Lecture 12: Naïve Bayes Classifier, Evaluation Methods

Ling 1330/2330 Intro to Computational Linguistics
Na-Rae Han, 9/29/2020
Overview

- Text classification; Naïve Bayes classifier
  - Language and Computers: Ch.5 Classifying documents
  - NLTK book: Ch.6 Learning to classify text

- Evaluating the performance of a system
  - *Language and Computers*:
    - Ch.5.4 Measuring success, 5.4.1 Base rates
  - NLTK book: Ch.6.3 Evaluation
  - Cross-validation
  - Accuracy vs. precision vs. recall
  - F-measure
Given D, chance of Spam?

\[ P(\text{SPAM} \mid D) = \frac{P(\text{SPAM}, D)}{P(D)} = \frac{P(\text{SPAM}, D)}{P(\text{SPAM}, D) + P(\text{HAM}, D)} \]

\[ P(\text{SPAM} \mid D) \]

\[ \Leftarrow \text{The probability of a given document } D \text{ being SPAM} \]

\[ = 1 - P(\text{HAM} \mid D) \]

\[ \Leftarrow \text{Can calculate from } P(\text{SPAM}, D) \text{ and } P(\text{HAM}, D) \]
A bit of background

- **P(A)**: the probability of A occurring
  - P(SPAM): the probability of having a SPAM document.

- **P(A | B)**: Conditional probability
  - the probability of A occurring, given that B has occurred
  - P(f1 | SPAM): given a spam document, the probability of feature1 occurring.
  - P(SPAM | D): given a specific document, the probability of it being a SPAM.

- **P(A, B)**: Joint probability
  - the probability of A occurring and B occurring
  - Same as P(B, A).
  - If A and B are independent events, same as P(A) * P(B).
    - If not, same as P(A | B) * P(B) and also P(B | A) * P(A).
  - P(D, SPAM): the probability of a specific document D occurring, and it being a SPAM.
P(A, B): Joint probability

the probability of A occurring and B occurring

- Same as P(B, A).
- If A and B are independent events, same as P(A)*P(B).
  If not, same as P(A|B)*P(B) and also P(B|A)*P(A).
- P(D, SPAM): the probability of a specific document D occurring, and it being a SPAM.

Throwing two dice.
A: die 1 comes up with 6.
B: die 2 comes up with an even number.
⇒ A and B are independent.
⇒ P(A,B) = P(A) * P(B)
   = 1/6 * 1/2 = 1/12

Throwing one die.
A: die comes up with 6.
B: die comes up with an even number.
⇒ A and B are NOT independent!
⇒ P(A,B) = P(A|B) * P(B)
   = 1/3 * 1/2 = 1/6
   = P(B|A) * P(A)
   = 1 * 1/6 = 1/6
Bayes' Theorem

\[ P(B \mid A) = \frac{P(B, A)}{P(A)} = \frac{P(A \mid B) \cdot P(B)}{P(A)} \]

- B: Pitt closing, A: snowing
- P(B | A): probability of Pitt closing, given snowy weather
- P(B, A): probability of Pitt closing and snowing

\( \textit{1: the probability of Pitt closing given it's snowing is equal to the probability of Pitt closing and snowing, divided by the probability of snowing.} \)
Snow vs. Pitt, Bayes style

\[ P(B | A) = \frac{P(B, A)}{P(A)} = \frac{P(A | B) \times P(B)}{P(A)} \]

- **B**: Pitt closing, **A**: snowing
  - Last year, there were 15 snowy days; Pitt closed 4 days, 3 of which due to snow.
  - \( P(B | A) \): probability of Pitt closing, given snowy weather
    \[ = \frac{P(B, A)}{P(A)} = \frac{3/365}{15/365} = \frac{3}{15} = 0.2 \]
  - \( P(B, A) \): probability of Pitt closing and snowing
    \[ = 3/365 \]

\[ \text{the probability of Pitt closing given it's snowing is equal to the probability of Pitt closing and snowing, divided by the probability of snowing.} \]
Bayes' Theorem

\[
P(B \mid A) = \frac{P(B, A)}{P(A)} = \frac{P(A \mid B) \cdot P(B)}{P(A)}
\]

- B: Pitt closing, A: snowing
- \( P(B \mid A) \): probability of Pitt closing, given snowy weather
- \( P(B, A) \): probability of Pitt closing and snowing

\( \boxed{2} \): the probability of Pitt closing AND it's snowing is equal to the probability of Pitt closing (=prior) multiplied by the probability of snowing given that Pitt is closed.

\( \leftarrow \) Corollary of \( \boxed{1} \)! You get this by swapping A and B and solving for \( P(B, A) \)
Bayes' Theorem & spam likelihood

1. \[ P(\text{SPAM} \mid D) = \frac{P(\text{SPAM}, D)}{P(D)} = \frac{P(\text{SPAM}, D)}{P(\text{SPAM}, D) + P(\text{HAM}, D)} \]

2. \[ P(\text{SPAM}, D) = P(D \mid \text{SPAM}) \times P(\text{SPAM}) \]
   \[ = P(\text{SPAM}) \times P(D \mid \text{SPAM}) \]
   \[ = P(\text{SPAM}) \times P(f_1, f_2, \ldots, f_n \mid \text{SPAM}) \]
   \[ = P(\text{SPAM}) \times P(f_1 \mid \text{SPAM}) \times \ldots \times P(f_n \mid \text{SPAM}) \]

- SPAM: document is spam, D: a specific document occurs
- P(\text{SPAM} \mid D): probability of document being SPAM, given a particular document
- P(\text{SPAM}, D): probability of D occurring and it being SPAM
- Which means: we can calculate P(\text{SPAM} \mid D) from P(\text{SPAM}, D) and P(\text{HAM}, D), which are calculated by 2.
Probabilities of the entire document

$H_1$ "D is a SPAM" is closely related to $P(D, \text{SPAM})$:

The probability of document $D$ occurring and it being a spam

$= P(\text{SPAM}) \times P(D | \text{SPAM})$
$= P(\text{SPAM}) \times P(f_1, f_2, ..., f_n | \text{SPAM})$
$= P(\text{SPAM}) \times P(f_1 | \text{SPAM}) \times P(f_2 | \text{SPAM}) \times ... \times P(f_n | \text{SPAM})$

- We have all the pieces to compute this.
- "Bag-of-words" assumption
- "Naïve" Bayes because assumes feature independence.

- If all we're going to do is rule between SPAM and HAM, we can simply compare $P(D, \text{SPAM})$ and $P(D, \text{HAM})$ and choose one with higher probability.
- But we may also be interested in answering:

"Given $D$, what are the chances of it being a SPAM? 70%? 5%?"

← This is $P(\text{SPAM} | D)$.
Naïve Bayes Assumption

- Given a label, a set of features $f_1, f_2, \ldots, f_n$ are generated with different probabilities
- The features are independent of each other; $f_x$ occurring does not affect $f_y$ occurring, etc.

Naïve Bayes Assumption

- This feature independence assumption simplifies combining contributions of features; you just multiply their probabilities:
  \[
  P(f_1, f_2, \ldots, f_n \mid L) = P(f_1 \mid L) \cdot P(f_2 \mid L) \cdot \ldots \cdot P(f_n \mid L)
  \]

"Naïve" because features are often inter-dependent.

- $f_1$: 'contains-linguistics:YES' and $f_2$: 'contains-syntax:YES' are not independent.
Homework 4: Who Said It?

Jane Austen or Herman Melville?
- *I never met with a disposition more truly amiable.*
- *But Queequeg, do you see, was a creature in the transition stage -- neither caterpillar nor butterfly.*
- *Oh, my sweet cardinals!*

Task: build a Naïve Bayes classifier and explore it

Do three-way partition of data:
- test data
- development-test data
- training data

15,152 sents

1,000 Test

1,000 Dev-test

Training
How did the classifier do?
- **0.951 accuracy** on the test data, using a fixed random data split.

Probably outperformed your expectation.

What's behind this high accuracy? How does the NB classifier work?

⇒ **HW4 PART [B]**

How good is 0.951?
Common evaluation setups

- **Training** vs. **testing** partitions
  1. Training data → classifier is trained on this section
  2. Testing data ← classifier's performance is measured

- Training, testing, + **development-testing**
  + 3. Development testing data
  ← In feature engineering, researcher can error-analyze the data to improve performance
Cross validation

- But what if our training/testing split is somehow biased?
  - We could randomize
  - or use cross-validation.

- \textit{n-fold cross validation method}
  - Partition the data set into equally sized \textit{n} sets
  - Conduct \textit{n} rounds of training-testing, each using 1 partition as testing and the rest \textit{n-1} partitions for training
  - And then take an \underline{average} of the \textit{n} accuracy figures

\iffalse
\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{Cross_val_diagram.png}
\caption{Cross-validation diagram}
\end{figure}
\fi

\begin{itemize}
\item More reliable accuracy score. Performance evaluation is less dependent on a particular training-testing split
\item We can see how widely performance varies across different training sets
\end{itemize}
Confusion matrices

- When classifying among 3+ labels, **confusion matrices** can be informative
- **L1 classification of ESL essays:**
Accuracy as a measure

- **Accuracy**: of all labeling decisions that a classifier made, how many of them are *correct*?
  - POS tagger
  - Name gender identifier
  - `whosaid`: Austen/Melville author classifier
  - Document topic identifier
  - Movie review classifier: positive/neg. ("sentiment classifier")
Accuracy as a measure

- **Accuracy**: of all labeling decisions that a classifier made, how many of them are *correct*?

Interpreting accuracy numbers
- A movie review sentiment classifier tests 85% accurate. Is this good or bad?
  - What if it turns out 80% movie reviews are positive?
  - How about 60%?
- A document topic identifier tests 60% accurate. Good or bad?
  - What if 55% of documents are on "Politics"?
  - What if there are as many as 20 different topics, and the largest category only accounts for 10% of the data?

These questions cannot be answered without considering base probabilities (priors).
Base probabilities

- **Base probabilities (priors)**
  The probability of a randomly drawn sample to have a label $x$
  - whosaid: 'melville' has a higher prior than 'austen'
  - POS tagger: 'Noun' may have the highest prior than other tags
  - Disease test: 'Negative' is typically much higher than 'Positive'

- **Base rate neglect**
  - A cognitive bias humans have
  - We tend to assume that base probabilities are equal

- **Base performance**
  - The "bottom line" for system performances
    $\text{ex. POS tagger: if } 20\% \text{ of all words are 'Noun', then the worst-performing system can be constructed which blindly assigns 'Noun' to every word, whose accuracy is } 20\%.$
When accuracy isn't a good measure

- A medical test for a disease is 96% accurate. Good or bad?
  - What if 95% of population is free of the disease?
- A grammatical error detector is 96% accurate. Good or bad?
  - Suppose 95% of all sentences are error-free.
    - Accuracy alone doesn't tell the whole story.
- We are interested in:
  - Of all "ungrammatical" flags the system raises, what % is correct?
    - This is the precision rate.
  - Of all actual ungrammatical sentences, what % does the system correctly capture as such?
    - This is the recall rate.
Outcome of a diagnostic test

- A grammatical error detector as a diagnostic test
  - Positive: has grammatical error
  - Negative: is error-free

<table>
<thead>
<tr>
<th>Test</th>
<th>Real</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Has grammatical error</td>
<td>Is error-free</td>
</tr>
<tr>
<td>positive</td>
<td>True positives</td>
<td>False positives</td>
</tr>
<tr>
<td>negative</td>
<td>False negatives</td>
<td>True negatives</td>
</tr>
</tbody>
</table>

- Accuracy:

\[
\frac{(Tp + Tn)}{(Tp + Tn + Fp + Fn)}
\]

- When the data is predominantly error-free (high base rate), this is not a meaningful measure of system performance.
Outcome of a diagnostic test

- A grammatical error detector as a diagnostic test
  - Positive: has grammatical error
  - Negative: is error-free

<table>
<thead>
<tr>
<th>Test</th>
<th>Real</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Has grammatical error</td>
<td>Is error-free</td>
</tr>
<tr>
<td>positive</td>
<td>1 True positives</td>
<td>False positives</td>
</tr>
<tr>
<td>negative</td>
<td>False negatives</td>
<td>True negatives</td>
</tr>
</tbody>
</table>

- **Precision**: Rate of "True positives" out of all positive rulings (1)
  \[ \text{Precision} = \frac{\text{Tp}}{\text{Tp} + \text{Fp}} \]
A grammatical error detector as a diagnostic test

- Positive: has grammatical error
- Negative: is error-free

<table>
<thead>
<tr>
<th>Test</th>
<th>Real</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Has grammatical error</td>
<td>Is error-free</td>
</tr>
<tr>
<td>positive</td>
<td></td>
<td>2 True positives</td>
</tr>
<tr>
<td>negative</td>
<td>False negatives</td>
<td>True negatives</td>
</tr>
</tbody>
</table>

**Recall:**
Rate of "True positives" out of all actual positive cases (2)
\[
= \frac{Tp}{(Tp + Fn)}
\]
Precision vs. recall

- **Precision** and **recall** are in a trade-off relationship.
  - **Highly precise** grammatical error detector:
    Ignores many lower-confidence cases → drop in recall
  - **High recall** (captures as many errors as possible):
    many non-errors will also be flagged → drop in precision

- In developing a real-world application, picking the right trade-off point between the two is an important usability issue.
  - A **grammar checker** for general audience (MS-Word, etc)
    - Higher precision or higher recall?
  - Same, but for English learners.
    - Higher precision or higher recall?
**F-measure**

- **Precision and recall** are in a trade-off relationship.
  - Both measures should be taken into consideration when evaluating performance.

- **F-measure**
  - Also called F-score, F₁ score
  - An overall measure of a test's accuracy:
    - Combines precision (P) and recall (R) into a single measure
  - Harmonic mean
  - Best value: 1,
    - Worst value: 0
  - \( F_1 = \frac{2PR}{P + R} \)
    - = average if P=R,
    - < average if P and R different
Wrapping up

- HW 4 Part B due on Thu
  - Don't procrastinate! Part B is more complex.

- Next class (Thu)
  - HW4 review
  - Take a little breather.
  - No outgoing assignment!