspecting the Document Term Matrix (DTM), the observed information is shown in Fig 2.9.

**Fig 2.9: Resultant DTM Built on Cleaned Corpus**

It is seen that number of terms, which are features, is quite high (38957). In order to reduce them, the terms which occur at least in 5 or more documents would be considered for the analysis. Thus, a dictionary of only those terms is created and the DTM is constructed on that basis.

**Fig 2.10: Code to Create Improved DTM**

Since the dataset was already randomized, the first 1500 rows are taken as train and remaining 500 as test.
Now the DTM objects are converted back to dataframes for ease of handling and the code for the same is depicted in Fig 2.12.

According to the Bernoulli Document Model proposed, the DTM contain each feature in the form of a 0-1 factor since the underlying distribution is Bernoulli. The code for conversion of features and labels to factors is depicted in fig 2.13.
Finally, Bernoulli Document model is fitted on the training dataset and used to make predictions on the test set.

After fitting the model, existing diagnostics is used to test the validity of the proposed model. The Confusion Matrix function is used to build the confusion matrix and get relevant statistic as shown below:
In Fig 2.15, the fitted model shows an accuracy of ~ 79% in classifying the unlabeled observations. Other model diagnostics are as below:

1. **Accuracy of 79%** implies closeness of the sample statistic to the population parameter.
2. **95% Confidence Interval (CI)** is (0.7495, 0.823). It is a combination of the estimates of intervals and probabilities. It implies that if the identical sampling approach is utilized for selecting distinct samples and an interval estimate is calculated for each of them, then the actual population parameter can be expected to be within the interval estimates for approximately 95% of times.
3. **No Information Rate**, the finest approximation conditional that zero information beyond the complete class distribution is provided, is 0.508.
4. **Kappa**: In the task where two binary variables are attempted by two entities in measuring the identical object, Cohen's Kappa (or simply Kappa) can be used as an agreement
measure between them. Its value is always $\leq 1$. A realization of Kappa 1 suggests agreement in the perfect sense and accordingly for less than 1, 

**Poor**: $< 0.20 \\
**Fair**: $0.20 \leq \text{Kappa} \leq 0.40 \\
**Moderate**: $0.40 \leq \text{Kappa} \leq 0.60 \\
**Good**: $0.60 \leq \text{Kappa} \leq 0.80 \\
**Very good**: $0.80 \leq \text{Kappa} \leq 1.00

In our case, Kappa takes the value 0.5772, which lies in the moderate range.

5. **Mcnemar's Test P-Value**: A small p-value suggests evidences of association.

For our example, it equals $4.994 \times 10^{-6}$ which shows an association between dependent and independent variable.

6. **Sensitivity**, the ability of the test to make correct true positive identification is 0.8821 which is very good.

7. **Specificity**, the ability of the test to make correct true negative identification, is 0.6969.

8. **Pred Value**: Positive Pred Value (PPV) and Negative Pred Value (NPV) are respectively, the proportions of positive and negative outcomes that are True Positive (TP) and True Negative (TN).

PPV is 0.7381 and NPV is 0.8592.

9. **Prevalence**: Here prevalence, share of cases in the given population at an instance, is 0.4920.

Above model diagnostics show that the fitted model is classifying the documents quite well.

A simple implementation of the Bernoulli Naïve Bayes model in Python is done using the following code:

```python
# Load libraries
import numpy as np
from sklearn.naive_bayes import BernoulliNB

# Create three binary features
X = np.random.randint(2, size=(100, 3))

# Create a binary target vector
y = np.random.randint(2, size=(100, 1)).ravel()
```
2.5 Conclusion

In this chapter, it is seen how categorization of unlabeled documents can be done using underlying Bernoulli distribution for words, based on the posterior probabilities obtained from the pre-labeled training dataset. This approach, which is lexical in nature, deals with the features obtained from the labeled training dataset only and focuses on just the presence/absence of words across the documents. Thus, in this way, the supervised approach classifies the unlabeled documents to either of the categories under study.