Machine Learning in Business
John C. Hull

Chapter 8

Natural Language Processing

Sentiment Analysis

- Sentiment analysis is the processing of textual data from surveys and social media to determine whether the market's opinion about something is positive or negative
- Can be done in real time
- Possible business applications:
  - Coca Cola's new formula
  - Gillette's new advertisement (the best men can be)
  - United Airline's PR disaster when it pulled someone off its plane

A trading strategy?

- Buy stocks with a positive sentiment
- Short stocks with a negative sentiment
- Zhang and Skiena (2010) found this to be profitable, but if markets are efficient it is likely to be less profitable today

Obtaining Labeled Data for Sentiment Analysis

- We need text that has been classified as according to whether it is positive or negative
- There are publicly available data sets that have been classified
- Movie reviews are sometimes used because they are given between one and five stars
- Alternatively it is necessary to collect past opinions and use human beings to classify them
- Note: human beings only agree about 80% of the time and so there are limits on the accuracy of NLP procedures

Pre-processing

To obtain a “vocabulary” from data, the following can be useful:

- Word tokenization
- Remove punctuation
- Remove stop words
- Stemming
- Lemmatization
- Correct spelling mistakes
- Recognize abbreviations
- Remove rare words

Bag-of-Words Model

- Uses words to analyze opinions without regard to the order in which they appear
- We might have a vocabulary of 10,000 words and a bag-of-words model will list the number of times each word occurs in an opinion

A Simple Approach

- Make a list of positive and negative words and count the number of times that each appear
- But there is no learning in this approach

Using ML

- Approaches using ML use labeled data and divide it into training set, test set, and (possibly validation set)
- The general approach is the same as in other ML applications
- The number of features (i.e., number of words) is large
- Two possibilities:
  - Base analysis on whether a word appears or not
  - Base analysis on the number of times a word appears
- The evidence indicates that multiple appearances of a word do not necessarily give more information than a single occurrence

Using Naïve Bayes Classifier

- If word j is in an opinion, define p_j as the probability that an opinion in the training set is positive when word j appears and q_j as the probability that it is negative when word j appears
- If word j is not in an opinion define p_j as the probability that an opinion in the training set is positive when word j does not appear and q_j as the probability that it is negative when word j does not appear

Using Naïve Bayes Classifier *continued*

$$\text{Prob(Positive|words)} = \frac{p_1 p_2 \dots p_m}{\text{Prob(words)}} \text{Prob (Positive)}$$

$$\text{Prob(Negative|words)} = \frac{q_1 q_2 \dots q_m}{\text{Prob(words)}} \text{Prob (Negative)}$$

$$\text{Prob(Positive|words)} = \frac{p_1 p_2 \dots p_m \times \text{Prob (Positive)}}{p_1 p_2 \dots p_m \times \text{Prob (Positive)} + q_1 q_2 \dots q_m \times \text{Prob (Negative)}}$$

$$\text{Prob(Negative|words)} = \frac{q_1 q_2 \dots q_m \times \text{Prob (Negative)}}{p_1 p_2 \dots p_m \times \text{Prob (Positive)} + q_1 q_2 \dots q_m \times \text{Prob (Negative)}}$$

Laplace Smoothing

- If any of the *p*'s are zero the probability that the opinion is positive is zero
- If any of the *q*'s are zero the probability that the opinion is negative is zero
- To avoid these extreme results we can add a small amount of imaginary data to avoid the zeroes
- This is known as Laplace smoothing

Other Algorithms

- The Naïve Bayes Classifier assumes conditional independence
- Other algorithms that can be used are
  - SVM
  - Logistic regression
  - Decision trees
  - Neural networks

Unigrams, bigrams, etc

- So far we have assumed that the bag-of-words model considers single words (unigrams)
- This would potentially misclassify an opinion such as “This product was not good”
- An alternative is to consider two words at a time (bigrams). This can work better with opinions that contain negative words
- We can even go one step further and consider three words at a time (trigrams). This might get a opinion “The product was not too bad” classified correctly.

Information Retrieval

- How can a search engine find the best document given certain search words
- We can define two measures:
  - Term frequency (TF): This is a function of (a) a search word and (b) a document that might be chosen. It is the number of times the word appears in the document divided by number of words in the document.
  - Inverse document frequency (IDF): This is a function of a search word. It is the logarithm of number of documents divided by number of documents containing the word.
- TF-IDF is the product of the two measures
- For each document we calculate the sum of the TF-IDFs across the search words. This is used as a measure of the relevance of the document

Word Vectors

- Two words have similar meanings if they tend to occur close to the same other words.
- We can define close as “within five words”
- This can lead to a 10,000 by 10,000 table of probabilities
- Using an autoencoder-type procedure it can be reduced to a 10,000 by 300 table (or even a 10,000 by 100 table)
- This means that each word is defined by a 300-long (or 100-long) vector of numbers
- We find that the vectors have certain (approximate) properties, e.g.   King – Man + Woman = Queen

Another application of NLP

- What is the probability of a particular word sequence?
- We might determine this by considering how often each consecutive pair of words in the sequence occurs in the training set
- This is useful for
  - Translating from one language to another
  - Speech recognition
  - Summarizing texts
  - Conversion of speech to text