CS224U Literature Review Ben Peloquin

A note on the project:

For my project I've been collecting a data set of 50K beer reviews scraped from RateBeer.com. The following set of reviews represent two different possible directions I could take this project.

The first makes use of the multiple dimensions of ratings included with each review. In addition to an overall score, reviewers explicitly give ratings on four dimensions - *taste*, *aroma*, *palate* and *appearance*. This first direction would consist of an exploration of supervised learning along these different aspects. Do we find that that prediction is more difficult between the aspects? What kind of hypotheses can we encode through feature engineering that aid in this task? Relatedly, can we recover the different aspects being talked about using techniques such as topic modeling? Along these lines I review the seminal work in sentiment classification by Pang, Lee & Viathyanathan (2002), as well as work by McAuely, Leskovec & Jurafsky (2012) who looked at multi-aspect classification using RateBeer.com data.

Another direction I might take the project makes use of the unique information I'm collecting. Previous RateBeer.com investigations focused on *beer level* information as opposed to *user level* information. The reason why I'm scraping this data set myself is so I can also collect user information including the number of beers, countries and locations the user has reviewed, as well as their country of origin. Given this user-specific level data we can investigate hypothesis about the influence of experience and expertise on language use. Accordingly, the second set of reviews includes work examining expertise detection and expertise effects on language use.

Here's an example of some user data I've been scraping and will be working with for this project:

https://github.com/benpeloquin7/rateBeerLingRel/blob/master/data/temp_store/101257.csv

"Thumbs Up? Sentiment Classification using Machine Learning Techniques"
Pang, Lee & Viathyanathan (2002)

Pang, Lee & Viathyanathan's (2002) seminal work on sentiment classification applied three machine-learning methods to determine positive and negative sentiment in movie review data. While the methodology used in the paper may seem relatively simple today – training three machine learning (ML) algorithms with limited feature engineering – in 2002 this represented a novel application of existing classification methods to a new domain.

At the time, previous work in machine learning (ML) classification had focused on topic categorization. That is, researchers attempted to sort documents according to their subject matter using ML methods such as Naïve Bayes, MaxEnt and SVM classifiers. With the rising popularity of review sites such as Rotten Tomatoes and IMDB in which reviews are paired with ratings, there was an abundance of labeled data available for supervised learning tasks to deal with sentiment directly.

The authors chose to focus on the domain of classifying sentiment in movie reviews. Using data from the Internet Movie Database (IMDb) archive, the author's selected reviews in which the author explicitly gave a star rating, converting the ratings into one of three categories: positive, neutral and negative. In total they collected a corpus of 752 negative and 1301 positive reviews from a total of 144 reviewers. In terms of text normalization the authors did not use stemming or stoplists. However, in order to address negation they did introduce a "_NOT" tag to every word between a negation word ("not", "isn't", etc) for unigram features.

The chief objective of this project was to *explore* the "difficulty" of sentiment classification and lay the groundwork for future optimization. "Difficulty" was relative to performance using similar ML algorithms for topic categorization. For a baseline algorithm the author's used human generated sentiment lexicons to make classifications based solely on the counts of items from each lexicon present in a given document. Baseline performance reached 58% and 64% on a data set in which random choice performance would achieve 50% (data set split 50% positive and negative). The author's extended this human-generated baseline by directly examining the test set and creating 7-word positive and negative lexicons improving baseline accuracy to 69%.

Following these baselines the author's investigate performance for a Naïve Bayes (NB) classifier, a MaxEnt (ME) classifier and SVM. Performance was assessed across these three learners as well as a number of feature dimensions including unigrams, binarized unigrams, unigrams + bigrams, bigrams, unigrams + POS tags, adjectives, top 2633 unigrams, and unigrams + position information. Due to efficiency concerns the authors imposed a cutoff of at least four occurrences for all unigram and bigram features.

Accuracy assessed via three-fold cross-validation for all the learners was well above the random choice baseline and also outperformed the human-selected-unigram feature baseline (81%, 80% and 83% for binarized unigrams for NB, ME, and SVM, respectively). While a distinct improvement over baseline, these numbers were still significantly lower than the 90% accuracies achieved in topic categorization tasks using similar features and learning algorithms. The author's argued this provided evidence that sentiment classification was more "difficult" than topic classification.

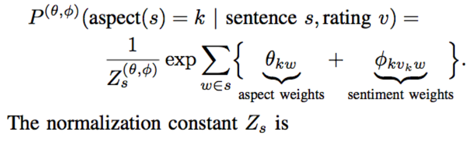
The result of this paper was a framework for addressing a new problem – sentiment classification using ML methods. While model performance did not reach levels achieved in topic classification settings, the author's showed that learning algorithms could provide a significant improvement over human-based-lexicon classifiers. Furthermore, the author's demonstrated potential in this domain – both because of the easily accessible labeled data as well as the room for improvement either through improved feature engineering or learning methods. Any project investigating supervised learning algorithms to recover sentiment or sentiment like aspects from ratings can trace itself back to this influential work.

"Learning Attitudes and Attributes from Multi-Aspect Reviews"
McAuely, Leskovec & Jurafsky (2012)

In "Learning Attitudes and Attributes from Multi-Aspect Reviews" the authors attempt to move beyond basic sentiment classification of online review text into the realm of automatically learning the *aspects* that contribute to the reviews. To do so, they consider product rating systems in which aspects are made explicit. That is, users give both overall ratings of the products as well as ratings among product dimensions. As an example, a single review by a user on the site RateBeer.com contains a single overall score as wells as scores for *taste, aroma, appearance and palate.*

There are several significant contributions from this work. The authors introduce a new data set of approximately five million reviews from different review sites that include multiple aspect ratings with over ten thousand manually annotated sentences. While there has been previous work focused on multi-aspect segmentation and summarization, those studies have dealt mostly with data sets of tens of thousands of observations. Consequently, the author's introduce a new model *Preference and Attribute Learning from Labeled Groundtruth and Explicit Ratings* (PALE LAGER), which can handle data sets in the millions of observations and under a variety of training scenarios including unsupervised, weakly supervised and fully supervised. PALE LAGER represents a unique approach to classification, separately modeling words that discuss aspect and words that discuss sentiment about an aspect. Unlike topic modeling, in which topics are represented as distributions over words, the PALE LAGER model learns sentiment neutral lexicons of words that describe an aspect (mostly nouns) as well as sentiment lexicons for each aspect (mostly adjectives).

PALE LAGER models aspects and aspect ratings as a function of words at the sentence level. They assume that each sentence in a review discusses a single aspect. This could of course be extended to the word, or paragraph level, however this design choice matched previous work. In order to differentiate words that describe aspect from words that discuss the related sentiment, they separate the model into two parameter vectors, θ, ϕ which encode this distinction. In their model the probability that a sentence s discusses a particular aspect k, given the ratings v associated with a review is:



$$P^{(\theta,\phi)}(\text{aspect}(s) = k \mid \text{sentence } s, \text{rating } v) = \frac{1}{Z_s^{(\theta,\phi)}} \exp \sum_{w \in s} \Big\{ \underbrace{\theta_{kw}}_{\text{aspect weights}} + \underbrace{\phi_{kv_k w}}_{\text{sentiment weights}} \Big\}.$$

The normalization constant Z_s is

The authors explore three learning schemes using increasing levels of supervision in the form of sentence labels. While increased supervision lead to better results overall, the authors argue that unsupervised results were also promising.

They evaluated the model on three tasks: segmentation, summarization and rating prediction. The *segmentation* task required the model to predict aspect labels for a given sentence in a review. The *summarization* task required the model to assign each aspect a representative sentence from a review. The rating prediction task attempted to predict numerical aspect ratings. Since this last task is most closely aligned with the project I'm considering, I'll consider it more closely here.

To measure performance on the ratings prediction task they trained on half of the reviews to predict ratings on the other half. They found that ratings predicted from unsegment text were less accurate, citing the issue of conflicting sentiments appearing for different aspects. Interestingly, model performance did not improve when they introduced segmented text. Performance in this case was on par with simple Support Vector Regression baselines and the authors argued that this occurred because aspect ratings were correlated and "predicting ratings from segmented text fails to account for this correlation."

In summary, the authors made several interesting contributions including a corpora of five million reviews and a model that simultaneously modeled aspect and associated sentiment by learning lexicons for both of these dimensions. Modeling both the aspect and associated sentiment lexicons allowed them to determine which parts of a review correspond to each rated aspect, which sentences best summarized a review and finally to predict aspect ratings from reviews. While I likely won't try an implementation of PALE LAGER, this research provides a foundation for examining how different aspects are represented in the review data and is possible motivation for a topic modeling approach to recover aspect in the reviews from my data set.

"From Amateurs to Connoisseurs: Modeling the Evolution of User Expertise through Online Reviews"

McAuley & Leskovec (2013)

In McAuley & Leskovec (2013), the authors implement a latent factor recommendation system that models user expertise change over time. Using a data set of 15 million user ratings, the model outperforms existing latent factor recommendation systems on a rating prediction task. Following a summarization of model implementation and evaluation the authors use the model's "expertise" latent factors to run additional analysis comparing experts to non-experts along a number of dimensions. For the purpose of this review we'll talk about the modeling, the assumptions made in operationalizing "expertise" and spend the most time on the expert vs non-expert analysis they conducted.

The author's modeled a user's experience level using a series of latent parameters constrained to be monotonically non-decreasing over time. In this formulation a user becomes more experienced (or stays at the same experience level) as they rate additional products. They learned a latent-factor recommendation system for different experience levels, so users were modeled as "progressing" between recommendation systems as they gained experience.

Importantly, the notion of “expertise” in this project is an interpretation of the model’s latent parameters.

In total, the author’s examine four models. The primary differences between the models is in how they model user progression through stages:
model 1 - as a group at uniform intervals
model 2 - as individuals at uniform intervals
model 3 - as a group at learned intervals
model 4 - as individuals at learned intervals.

While the fourth model appears to mirror (my) intuition regarding user-level progression toward expertise, it also introduces the potential to overfit and is flexible enough to subsume the previous three models.

The authors evaluate the four models and a classic latent factor model comparing MSE on two tests – one being the most recent reviews for each user and one being a random sample of reviews from each user. The fourth model outperforms all other models across these data sets (which include data from seven different rating websites).

Given the models expertise assignments (a categorical variable ranging from 1 – 5) the authors then proceed to conduct a qualitative analysis of “expert” and “novice” users. They define “experts” as users who have been assigned a latent factor value of 5, while “novices” have been assigned latent factors 1. Here I enumerate a number of their findings, some of which I believe we can replicate in our study using “number of reviews” as a proxy for expertise:

1) Expert’s ratings are more predictable (lowest MSE) by their model. Interestingly, they found that the user group level most difficult to model were the “near experts” (latent factor values of 4), so predictability was not a linear function of expertise.
2) Experts tend to have cohesive ratings – they agree with each other more than other groups. To assess this the authors examined rating agreement for matching products among users at matching experience level, computing the variance of these ratings.
3) Experts have more extreme views – that is experts give “better” products higher ratings and “worse” products lower ratings. The author’s frame this as type of *acquired taste* where it takes more expertise to appreciate the finer products.
4) There are cohesive *genres* of products (beer styles in RateBeer.com) that experts tend to like or novices tend to like. For example, the authors found that experts tended to like strong ales more than non-experts, while non-experts tended to enjoy lagers more. These findings are apparently consistent with beer drinker intuitions.

This work provides a nice basis for an analysis of expertise in our paper. Interestingly, the author’s did not make use of the actual number of total ratings by users, rather only the cumulative number of ratings during a given time period. As a starting point we should be able to segment users based on the total number of ratings they’ve given. While we can look at the four expert insights they provided here, we can additionally provide a language analysis of experts vs non-expertsin our work.

“No Country for Old Members: User Lifecycle and Linguistic change in Online Communities”

Danescu-Niculescu-Mizil, et al. (2013)

Danescu-Niculescu-Mizil, et al. (2013) propose a framework for studying linguistic change using RateBeer and BeerAdvocate review corpora. They find that users show a measurable lifecycle with respect to linguistic change. In particular, new users tend to display higher degrees of receptiveness to linguistic norms of the community up to around a third of the their eventual lifespan in the community at which point the user reaches “maximum synchrony with the language of the community.” Following this period, users tend to become increasingly unreceptive to changing linguistic norms until the moment they abandon the site. The authors show that the basic form of this trend is robust across user lifespans (e.g. it holds for users who use the site for shorter and longer periods).

Building from these observations the authors show that a user’s patterns of linguistic change can be used to make predictions about that user’s future lifetime in a community. Engineering features to capture the rate at which a member conforms to a community, the authors employ ML models to predict a user’s total lifetime within a community using language from their first few reviews.

To measure the relationship between a given user’s language and that of the community, the authors employ a series of “snapshot language models” – essentially language models for each month in the life of the community. Formally, these are bigram models with Katz back-off smoothing estimated from a held-out set of posts from each month. For a given post *p* the authors can then quantify how surprising *p’s* language is with respect to the language of the community during the same month by calculating *p*’s cross-entropy according the snapshot language model (SLM) for that month: $H\big(p, SLM_{m(p)}\big) = -\frac{1}{N}\sum_i \log (P_{SLM_{m(p)}}(b_i))$ where *p* is the current post, *SLM* is the snapshot model, and *b* is the bigram under *SLM*. Higher cross-entropy values indicate posts that deviate the most from the linguistic state of the community at that particular point in time.

The authors examine two types of linguistic change: 1) at the user level and 2) at the community level. At the user level the authors found that use of singular first-person pronouns *decreases* as the user contributes more reviews. Alternatively, a users use of beer specific vocabulary *increases* with experience in the community. At the community level, the authors note two interesting conventions – the gradual adoption of “smell” over “aroma” and the introduction of more “fruity” words over the course of the community life-cycle. Interestingly, the authors also find that community language entropy across months declines as the community matures – so the community as a whole becomes more “cohesive” or predictable across the nearly 7-year time-frame that the authors consider.

Moving from analyzing user and community linguistic change in isolation, the authors attempt to measure a user’s reaction to linguistic change at a given stage in their community-life. They find that a user’s average cross-entropy across their life-span follows a U-shape – when they first join their language is far from that of the community (high cross-entropy), then they gradually adopt language closer to community convention (decreasing cross-entropy), then once again distancing from the community until eventually leaving. Interestingly, users tend to move away from community norms towards the end of their life-cycle. There are two possible explanations for this – either the user is moving away from the community and starting to use language that is foreign to the current state of the community or the user has stopped adapting to the community norms.

To examine these hypotheses the author's examine the similarity between a given user's post and their preceding 10 posts using Jaccard similarity. They find that on average users' language stabilizes in the first third of their lifespan and then rigidifies, a finding that largely supports the second hypothesis. Among some other interesting findings from this paper, the authors find that users tend use language learned earlier in their life-cycle, employing a metric they call *linguistic progressiveness* which finds the month with lowest cross-entropy in +/- 12 month window. They find that users employ increasingly progressive language through their linguistic adolescence and then use language that is more past-leaning and tie this finding to the sociolinguistic adult language stability assumption.

Finally, using insights from their initial analysis the authors attempt to predict for each user whether they will have 'departed' or still belong to the community given each user's first *w* posts (for some small *w* like *w = 20*). A user is in the "departed" class if she abandoned the community before writing *m* more posts for a small *m* (eg. m = 30). They use the following features:

- **Cross-entropy** for the post according to a snapshot language model of that month.
- **Jaccard self-similarity** of the current post with ten immediately preceding posts.
- **Adoption of lexical innovations** an indicator function which is true if the current post contains a lexical innovation introduced in the community in the previous three months.
- **Number of words**: the authors found that long-term contributors generally wrote longer reviews.

As a baseline the authors fit a logistic regression model with only two features, "frequency" (the average time between posts) and "month" the month of the last review. They find significant model improvement incorporating the five linguistic features (plus the baseline "activity" features) on both precision and recall. Best model performance (F1 = 56.0) occurs with a full model, *w = 20* and the departed range 20-50.

"Early Detection of Potential Experts in Question Answering Communities"
Pal et al. (2011)

In Pal et al. (2011) the authors build expert identification models using machine learning on a question answering community data set. Question answering communities provide information to users who participate in social interactions with another. Importantly, these communities are largely sustained by the involvement of "experts" – a small set (roughly 0.01% of the population) of community members that provide a substantial portion of best answers. The current community of interest is the TurboTax Live Community (TLLC), a community that allows users to ask and answer task related questions.

Given the importance of expert contributions in communities like TLLC companies invest in methods for identifying and cultivating these members, often giving them special status once they've been identified. At the time of this project Intuit (the company that runs TLLC) had a team of employees, which manually identified experts, a high precision, low recall and overall time consuming enterprise. The primary objective of this work was to use machine learning to automatically identify high-potential users in the first few weeks of participation.

The authors identified two primary dimensions for feature engineering, which the called "motivation" and "ability." Motivation captures the idea that an expert should be highly motivated to help others and was operationalized through the features:

- Quantity of contributions
- Frequency of contributions
- Commitment towards the community (# of logins and time on site)

Ability captures the idea that the expert should have the ability to answer questions correctly and was operationalized through the features:

- Domain knowledge - # of best answers
- Trustworthiness - # of votes on answers, # of positively voted answers, ratio of answers with negative votes to positively voted answers.
- Politeness and clarity – language analysis with the following features:
    - Spelling mistakes
    - Bad words
    - SMS language
    - Usage of singular pronouns
    - Usage of negative terms
    - Usage of greetings
    - Usage of special characters

Using the set of features outlined above the authors compared a SVM classifier and C4.5 a Decision Tree classifier using 10-fold cross-validation. Overall the authors found a fairly large discrepancy in terms of precision and recall between these two methods with C4.5 performing better on Recall and getting a better F1 score overall (0.37 for SVM and 0.5 for C4.5).

Having identified a set of users from their classifiers the authors asked Intuit employees experienced with the task of expert identification evaluate the machine classified users. Overall about half the users identified by the learning algorithm were confirmed to have the potential to become a successful super-user and over three quarters were believed to have potential to become a super-user.

This work is relevant to our current study in that the authors used language-level features to identify experts. In our current work we can investigate the inverse problem – what are the salient language features that are unique to experts vs. non-experts.

Summary

In summary, I've presented five different papers, which involve two primary directions I could take this project, the first investigating ML applications of multi-aspect review classification and potentially topic modeling. The second involving the effects of experience and expertise on language use. I believe it will be most productive to start out concentrating on one of these directions, however one certainly doesn’t preclude the other. In both cases I’m interested in interpretable models – I don’t simply want to aim for the best classifier for sentiment/aspect prediction or expertise detection. Instead, I’d like to apply ML methods in a way that allows us to draw some conclusions about the relationship between language use and the learning task at hand, whether that is examining the task of predicting aspect or analyzing the language behaviors of experts and non-experts.