Lecture 13: Naïve Bayes Classifier

LING 1330/2330: Introduction to Computational Linguistics
Na-Rae Han
Overview

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

- Bayes' Theorem
Automatic classification

- A **classifier** is an algorithm that processes a linguistic input and assigns it a **class** from a user-defined set.
  - It usually denotes a **statistical** model induced through **machine learning**.
  - The algorithm works off a set of **weighted contextual features**.

- What are examples of classifiers?
# Classifier examples

<table>
<thead>
<tr>
<th>Example</th>
<th>Unit of linguistic input</th>
<th>What type of classes</th>
<th>Class labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>POS-tagging a text</td>
<td>word</td>
<td>Part-of-speech</td>
<td>Noun, Verb, Adj, Det...</td>
</tr>
<tr>
<td>Grammar error checker</td>
<td>sentence</td>
<td>Grammaticality</td>
<td>grammatical/not</td>
</tr>
<tr>
<td>Pronoun resolution</td>
<td>NP + pronoun</td>
<td>Co-reference</td>
<td>co-refers/not</td>
</tr>
<tr>
<td>Spam filter</td>
<td>document (email)</td>
<td>Spam or not</td>
<td>Spam/Ham</td>
</tr>
<tr>
<td>Language identifier</td>
<td>document</td>
<td>Which language</td>
<td>ENG, SPA, FRN, JAP, CHI, KOR, ...</td>
</tr>
<tr>
<td>Sentiment analysis</td>
<td>document</td>
<td>What &quot;sentiment&quot;</td>
<td>Positive, (neutral), negative, ...</td>
</tr>
<tr>
<td>Automatic essay grader</td>
<td>document (essay)</td>
<td>Quality of writing</td>
<td>5, 4, 3, 2, 1, 0</td>
</tr>
<tr>
<td>Military intelligence</td>
<td>document (message)</td>
<td>Threat assessment</td>
<td>Contains a threat/not</td>
</tr>
</tbody>
</table>

4/6/2017
Example: name and gender

- Are the following first names **male** or **female**?
  - James, Elizabeth, Hillary, Dana, McKayla
  - Joffrey, Tyrion, Arya
  - Kimiko, Tae-hyun, Na-Rae

- Not knowing the actual referent, a speaker of the language nevertheless *guesses* the gender, often with high accuracy
  - Guessing based on what?
    - Based on the presence/absence of certain **observable features**.

- We can build a **model** that replicates the above human cognitive process.
Boy or girl? A Naïve-Bayes classifier

```python
>>> boyorgirl = nltk.NaiveBayesClassifier.train(train_set)
>>> gender_features('Neo')
{'first_letter': 'N', 'last_letter': 'o'}
>>> boyorgirl.classify(gender_features('Neo'))
'male'
>>> boyorgirl.classify(gender_features('Arya'))
'female'
>>> boyorgirl.classify(gender_features('Na-Rae'))
'female'
>>> nltk.classify.accuracy(boyorgirl, test_set)
0.768
```

- Classifier built from NLTK's names corpus
- Uses two features: first letter, last letter
- See [Lab19-shell.txt](http://nltk.org/book/ch06.html#gender-identification) and also NLTK book section
Example: movie reviews

- Classify each document as "positive" or "negative"
  ➡️ A type of sentiment analysis
- What "features" can we use?
  - Words themselves
  - N-grams, length, ...

all of this, of course, leads up to the predictable climax. But as foreseeable as the ending is, it's so damn cute and well-done that I doubt any movie in the entire year contains a scene the evokes as much pure joy as this part does. When Ryan discovers the true identity of her online love, I was filled with such, for lack of a better word, happiness that for the first time all year, I actually left the theater smiling.

the acting is below average, even from the likes of Curtis.. Sutherland is wasted and Baldwin, well, he's acting like a Baldwin, of course. The real star here are Stan Winston's robot design, some schnazzy CGI, and the occasional good gore shot, like picking into someone's brain. So, if robots and body parts really turn you on, here's your movie. Otherwise, it's pretty much a sunken ship of a movie.
Document classification is an example of computer science engineering called **machine learning**

- Just like humans learn from "experience", a computer algorithm *learns* from data
  - Learns what exactly? → Statistical patterns
- Machine learning is not limited to linguistic data
  - Example?

**Machine learning requires:**
- *Training* data, often lots of them
- *Testing* data for evaluation
  (Also: sometimes *development test data* for error analysis)
Machine learning (supervised)

Source: NLTK book
Features and evidence

- A classification decision must rely on some observable evidence \(\rightarrow\) features
  - Female or male names?
    - The last letter of the name: 'a', 'k', etc.
  - What POS is *park*? What about *carbingly*?
    - Is *park* preceded by *the*? *to*?
    - Does *carbingly* end with 'ly'? 'ness'?
  - Is this document SPAM or HAM?
    - Does it contain the word *enlargement*?
    - Does it contain *linguistics*?
Feature engineering

- Deciding what features are relevant. Two types:
  - **Kitchen sink** strategy
    - Throw a set of features to the machine learning algorithm, see what features are given greater weight and what gets ignored
    - Example: using every word in a document as a feature:
      - 'has-cash': True, 'has-the': True, 'has-linguistics': False, ..
  - **Hand-crafted** strategy
    - Utilizing expert knowledge, determine a small set of features that are likely to be relevant
    - Example: grammatical error detection
      - For each sentence, determine grammatical/ungrammatical
      - Hand-coded features:
        - Subject-verb agreement, fragment or not, etc.
Weighting the evidence

- A classification decision involves reconciling multiple features with different levels of predictive power.

Different types of classifiers use different algorithms for:

1. Determining the **weights of individual features** in order to maximize its labeling success in the training data
2. When given an input, using the feature weights to **compute the likelihood of a label**

Popular machine learning methods:

- Naïve Bayes
- Decision tree
- Maximum entropy (ME)
- Hidden Markov model (HMM)
- Neural network → Deep learning (!!)
- Support vector machine (SVM)
A spam filter as a Naïve Bayes model

- **Unit of linguistic input:**
  - A document (email text)
  - **Features:** Words in document (kitchen sink strategy)
  - A document is reduced to **the set of words it contains**
    ⇢ "Bag of words" assumption

- **Classifier type:**
  - Naïve Bayes

- **Class labels:**
  - SPAM/HAM
The Naïve Bayes Classifier

- A rough sketch of a Naïve Bayes algorithm:
  - Given an email document (D), process each piece of evidence (f₁, f₂, f₃... fₙ) to support the two hypothesis:
    - H₁: D is a SPAM.
    - H₂: D is a HAM.
  1. It starts with the **base probabilities**, also known as **priors**.

  Suppose 70% of all email in the training data is SPAM:
  - H₁: D is a SPAM: 70%
  - H₂: D is a HAM: 30%.

  2. Each piece of evidence (=feature) will strengthen one hypothesis and weaken the other.

    - 'contains-cash:YES' \(\Rightarrow\) H₁ is now 85%, H₂ 15%.

  3. Repeat 2. for all features.

  4. When done, rule for hypothesis with higher probability.
Inducing feature weights

- But how is the feature weight determined?

  **f1: 'contains-cash:YES'**

  \[ P(f1|SPAM) \] = the probability of 'cash' occurring, given a spam document

  \[ P(f1|HAM) \] = the probability of 'cash' occurring, given a ham document

- In the training data:

<table>
<thead>
<tr>
<th></th>
<th>SPAM</th>
<th>HAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>day</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>linguistics</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>cash</td>
<td>90</td>
<td>3</td>
</tr>
<tr>
<td>Viagra</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>the</td>
<td>138</td>
<td>60</td>
</tr>
</tbody>
</table>

| TOTAL # of docs | 140 | 60 |

\[ \frac{1}{2} = \frac{90}{140} = 0.64 \]

\[ \frac{3}{60} = 0.05 \]

SPAM-to-HAM **odds ratio** of f1:

\[ \frac{1}{2} = 12.8 \] \( \leftarrow \) Feature strength

How about **linguistics** (f2)?

\[ P(f2|SPAM) = 1/140 = 0.007 \]

\[ P(f2|HAM) = 15/60 = 0.25 \]

HAM-to-SPAM **odds ratio**: **35.7**
Smoothing

- We have to account for cases where a feature is never observed with a label.

<table>
<thead>
<tr>
<th></th>
<th>SPAM</th>
<th>HAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>day</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>linguistics</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>cash</td>
<td>90</td>
<td>3</td>
</tr>
<tr>
<td>Viagra</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>the</td>
<td>138</td>
<td>60</td>
</tr>
<tr>
<td>TOTAL # of docs</td>
<td>140</td>
<td>60</td>
</tr>
</tbody>
</table>

- 'Viagra' never occurred in a HAM document.
  - P(f3 | SPAM) = 20/140 = 0.14
  - P(f3 | HAM) = 0/60 = 0
  - SPAM-HAM odds ratio = 0.14/0  !!!
  - HAM-SPAM odds ratio = 0/0.14 = 0
     Not good, because it single-handedly renders P(HAM | D) to 0, regardless of what other features are present

- Just because we haven't seen a feature/label combination occur in the training set, doesn't mean it's impossible for that combination to occur.

- **Smoothing** is a process through which a very small probability is assigned to such features.
H₁ "D is a SPAM" is closely related to $P(D, \text{SPAM})$:
The probability of document D occurring and it being a spam

$$= P(\text{SPAM}) \times P(D\mid\text{SPAM}) = P(\text{SPAM}) \times P(f₁\mid\text{SPAM}) \times P(f₂\mid\text{SPAM}) \times \ldots \times P(fₙ\mid\text{SPAM})$$

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

Why is this assumption naïve/unreasonable?
- 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 $P(\text{SPAM}\mid D)$. 


A bit of background

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

- **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.
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

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

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

- **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

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

- **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

\[ \text{Corollary of } \textit{1: } \text{the probability of Pitt closing AND it's snowing is equal to the probability of Pitt closing (}=\text{prior}) \text{ multiplied by the probability of snowing given that Pitt is closed.} \]

\[ \text{Corollary of } \textit{1! You get this by swapping A and B and solving for P(B,A)} \]
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:

$$P(f_1, f_2, \ldots, f_n | L) = P(f_1 | L) * P(f_2 | L) * \ldots * P(f_n | L)$$

"Naïve" because features are often inter-dependent.

$\text{f1:'contains-linguistics:YES'}$ and $\text{f2:'contains-syntax:YES'}$ are not independent.
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.
Homework #10: 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
  - [http://www.pitt.edu/~naraehan/ling1330/hw10.html](http://www.pitt.edu/~naraehan/ling1330/hw10.html)

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

- HW 10 out
  - Week-long: 1/2 due Tuesday, and the rest next Thursday
  - *Don't procrastinate* – start now!
  - Recitation students: work on PART A before tomorrow.

- Next class (Tue)
  - Evaluating performance of a classifier

- Interesting talks ([details here](#))
  - Allison Hegel, "Context at Scale: Text Analysis of Amateur and Professional Reviews" TODAY 3-4:30pm, 501 CL
  - Allison Hegel, Hands-on Workshop, Collocations Analysis, April 7 (Fri) 10-11am, 435 Cathedral of Learning