Representation of Complex Features for Rating Prediction on the Yelp Dataset

Nikhil Bose
University of California, San Diego
ndbose@ucsd.edu

ABSTRACT

Within the context, of the recent proliferation of Machine Learning-based intelligent systems, recommender systems using data mining techniques have shown the most initial success and traction (cite example)

Companies have started to take note and are tracking more and more data points to gain a richer understanding of user behavior and preferences. Such insights can be used for numerous applications such as content discovery to increase user engagement on the product.

The goal of this paper is to understand the appropriate encodes of complex web reviews for services such as yelp in order to best predict rating

KEYWORDS

Recommender Systems, Web Mining, Yelp Dataset Challenge, Latent Factor Models, Collaborative Filtering, Sentiment Analysis

1 Dataset

The Yelp Dataset is an education-focused all-purpose dataset with a myriad of rich features including user social networks, business attributes and images from different user reviews. For the scope of this project, the dataset of from the *12th round of the Yelp Dataset Challenge* will be chosen, focusing primarily on the businesses and review datasets. The overall dataset contained over 180,000 unique businesses and around 2.1 million reviews.

The businesses included the dataset are heavily geographically distribution across Canada, the United States and European countries such as the Czech Republic. In order to constrain the dataset, businesses were sorted into the respective states they were located in. The state of Arizona had the most businesses with 28203 unique businesses and 154,836 reviews pertaining to those reviews. Further assumptions can be made as to users within a certain geographical boundary, would rate restaurants and items more similarly than across cultural boundaries if comparing the European reviews. Furthermore, in order to prototype our solution and verify our solutions quickly, the dataset was split into 50,000 data points for each of the training, validation and test sets.

1.1 Exploratory Analysis

The review and business datasets were analyzed to understand some of the different dynamics of the features and how they differed amongst the dataset.

First, we modeled the star ratings based different units of analysis: the business, the individual user, and the entire dataset of reviews

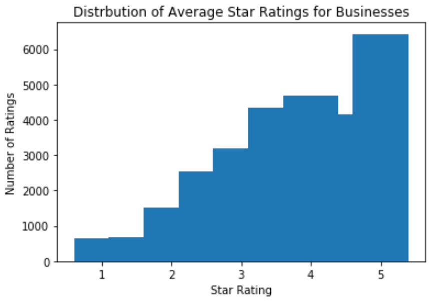

Figure 1a: The distribution of average rating of businesses across the dataset

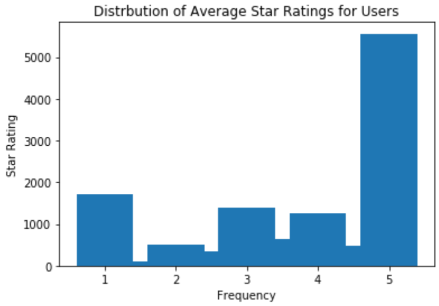

Figure 1b: The distribution of average rating of users across the dataset

*Article Title Footnote needs to be captured as Title Note

†Author Footnote to be captured as Author Note

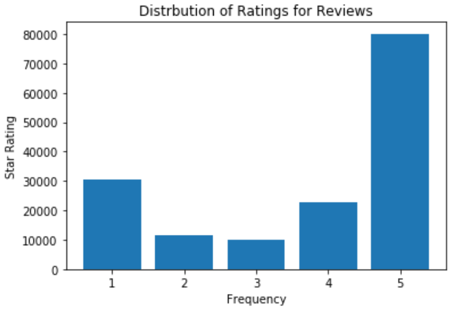

Figure 1b: The distribution of review ratings across the dataset

Through such simple analysis, basic statistics were received over the nature – this could later be used for trivial models of the data. The mean of the review ratings was 3.71, with the median being 4.0 and Standard deviation being. 1.06.

As evidenced by the graphs above, most Yelp reviews are mapped to the extremities of the rating distribution, confirming the intuition that users usually only rate things if they are really good, or really bad. This sort of dataset leads itself to sentiment classification of the reviews, meaning that given a specific review, can it be labeled as positive or negative, but would be trivial for the nature of this assignment.

Next, geospatial features were analyzed. As mentioned before, the dataset was filtered to only consider businesses and reviews within the state of Arizona. The businesses were heavily collocated within specific counties in Arizona, such as Maricopa, which is where the largest city, Phoenix is located.

Given data was heavily grouped in similar geolocation, it gave way to the use of K-Means Clustering to provide a simpler encoding for geolocation data. I iterated through the coordinates of the business dataset, creating a K-Means encoding for each coordinate.

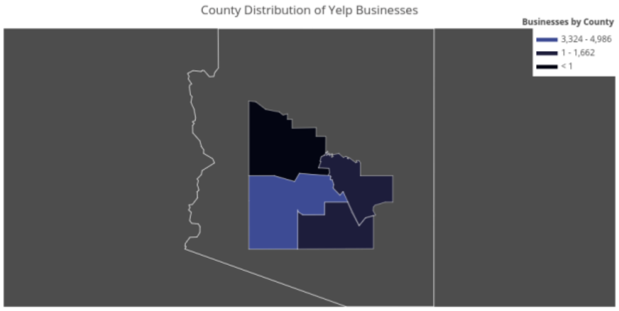

Figure 2a: Choropleth Map of the counties that are covered within the business dataset.

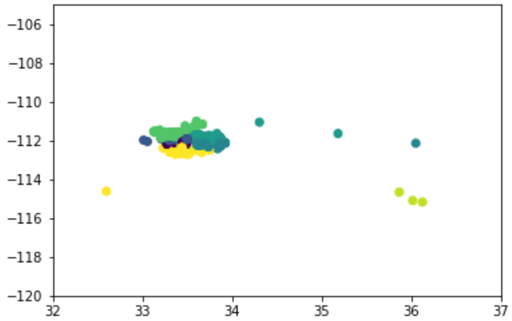

Figure 2b: K-Means clustering of Maricopa county in the coordinates given

Next some of the text-based data points were examined. Reviews went through a data cleaning pipeline before they were properly counted and analyzed. This pipeline was later used to fuel some of the later feature encodings with Latent-Dirchlet Allocations and Bag-of-Word models. The data was first tokenized using a Regular Expression based tokenizer. Later punctuation and stop words were removed. Words were also lemmatized to preserve greater meanings of the word and remove double counting.

As evidenced by the word cloud below, there were some words that are prevalent in strictly positive contexts (eg. *great)*, therefore it would be import to include features about the sentiment of our text in our feature encodings.

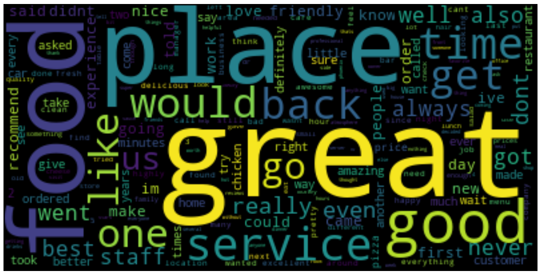

Figure 3: WordCloud of Reviews in Arizona

2 Predictive Task

In this assignment, we want to solve the common predictive task of rating review. More concretely, we want to predict the rating *r*, given a specific user *u* and a specific business *b*, or a *user-item* pair in the terms of collaborative filtering, a popular technique such predictions.

Given the popularity of this dataset, and the frequency of this task, there is a lot of literature to help shape the approach of this paper. Since the dataset has a combination of different type of complex features, an ensemble like model can be used to combine text, temporal, and binary features.

2.1 Evaluation

2.1.1 Evaluating Model - In order to evaluate the efficacy of the different models, a standard measure must be established. A standard established for similar regression tasks is RMSE or root-means-squared-error, as used by the famed Netflix Prized. The derivation is as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{u,i\,\in test}\big(R(u,i) - r_{u,i}\big)^2}$$

Where *R(u,i,t)* is the predicted rating, and $r_{u,i}$ is the actual rating for the given *(u,i)* pair. Note, that each *(u,i,t)* contains a timestamp and text of the review containing the date of the review.

1.1.1 Ensuring Validity - In order to provide generalizable results and prevent overfitting, we will maintain a test, validation and training split of 50,000 data points each. Each model will be fitted to the train set, and then tuned to the validation set using specific hyperparameters to regularize the model against overfitting. Whichever model produces the best result on the validation set will be later used for the test set.

2.2 Baselines

In order to measure the relative effectiveness of the models in question, some baselines were established.

2.2.1 Trivial Baseline. The most trivial baseline predictor would be of the form:

$$R(u,i) = \alpha$$

Where we predicted the global offset term, or the average, for every single prediction. As expected the performance was not impressive as the RMSE reported was 1.606 or roughly analogous to the variance of the review dataset.

2.2.2 –Competitive Baseline. In order to provide a more competitive baseline, we can adopt the form of the below taking into account the implicit user biases. This provides a significant increase in performance having RMSE around 1.201. Interesting to note that the best RMSE was given by finding the user bias / item biases by calculating the average rating for the item, and average rating for the user and subtracting from alpha to get biases, not the iterative ALS approach as proposed in class which had a RMSE of 1.30

$$R(u,i) = \alpha + \beta_i + \beta_u$$

For a model to be considered, successful we would like to see considerable improvements over baseline performance.

3 Models

3.1 Latent Factor Models

3.1.1 ***Matrix Factorization with Biases*** – Matrix factorization is very much an industry standard-approach with regards to approach recommender systems problems. The assumption of Matrix factorization techniques is that there a *k* amount of latent factors that help model the relationship between the users and items.

Therefore, we want to create a matrix of user, item interactions that capture the users past rating history in hoping of understanding the latent dimensions. The resultant matrix would be of dimensions *U* x *I* where *U* is the number of users, and *I* is the number of businesses. In the train dataset the dimensions were (36075 x 14621).

$$R \simeq PQ^t$$

Where R = *U* x *I, and P* = *U* x *k* a lower dimensional representation of user preferences, and *Q* = *k* x *I* a lower dimensional representation of the attributes of the business.

By adding biases found in the baseline, we can create a more complex model:

$$f(u,i) = \alpha + \beta_u + \beta_i + \gamma_u \cdot \gamma_i$$

Where the predicted rating is a function of the global offset, the user/item biases and the individual lower-dimensional representation of the user preferences and item attributes found using the matrix factorization just mentioned.

The model can be optimized using SGD, to find the parameters suited to the dataset. The resultant MSE was a significant improvement of the baseline model leading to an RMSE of 1.145.

3.2 Sentiment Analysis

Thus far, the models described have based their predictions using collaborative filtering – solely basing it of the combination of (user, item) pairs and their respective labels. There is an additional opportunity to improve models given the rich dataset of text features given using supervised learning approaches.

The key assumptions with these models are that they assume that ratings are solely dependent on the sentiment of the text within the review. Based on the composition of words and their attached sentiment, we can predict what the rating of the review would be.

3.2.1 ***Bag-of-Words Model*** – Bag of Words is the most brute-force solution to dealing with text data. In order to encode the text data, a simple word count was established for every document (review text) in the corpus (set of reviews). A fixed dimension encoding was used to limit dimensionality and utilize only the most frequently occurring words. The fixed-length encoding was cross-validated to fine the appropriate trade-off between accuracy and performance. It was found that increasing the dimension of the BoW modeled lead to more performance, until ~6000 words where the increase in vocabulary size provided only marginal increases in accuracy.

The BoW model was paired with different regressors in order to directly predict sentiment of the rating. Ridge Regression, LinearSVR and SGDRegressor, were some alternatives used within the Sklearn package, to most accurately model sentiment. Rather surprisingly, Ridge regression performed the best with the hyperparameter λ = 210, leading to a RMSE of 1.027. This goes in line with the intuition that text reviews were the most influential feature in review rating prediction.

3.2.2 ***TF-IDF Model*** – In order capture the importance of words that are infrequent but could carry a lot of importance. For example the word "Horrendous" could be a low proportion of reviews, but could be very descriptive in negative sentiment. I used the TD-IDF package in sklearn to quickly procure a TF-IDF representation of the documents in the corpus. This representation produced an RMSE of 1.114, performing worse than the BoW model, while having an immensely higher dimension.

3.3 Business-Feature Based Models

In reality, users do not predict based their reviews in isolation, usually there are some correlations between the way that the user would predict and the way most of the users have rated the same business in the past. There is a possibility that the rating could be a function of the attributes of the business itself.

In order to test that hypotheses some simple models were established using solely the attributes of the business as features.

3.3.1 ***Solely Business Features*** – In order to capture the simple attributes of a business, I created a simple encoding:

[attr,price,cat,hours]

Where `cluster` is an encoding of the geolocation, using K-means Clustering, `Price` is the price category if the business was a restaurant (the top category). `Cat` – which specific type of business and `hours` the number of hours it was open per week. By including these specific variables, we want to analyze the relationship between them and the rating. I combined this encoding with Ridge Regression similarly used in sentiment analysis, leading to an RMSE of 1.208, which was rather unsatisfying. `attr` was the a binary vector of the top 4 attributes a business could have (ie. `acceptsCreditCards) -` The hypothesis here was that there are these specific attributes that correlate positively with high rating.

3.3.2 ***Sentiment and Business Features*** – In order to characterize the sentiment of the text, and understand the individual attributes of the business, a sentiment anaylsis component was added.

[sent_score,attr,price,cat,hours]

Instead of training a separately, I opted to use NLTK's pretrained Vader Module to give quick sentiment analysis scores. `sent_score` is the individual sentiment of the document. If it was positive, the sentiment score would be positive, otherwise the score would be negative. This created a significant jump in performance, further validating the importance of sentiment in our predictoions. Resultant RMSE was 1.067, curiously still performing worse than the simple BoW model.

Curios, I decided to opt for 4000 word BoW model to encode, sentiment while still retaining business feature - his lead to another breakthrough, as the sentiment analyzer was specifically trained for this dictionary of words. The resultant encoding was:

[BoW,attr,price,cat,hours]

The resultant RMSE was 1.030.

3.4 Complex Models

Following, the success of the sentiment-based models, and the promise of the value of individuals feature I decided to adopt a complex model using an aggregation of features from the review and the business.

3.4.1 ***Latent Dirichlet Allocation***– In order capture the importance of specific topics, within a corpus, I leveraged Latent Dirichlet Allocation topic model, analogous to the matrix factorization approaches based earlier, in terms of leveraging latent factors. In order for efficient modeling and representation, I leveraged the Gensim package that has library functions for both word counts and LDA modeling. By doing so, I found the top 40 topics within the corpus, and assigned each sample in my train set a topic according to its LDA distribution

3.4.2 ***Incorporating Clustering and Temporal Features*** – Using Clustering and adding temporal features I was able to best capture the full complexity within the reviews and business datasets. For each sample I encoded the geolocation of the business, using the K-Means clustering described earlier. I cross-validated 12 clusters as the appropriate balance for accuracy. Furthermore, I added the date of the review in hours from 10/1/2004 – the date of Yelp's inception. Based on Koren, Bell's and Volinsky's groundbreaking paper on the Netflix prize – there are some significant correlations between ratings and temporal features [1]. The final encoding was:

[BoW,Attr,Price,Cat,Hours,Date,LDA,Cluster]

4 Literature and Related Work

I was heavily influenced by Koren's, Bell, Volinsky's work on the Netflix Prize, specifically on the benefits of matrix factorization and the benefits of integrating temporal features. Matrix Factorization is especially suitable for when there are no user /item features, or there is the possibility of integrating implicit feedback, unfortunately in this instance the latent factorm model performed more poorly than feature-based models [1].

Since the Dataset has been publicly available for over 12 years now, there is an extensive amount of industry publications and research on the topics[3]. Many of the topics, are focused on applying new and emerging subsets of machine learning, such as deep learning, and more complicated feature encodings. One of the most informative papers, were by Prof. McAuley himself, on the importance of modeling latent topics, and based on the advancements he proposed, I integrated LDA into my model. I tried implementing LFM that uses text input, but had difficulty in implementation [2].

Alternate models proposed by Gong et. Al have shifted towards a clustered form of sentiment analysis and performing classification on those clustering's, providing a more accurate form of sentiment analysis.[5]

Some of the more advanced approaches include passing the reviews through various layers of Recurrent Neural Networks as shown by Ming et Al, to provide more accurate NLU of the review text to provide more accurate recommendations. [4]

Lastly, many of the models used such as Linear Regression, TF-IDF and BoW were used as taught in class. The dataset in particular had many similarities to the Amazon purchase reviews, especially in terms of the review text, so many approaches could have been carried over.

5 Results and Conclusion

By implementing the complex model mentioned in section 3.4.2 I was able to achieve my best performance of 0.919 RMSE on the test set. Below is the how the model compared to the baselines and comparable models.

Model	RMSE
Competitive Baseline	1.30
Matrix Factorization	1.145
Bag-of-Words and Ridge Regression	1.027
TD-IDF and Ridge Regression	1.114
Business Features and Sentiment	1.030
Final Complex Model	0.919

As displayed above, the final model provided the best performance in terms of RMSE, while all of the models here provided a significant jump over the baseline solution.

From the results it was clear that user sentiment, as inferred through the text was the most important feature throughout the entire dataset. It was interesting to note that LDA was not very effective on its own, it only increased performance when paired with BoW, which is counterintuitive, since there should be some redundancy in the lower-dimensional encoding. Perhaps, the result of this could be that, the LDA topics provided the high-level topics encoded in the text, (ie. Like the direction of a vector), where the BoW model, could better encode the magnitude of the vector, so therefore there could be some use cases of having both. I was generally impressed by the performance of the BoW model paired with ridge regression. In some ways it is a naieve approach, but similar to what Prof. McAuley said in lecture, it provides a "good-enough" representation of the text to characterize things like sentiment, especially with a large fixed-length encoding. It would have been interesting, If I had the chance to try out bigrams, or perform colocation analysis to characterize salient phrases and sentiment.

Furthermore, I was a little bit disappointed with the performance of the latent factor models. They provided good initial performance, but had difficulty generalizing to unseen data or predicting pairs that were generally uncommon. I had attempted approaches using a combination of latent factor models and user/item features, but had difficulty implementing/solving gradient descent for such matrix factorization[2].

In the future, it would be better to have had a model that accounted the interactions between all of these complex features, besides introducing linear models [1].

ACM Reference format:

FirstName Surname, FirstName Surname and FirstName Surname. 2018. Insert Your Title Here: Insert Subtitle Here. In *Proceedings of ACM Woodstock conference (WOODSTOCK'18). ACM, New York, NY, USA, 2 pages.* https://doi.org/10.1145/1234567890

REFERENCES

[1] Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, (8), 30-37.

[2] McAuley, Julian, and Jure Leskovec. "Hidden factors and hidden topics: understanding rating dimensions with review text." In Proceedings of the 7th ACM conference on Recommender systems, pp. 165-172. ACM, 2013.

[3] https://www.yelp.com/dataset.

[4] Ming, Yao, Shaozu Cao, Ruixiang Zhang, Zhen Li, Yuanzhe Chen, Yangqiu Song, and Huamin Qu. "Understanding hidden memories of recurrent neural networks." arXiv preprint arXiv:1710.10777 (2017).

[5] Zhang, Yongfeng. "Incorporating phrase-level sentiment analysis on textual reviews for personalized recommendation." Proceedings of the eighth ACM international conference on web search and data mining. ACM, 2015.