Assignment 1: Classification of review sentiments

Barbara Engelhardt, Princeton University out 02/06/2018; due 02/27/18

Background

Many popular online sites gather reviews from users: Yelp, Netflix, IMDB, and Amazon are well-known examples. While these websites compile numeric ratings about their products, the text reviews reveal additional structure and rationale behind those numeric ratings. Sentiment analysis, or the classification of text snippets into—here—positive and negative reviews helps to shed light on these numbers, for example, understanding whether most users were lukewarm about a 6.9/10 movie, or whether opinions were sharply bimodal (i.e., "loved it or hated it," the canonical example of which is *Napoleon Dynamite*). Sentiment analysis has been used in Facebook and Twitter feeds to characterize the social media posts and understand the propagation of sentiments among social networks, with Facebook famously publishing a paper describing the manipulation of users' sentiments by filtering and ordering their feeds using posts of specific sentiment types. A number of difficulties in this domain exist, including these reviews and posts being often very short, using vocabulary that is not available in standard English dictionaries, using non-standard negation patterns in sentences, and using irony or sarcasm.

Project definition

Your goal in this homework project is to use a data set consisting of 3,000 reviews to build a sentiment classifier, or software that classifies a review as *positive* or *negative*. We have supplied you with the training and test review data sets on the Piazza website and also a Python script to identify a simple dictionary of words from the review, creating, for each review, a bag-of-words representation using the dictionary words. Your first step in the process is to download the script and the data and run the script on the training data to build a vocabulary and create a bag-of-words feature representation for each review; see the `readme.txt` file in Piazza/homework1/ folder for the initial steps in the process. Feel free to extend the feature set in interesting and well-motivated ways (see *Extensions*, below).

Then, you should build (multiple) classifiers that take in the feature sets and the classifications of those feature sets and fit a classifier. Feel free to use the classifiers we have or will discuss in class as well as others mentioned in our text books, described in the scientific literature, or implemented in software. You may also use more sophisticated classifiers (see *Extensions*) specifically built for the problem of sentiment analysis. Because of the large number of possible features, we recommend using some type of feature selection to reduce the number of features. Finally, you should evaluate the classifiers you apply to this problem according to

(at a minimum) the Receiver Operating Characteristic (ROC) curves on the test data set, which consists of 600 held out reviews.

Essential to any data analysis task is the interpretation of the results. What features were most important for sentiment classification, and what do these features tell us about the problem? What is worse: classifying a negative review as positive, or classifying a positive review as negative? What types of reviews were easy to classify for all approaches, and on what types of reviews did the approaches disagree? Simply building a machine learning approach to solve the problem does not constitute a data analysis; recovering and characterizing signal from these results does, viewed through the lens of the methodological assumptions.

Deliverables

Your deliverables for this project include:

- A five page (not including citations) summary of the project work, which should contain (as described in the Example project write up on Piazza):
  - A title, authors' names, and abstract for the project;
  - an introduction to the problem being addressed;
  - a description of the data;
  - a clear description of the methods used, and how they were fitted using training data;
  - a one page description of one of the classifiers, starting from first principles and ending with how the method was fit to the data;
  - a presentation of the results of the methods applied to the test data;
  - a discussion of the results, including specific examples of reviews and features that highlight the behavior of the classification models;
  - a short summary and conclusion, including extensions that you believe would be particularly valuable based on the results;
  - a *complete* bibliography to support the databases, feature selection, classifiers, code bases, and related work that are relevant to your project.
- A .zip file of your Python code that you developed for the project, with a README about how you ran the code.

Please put your PDF write up of the project, the Python code for the project, and a readme about how to run the code into
`https://dropbox.cs.princeton.edu/COS424_S2018/Assignment1`
by midnight on the assignment due date, with the file names
`<author1PUID>_<author2PUID>_hw1.pdf` (for write up)
`<author1PUID>_<author2PUID>_hw1.zip` (for code and readme).
Please only submit one PDF per pair of authors.

We strongly recommend *writing as you go* in the project, which means starting to write the project report as you are downloading and analyzing the data. That said, you should avoid speculative writing, and only write about results once you have them in hand.

Extensions

If you would like to extend this assignment to more interesting ground after first completing the basic deliverables for the project, you might consider the following:

- *Extend the data set*: The reviews compiled here represent a very small training set for a text classification task. There are a number of publicly available sentiment classification corpora with fully labeled data. Processing and incorporating other data sets—including ones you personally compile—and releasing these data with appropriate permissions would be worthwhile. In these data, we have separated out training and test data, but feel free to use K-fold cross validation on the larger data sets if you would like. One such data set includes 1.5M tweets with positive/negative labels: `http://thinknook.com/twitter-sentiment-analysis-training-corpus-dataset-2012-09-22/`
- *More interesting features*: while we have only asked you to work with a simple word dictionary, there are many extensions to this to consider, including features involving:
  - bigrams, punctuation, proper nouns, negations, capitalizations
  - review text length and distribution of length
  - analysis of non English words, such as URLs
- *More complex classifiers*: there are a number of exciting classifiers that might be used for this task, including, e.g., supervised topic models [Zhu et al., 2009, Mcauliffe and Blei, 2008], conditional tensor factorization [Yang and Dunson, 2013], or something of your own design that might identify latent structure in the data that is predictive of sentiment. *Ensemble classifiers* that combine sentiment classifications from a number of classifiers to improve results may be built from a number of the more simple classifiers used in your basic analyses.
- *More classes of sentiments*: There are many more sentiments other than *Positive* and *Negative*. How would you include those here?
- *Better evaluation metrics*: what are better metrics that you might use to evaluate these classifiers? How can classification uncertainty be considered in these metrics? Can you improve model evaluation using cross validation instead of our simple training and test sets?
- *Additional types of problems*: What about reviews that do not have class labels? You might consider developing an active learning method that will ask users to classify reviews as

positive/negative that will, in expectation, reduce uncertainty maximally across all unlabeled reviews. You also might try to develop adaptive sentiment classifiers that can be refitted as new types of sentiments, reviews, or vocabulary words arise?

Resources

There is a large literature on sentiment classifiers. Many involve fairly simple classification methods and large numbers of features or reference data sets. There are also a number of reviews available. Review some of this literature to get ideas on ways to create a really great classifier.

References

Jon D Mcauliffe and David M Blei. Supervised topic models. In *Advances in neural information processing systems*, pages 121–128, 2008.

Yun Yang and David B Dunson. Bayesian conditional tensor factorizations for high-dimensional classification. *arXiv preprint arXiv:1301.4950*, 2013.

Jun Zhu, Amr Ahmed, and Eric P Xing. Medlda: maximum margin supervised topic models for regression and classification. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1257–1264. ACM, 2009.