Naive Bayes Classifier
and application
Laxmiprasad Iyer(109275242)
Prof. Anita Wasilewska
References

Research Paper
Thumbs up? Sentiment Classification using Machine Learning Techniques;
Bo Pang; Lillian Lee; Shivakumar Vaithyanathan; EMNLP 2002

Textbook
Introduction to Information Retrieval - Christopher Manning et al.
Overview

• Text Categorization Problem
• Apriori Probabilities
• Posterior Probabilities and Conditional Independence Assumption
• Comparison of Naive Bayes Classifier to other classifiers and choosing a classifier
• Conditional Independence Assumption
• Research Paper (application of naive bayes classifier)
Consider a Text Categorization Problem

Movie Reviews

Positive Reviews

(love, wonderful, best, great, superb, still, beautiful)

Negative Reviews

(bad, worst, stupid, waste, boring, ?, !)
Dataset (1000 positive, 1000 negative)

<table>
<thead>
<tr>
<th>#love</th>
<th>#wonderful</th>
<th>#best</th>
<th>......</th>
<th>#stupid</th>
<th>#wordn</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>......</td>
<td>0</td>
<td>1</td>
<td>Positive</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>......</td>
<td>0</td>
<td>1</td>
<td>Positive</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>......</td>
<td>1</td>
<td>1</td>
<td>Negative</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>......</td>
<td>1</td>
<td>0</td>
<td>Negative</td>
</tr>
</tbody>
</table>

# is whether the word is present or not
n is the total number of words in the vocabulary (in our case 16165)
Vocabulary is the set of words in the all the reviews
Split Dataset into training and testing data

Training Data = 800 Positive records and 800 negative records
Test Data = 200 Positive records and 200 negative records
Training of naive bayes classifier

The training process is:
Learning(calculating) apriori probabilities
Apriori Probabilities (calculated from training data)

\[
P(\#\text{word1} = 0 \mid \text{positive}) = \frac{\text{count of records from positive class with word1} = 0}{\text{total number of records in positive class}}
\]

\[
P(\#\text{word1} = 1 \mid \text{positive}) = \frac{\text{count of records from positive class with word1} = 0}{\text{total number of records in positive class}}
\]

\[
P(\#\text{word1} = 0 \mid \text{negative}) = \frac{\text{count of records from negative class with word1} = 0}{\text{total number of records in negative class}}
\]

\[
P(\#\text{word1} = 1 \mid \text{negative}) = \frac{\text{count of records from negative class with word1} = 1}{\text{total number of records in negative class}}
\]
Apriori Probabilites (calculated from training data)

Class Probabilities

\[
P(\text{Positive}) = \frac{\text{number of records from positive class}}{\text{total number of records}}
\]

\[
P(\text{Negative}) = \frac{\text{number of records from negative class}}{\text{total number of records}}
\]
Total number of model parameters (to be learnt)

\[ = k^*(n^d) \]

where \( k \) is the total number of classes
\( d \) is the number of attribute values
\( n \) is the total number of attributes

For our example, 522614450, \( k=2, d=2, n=16165 \)
Testing

For each test record (labelled)
1. Calculate posterior probabilities
2. Output label with highest posterior probability
3. If output label is same as original label no error, else error
   Error = Fraction of misclassified records
Posterior Probabilities

Given unseen data $d < d_1, d_2, \ldots, d_n>$

$P(\text{Positive} \mid D)$
$P(\text{Negative} \mid D)$
Bayes Rule

\[ P(\text{Positive} \mid D) = P(D \mid \text{Positive}) \times P(\text{Positive}) \]
\[ P(\text{Negative} \mid D) = P(D \mid \text{Negative}) \times P(\text{Negative}) \]
Conditional Independence Assumption

\[ P(d_1, d_2, d_3, d_4, \ldots d_n \mid \text{Positive}) = P(d_1 \mid \text{Positive}) \cdot P(d_2 \mid \text{Positive}) \cdot \ldots \cdot P(d_n \mid \text{Positive}) \]

\[ P(d_1, d_2, d_3, d_4, \ldots d_n \mid \text{Negative}) = P(d_1 \mid \text{Negative}) \cdot P(d_2 \mid \text{Negative}) \cdot \ldots \cdot P(d_n \mid \text{Negative}) \]
Posterior Probabilities (using bayes rule)

Given unseen data $D <1, 0, 1, 1 \ldots, 0>$

$$P(\text{Positive} \mid D) = P(\text{word}_1 = 1 \mid \text{Positive}) \cdot P(\text{word}_2 = 0 \mid \text{Positive}) \cdot \ldots \cdot P(\text{word}_n = 0 \mid \text{Positive}) \cdot P(\text{Positive})$$

$$P(\text{Negative} \mid D) = P(\text{word}_1 = 1 \mid \text{Negative}) \cdot P(\text{word}_2 = 0 \mid \text{Negative}) \cdot \ldots \cdot P(\text{word}_n = 0 \mid \text{Negative}) \cdot P(\text{negative})$$
Final Naive Bayes Formula

Naive Bayes assumption:

\[ P(a_1, a_2 \ldots a_n | v_j) = \prod_i P(a_i | v_j) \]

which gives

**Naive Bayes classifier:**

\[ v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i | v_j) \]
Final Classifier is the end product

After training and testing
Conditional Independence Assumption

Does the conditional independence assumption hold in practice?

It is very limiting! Yet naive Bayes performs effectively.
Simple interpretation of the formula

$$c_{map} = \arg \max_{c \in \mathcal{C}} \left[ \log \hat{P}(c) + \sum_{1 \leq k \leq n_d} \log \hat{P}(t_k | c) \right].$$
Comparison of naive bayes classifier with other classifiers

26 different datasets, Naive Bayes performed at par with SVM or Decision Tree Classifier except in 3 or 4 cases.
Research Paper

Exciting problem solved using the naive bayes classifier.

Thumbs up? Sentiment Classification using Machine Learning Techniques;
Bo Pang; Lillian Lee; Shivakumar Vaithyanathan;

Journal
EMNLP 2002
3 fold cross validation (rotation estimation)

Divide dataset into 3 folds
Use 2 folds for training and 1 fold for testing
Find the cross validation accuracy
Repeat choosing different combination
Find the average cross validation accuracy
Results (average 3 fold cross validation accuracies)

<table>
<thead>
<tr>
<th>Features</th>
<th># of features</th>
<th>frequency or presence?</th>
<th>NB</th>
<th>ME</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) unigrams</td>
<td>16165</td>
<td>freq.</td>
<td>78.7</td>
<td>N/A</td>
<td>72.8</td>
</tr>
<tr>
<td>(2) unigrams</td>
<td></td>
<td>pres.</td>
<td>81.0</td>
<td>80.4</td>
<td>82.9</td>
</tr>
<tr>
<td>(3) unigrams+bigrams</td>
<td>32330</td>
<td>pres.</td>
<td>80.6</td>
<td>80.8</td>
<td>82.7</td>
</tr>
<tr>
<td>(4) bigrams</td>
<td>16165</td>
<td>pres.</td>
<td>77.3</td>
<td>77.4</td>
<td>77.1</td>
</tr>
<tr>
<td>(5) unigrams+POS</td>
<td>16695</td>
<td>pres.</td>
<td>81.5</td>
<td>80.4</td>
<td>81.9</td>
</tr>
<tr>
<td>(6) adjectives</td>
<td>2633</td>
<td>pres.</td>
<td>77.0</td>
<td>77.7</td>
<td>75.1</td>
</tr>
<tr>
<td>(7) top 2633 unigrams</td>
<td>2633</td>
<td></td>
<td>80.3</td>
<td>81.0</td>
<td>81.4</td>
</tr>
<tr>
<td>(8) unigrams+position</td>
<td>22430</td>
<td>pres.</td>
<td>81.0</td>
<td>80.1</td>
<td>81.6</td>
</tr>
</tbody>
</table>
Final Classifier

Train using whole training data
Feature selection

Two main purposes:
1. Training and applying a classifier more efficient
2. Feature selection often increases classification accuracy by eliminating noise features
Feature Selection Algorithm

\[
\text{SELECTFEATURES}(\mathcal{D}, c, k)\\
1. \quad V \leftarrow \text{EXTRACTVOCABULARY}(\mathcal{D})\\
2. \quad L \leftarrow []\\
3. \quad \text{for each } t \in V\\
4. \quad \text{do } A(t, c) \leftarrow \text{COMPUTEFEATUREUTILITY}(\mathcal{D}, t, c)\\
5. \quad \text{APPEND}(L, (A(t, c), t))\\
6. \quad \text{return FEATURESWITHLARGESTVALUES}(L, k)
\]
Selecting Top 2633 features by frequency

Does ignoring noisy features which do not actually matter improve the performance?