Sentiment analysis

CS440

Positive or negative movie review?

- unbelievably disappointing
- Full of zany characters and richly applied satire, and some great plot twists
- this is the greatest screwball comedy ever filmed
- It was pathetic. The worst part about it was the boxing scenes.
Twitter sentiment versus Gallup Poll of Consumer Confidence


Twitter sentiment

Bollen et al. (2011)
- CALM is predictive of DJIA 3 days later

Why sentiment analysis?
- **Movie**: is this review positive or negative?
- **Products**: what do people think about the new iPhone?
- **Public sentiment**: how is consumer confidence?
- **Politics**: what do people think about this candidate or issue?
- **Prediction**: predict election outcomes or market trends from sentiment
Sentiment Analysis

• Sentiment analysis is the detection of **attitudes**
  “enduring, affectively colored beliefs, dispositions towards objects or persons”

  **Type of attitude**
  • From a set of types
    – *Like, love, hate, value, desire,* etc.
  • Or (more commonly) simple weighted **polarity**:
    – *positive, negative, neutral,* together with **strength**

  **Text** containing the attitude
  • Sentence or entire document

---

Sentiment Analysis

• Simplest task:
  – Is the attitude of this text positive or negative?

• More complex:
  – Rank the attitude of this text from 1 to 5

• Advanced:
  – Detect complex attitude types
Sentiment Classification in Movie Reviews


• Polarity detection:
  – Is an IMDB movie review positive or negative?

• Data: Polarity Data 2.0:
  – http://www.cs.cornell.edu/people/pabo/movie-review-data

IMDB data in the Pang and Lee database

✓

when _star wars_ came out some twenty years ago , the image of traveling throughout the stars has become a commonplace image . […] when han solo goes light speed , the stars change to bright lines , going towards the viewer in lines that converge at an invisible point . cool .

✗

“snake eyes” is the most aggravating kind of movie ; the kind that shows so much potential then becomes unbelievably disappointing . it’s not just because this is a brian depalma film , and since he’s a great director and one who’s films are always greeted with at least some fanfare . and it’s not even because this was a film starring nicolas cage and since he gives a brauvara performance , this film is hardly worth his talents .
Baseline algorithm

- Tokenization
- Feature extraction
- Classification using different classifiers
  - Naïve Bayes
  - MaxEnt
  - SVM

Sentiment tokenization

- Deal with HTML and XML markup
- Twitter mark-up (names, hash tags)
- Capitalization (preserve for words in all caps)
- Phone numbers, dates
- Emoticons
- Useful code:
  - Christopher Potts sentiment tokenizer [http://sentiment.christopherpotts.net/](http://sentiment.christopherpotts.net/)
  - Brendan O’Connor twitter tokenizer [https://github.com/brendano/tweetmotif](https://github.com/brendano/tweetmotif)
Extracting features for sentiment classification

• How to handle negation
  – I didn’t like this movie
  vs
  – I really like this movie

• Which words to use?
  – Only adjectives
  – All words
    • All words turns out to work better, at least on this data

Negation


Add NOT_ to every word between negation and following punctuation:

didn’t like this movie, but I

didn’t NOT_like NOT_this NOT_movie but I
Reminder: Naïve Bayes

\[ c_{NB} = \arg\max_{c_j \in C} P(c_j) \prod_{i \in \text{positions}} P(w_i | c_j) \]

\[ \hat{P}(w_i | c) = \frac{\text{count}(w_i, c) + 1}{\sum_{w \in V} \text{count}(w, c) + |V|} \]

Binarized (Boolean feature) Multinomial Naïve Bayes

- Intuition:
  - For sentiment (and probably for other text classification domains)
  - Word occurrence may matter more than word frequency
    - The occurrence of the word *fantastic* tells us a lot
    - The fact that it occurs 5 times may not tell us much more.
  - Boolean Multinomial Naïve Bayes
    - Clips all the word counts in each document at 1
Boolean Multinomial Naïve Bayes: Learning

- From training corpus, extract *Vocabulary*

- Calculate $P(c_j)$ terms
  - For each $c_j$ in C do
    - $docs_j ←$ all docs with class $= c_j$
    - $P(c_j) ← \frac{|docs_j|}{|total\#\ documents|}$

- Calculate $P(w_k \mid c_j)$ terms
  - For $w_k$ in Vocabulary
    - $n_k = # of\ occurrences\ of\ w_k\ in\ docs_j$
    - $P(w_k \mid c_j) ← \frac{n_k + \alpha}{n + \alpha \mid V|}$

- Remove duplicates in each doc:
  - For each word type $w$ in doc
    - Retain only a single instance of $w$

Boolean Multinomial Naïve Bayes on a test document $d$

- First remove all duplicate words from $d$
- Then compute NB using the same equation:

$$c_{NB} = \arg\max_{c_j \in C} P(c_j) \prod_{i \in positions} P(w_i \mid c_j)$$
## Normal vs. Boolean Multinomial NB

<table>
<thead>
<tr>
<th>Normal</th>
<th>Doc</th>
<th>Words</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>1</td>
<td>Chinese Beijing Chinese</td>
<td>c</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Chinese Chinese Shanghai</td>
<td>c</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Chinese Macao</td>
<td>c</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>Tokyo Japan Chinese</td>
<td>j</td>
</tr>
<tr>
<td>Test</td>
<td>5</td>
<td>Chinese Chinese Chinese Tokyo Japan</td>
<td>?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Boolean</th>
<th>Doc</th>
<th>Words</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>1</td>
<td>Chinese Beijing</td>
<td>c</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Chinese Shanghai</td>
<td>c</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Chinese Macao</td>
<td>c</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>Tokyo Japan Chinese</td>
<td>j</td>
</tr>
<tr>
<td>Test</td>
<td>5</td>
<td>Chinese Tokyo Japan</td>
<td>?</td>
</tr>
</tbody>
</table>

## Binarized (Boolean feature) Multinomial Naïve Bayes


- Binary seems to work better than full word counts
  - This is not the same as Bernoulli Naïve Bayes
- BNB doesn’t work well for sentiment or other text tasks
Problems: 
What makes reviews hard to classify?

• Subtlety:
  – Perfume review in *Perfumes: the Guide*:
    “If you are reading this because it is your darling fragrance, please wear it at home exclusively, and tape the windows shut.”

Thwarted expectations and ordering effects

• “This film should be brilliant. It sounds like a great plot, the actors are first grade, and the supporting cast is good as well, and Stallone is attempting to deliver a good performance. However, it can’t hold up.”

• Well as usual Keanu Reeves is nothing special, but surprisingly, the very talented Laurence Fishbourne is not so good either, I was surprised.
Lexicons: annotating words for their sentiment

- There are many resources that provide annotations of words and their associated sentiment.

SentiWordNet


Home page: [https://github.com/aesuli/sentiwordnet](https://github.com/aesuli/sentiwordnet)

All elements in WordNet automatically annotated for degrees of positivity, negativity, and neutrality/objectiveness.
LIWC (Linguistic Inquiry and Word Count)


- Home page: http://www.liwc.net/
- 2300 words, >70 classes
- **Affective Processes**
  - negative emotion (bad, weird, hate, problem, tough)
  - positive emotion (love, nice, sweet)
- **Cognitive Processes**
  - Tentative (maybe, perhaps, guess), Inhibition (block, constraint)
- **Pronouns, Negation** (no, never), **Quantifiers** (few, many)

MPQA Subjectivity Cues Lexicon


- Home page https://mpqa.cs.pitt.edu/lexicons/subj_lexicon/
- 6885 words from
  - 2718 positive
  - 4912 negative
- Each word annotated for intensity (strong, weak)
- GNU GPL
How would you use lexicons to predict sentiment?

Lexicons for detecting document affect: simple unsupervised method

\[
\begin{align*}
  f^+ &= \sum_{w \text{ s.t. } w \in \text{positive lexicon}} \theta_w^+ \cdot \text{count}(w) \\
  f^- &= \sum_{w \text{ s.t. } w \in \text{negative lexicon}} \theta_w^- \cdot \text{count}(w)
\end{align*}
\]

\[
\text{Sentiment} = + \quad \text{if} \quad f^+ > f^-
\]
How to deal with stars?

1. Treat as a classification problem
2. Use regression or ordinal regression

Summary

• Generally modeled as classification or regression task
• Comments:
  – Negation is important
  – Using all words (in Naïve Bayes) works well for some tasks
  – Finding subsets of words may help in other tasks
  – Hand-built polarity lexicons
• Naïve Bayes is a good baseline, but other classifiers typically work better