Ratings Prediction Using Review Text To Address Cold Start Conditions

Rahul Bhalerao [*]
rbhalera@eng.ucsd.edu

Mrinmayee Hingolikar [†]
mrinmayeeh@eng.ucsd.edu

Sanjeev Rao [‡]
sjrao@ucsd.edu

ABSTRACT

Collaborative filtering methods like latent factor models have been used to predict ratings for an item by a user. However, the performance of latent factor models depends on a dense training dataset in which each user has made several reviews and each item has also received several reviews. In most practical applications, when a new item or a new user enters the system, it is hard to find this sort of a dense dataset for these new users and items. We have used the Yelp academic dataset and limited our study to the restaurant reviews in that dataset. This paper presents a model to take advantage of signals from review text to predict the rating that a restaurant may receive from a user. Our model uses LDA to learn topics from restaurant reviews and uses the distribution of topics in the reviews to extract features for our regression model. Our model relaxes the strong requirement of a dense dataset placed by collaborative filtering methods and instead needs a lesser restrictive requirement of at least one review received by an item and one review posted by a user to make predictions. We compare the performance of our model with other models based on collaborative filtering methods under three different data set conditions - sparse data, medium dense data and dense data. We find that our model significantly outperforms latent factor methods under sparse conditions while the gap in performance gets closer as the density increases. We show trends to explain the conditions under which collaborative filtering models do better than the model presented in this paper and vice versa. We also show some interesting topics learned by our simple model and explain how this helps to make good predictions under sparse data conditions.

[*]Rahul insisted his name be first and also he comes first when ordered by last name :-)

[†]We disavow any knowledge of this author's actions.

[‡]This author is the one who did all the really hard work. Just kidding. Please do not take the title notes seriously

WOODSTOCK '97 El Paso, Texas USA

DOI: 10.475/123_4

Keywords

Collaborative Filtering, Latent Factor Models, Topic Modeling, Latent Dirichlet Allocation, Text Mining, Bag of Words, Linear Regression

1. INTRODUCTION

This paper discusses methods, techniques and results that have been developed on the Yelp data set. Latent factor models that work on the foundation of matrix completion techniques have been used to predict ratings for user item pairs given a large training dataset of users and their ratings on several items. These methods however perform very poorly when the training dataset is very sparse. We say that an item is sparse when it has received very few reviews. We say that a user is sparse when the user has posted very few reviews. We define sparsity in our dataset more accurately in section 3. In this paper, we present a model that significantly outperforms the latent factor model under sparse data conditions. Our model is a linear regression model that takes advantage of the signals from review text to build features for regression. It builds these features by identifying the low dimensional representation of the user's preferences from the reviews posted by the users. It identifies the restaurants low dimensional features from the reviews that the restaurant receives. Both the restaurant's features and user's features are then combined to generate the features for the linear regression model. The rest of the paper is organised as follows. Section 2 discusses work that has inspired our ideas, Section 3 explains how we have constructed our datasets and explains the notion of sparsity in our dataset which is critical to the claims made in this paper. Section 5 explains the interesting topics that were learnt by performing LDA on our dataset. Section 4 defines our prediction task. Section 6 discusses our prediction models along with the baseline models against which we have compared performance. Section 7 compares the performance of our models against the baseline models. Section 8 summarizes the main conclusions and takeaways of this paper.

2. RELATED WORK

We take inspiration from [3], where the authors argue that most existing recommender systems consider only the ratings and do not effectively use the review's rich text data. It describes a model - Hidden Factors as Topics(HFT) Model, which attempts to combine the ideas from Latent-Factor Recommender Systems and Latent Dirichlet Allocation.
In HFT, the γ_i (K dimensional latent features for item i)

and θ_i (K dimensional topic distribution for item i) are not learned independently, but are linked with the hope that if a product exhibits a certain property (high $\gamma_{i,k}$) this will correspond to a particular topic being discussed (high $\theta_{i,k}$). However, learning them together is non trivial, since one of these is stochastic while the other is not, and so the authors define a complex transformation these two parameters. We differ from this paper in that we do not learn the two parameters together, and propose a simple linear regression model which considers the topic distribution as a feature. We believe that for a sparse dataset even a simple model which takes into consideration the LDA topic distribution can outperform the collaborative filtering approach.

[2] is one of the Yelp's Dataset Challenge winners, which describes a method using Latent Dirichlet Allocation(LDA) to extract hidden subtopics from review text. The goal of the paper is to help improve the restaurants by finding out what the users are most concerned about from the user reviews. They use this information to suggest improvements to the restaurants. We also use LDA to discover hidden topics, however we use the user-item interaction as a feature for each of the discovered topic to predict restaurant rating.

3. DATASET

The Yelp academic dataset[1] consists of a total of 958,777 reviews. Since we wanted to study the prediction of ratings for different sparsity/density of the data, we constructed different sets of data based on different sparsity/density. For this we divided the businesses into different bins based on the number of reviews of that business. For creating the sparse data set, we considered all the data reviews of all businesses which had upto 12 reviews each. Similarly, for the dense data set we took the reviews of the businesses which had more than around 1000 reviews. In other words, the sparse data set contained the reviews of the businesses which had the least reviews while the dense data set contained the reviews of the businesses which had the most reviews. From each of this sparse and dense data sets, we iterated over each of the review and first 8 reviews were put into the train set, every 9th was put in the validation set and every 10th review was put into the test set.

In this way, the sparse train, validation and test set consisted of 44320, 5540 and 5540 reviews respectively. However, on a study of the distribution of the data, we found that approximately 45% of the users in the validation and test set were not present in the train set while around 2% of the businesses in test and validation set were not present in the train set. This meant that there were some reviews in the validation and test set whose business or user was not seen earlier. This can be handled by reporting a global average in case of unseen business or user, but since we wanted to study collaborative filtering, we wanted each of the test and validation review to be such that its user and business had at least one review in the train set. This filtered data set formed our medium dense data set. However, putting such a condition reduced the number of reviews below 50k, and so we considered businesses which had upto 21 reviews. That is why this data set is named as medium dense data set, as its density is higher than the spare data set. This way the medium dense data train, validation and test data set consisted of 67392, 6157, and 6215 reviews respectively.

We curated the dense data set in a similar way, by putting a condition that each review in the validation and test data set must be such that its user and business must have atleast one review in the train data set. We encountered scalability issues while training our models on the dense data set, so to reduce the size of the dense data set, we removed from the train set all the reviews whose business or user did not appear in the reviews of either the validation set or test set. This way our reduced dense train, validation and test data set consisted of 7826, 2173, and 2242 reviews respectively.

4. PREDICTION TASK

We aim to predict the rating that a user u may give to a restaurant i. Our primary goal to accomplish this task is to develop a model that uses signals from review text of all the reviews received by i and review text of all reviews given by u to other restaurants. We use Latent Dirichlet Allocation model to learn topics from review text and a linear regression model to understand user-item interaction for each topic. LDA and regression with features from topic modeling is explained in detail in Section 5 and 6 respectively.

We asses the performance of our model on the datasets of varying densities as described in the previous section. We compare the performance of our model with two baseline models based on collaborative filtering. In collaborative filtering, we use models with and without latent factors of user-item interaction. The intuition is that for sparse dataset, in which reviews for i or reviews from u are sparse, our model will perform better as collaborative filtering is prone to cold-start problems. As the dataset gets denser, we expect collaborative filtering to perform better as there is sufficient data to discover biases and latent factors. Our secondary goal is to affirm this hypothesis.

5. TOPIC MODELING USING LATENT DIRICHLET ALLOCATION

Latent Dirichlet Allocation is a method widely used in natural language processing to generate probabilistic distribution of text across a given set of topics. The intuition behind using this model is that review text, even for a sparse dataset can convey rich information about the user and restaurant. K topics are discovered from the review text which are assumed to be distributed(θ) according to Dirichlet distribution. For each review text $d \epsilon D$, LDA associates a K dimensional topic distribution, describing the fraction of words in d that discusses each topic $k \epsilon K$. Each topic k has a word distribution denoted by ϕ_k that denotes the probability that a particular word is used for that topic. The probabilistic model under LDA for a text corpus τ is:

$$p(\tau|\theta,\phi,z) = \prod_{d}\prod_{j=1}^{N_d} \theta_{d,z_{d,j}}\phi_{z_{d,j},w_{d,j}} \quad (1)$$

where $z_{d,j}$ is the topic assignment for each word in $d \epsilon D$. [3]

In our implementation we first find the vocabulary of frequently used 1000 words excluding stopwords in restaurant reviews across all the reviews in training set. We then train Latent Dirichlet Allocation model to discover 50 latent topics from reviews. LDA performs very well on the review text and discovers fine grained topics that describe various cuisines, service of the restaurant, staff, ambiance etc. Figure 1 shows word cloud of some sample topics of the 50 discovered topics. Each figure shows the top words that are

associated with one topic. Notice that we have learnt information about a restaurant like the cuisine it serves, its kid friendliness, decor, service level, type of food (drinks, coffee shop, lunch) etc. These topics have been automatically discovered from training data through LDA.

6. PREDICTION MODELS

6.1 Regression with Features from Topic Modeling

Let us define some terms for ease of understanding. Let document d_i denote all the reviews received by restaurant i. Let document d_u denote all the reviews made by user u. All the reviews are organised as d_i for every restaurant i and then LDA is performed to extract 50 topics. The intuition behind this approach is that this sort of aggregation will contain information about restaurant i rather than the opinions of users. We want our topics to capture this sort of information. The same is suggested by [3]. The trained LDA model gives us the word to topic allocations, which is then used to compute the distribution of each topic for every restaurant and the distribution of each topic in the reviews of every user.

Once our LDA model is trained, the topic distribution of each restaurant i is obtained by computing the topic distribution of document d_i with the trained LDA model. This is denoted as γ_i. Similarly, the topic distribution of each user u is obtained by computing the topic distribution of each document d_u with the trained LDA model. This is denoted as γ_u.

Θ_k for k ranging from 0 to 52 denote the parameters that we are learning through linear regression. Θ_0 is the offset in the regression model. α_u is the average rating given by user u and α_i is the average rating received by the restaurant i. γ_u and γ_i are both 50 dimensional. We capture the correlation between the users's preferences in each topic and the restaurants inclination towards each topic through the element wise product in our model. We believe this is crucial to the performance of our model.

$$f(u,i) = \Theta_0 + \Theta_1 * \alpha_u + \Theta_2 * \alpha_i + \sum_k \Theta_k * \gamma_{uk} * \gamma_{ik} \quad (2)$$

γ_u and γ_i are both learned through unsupervised learning using LDA while all the Θ_k for k ranging from 0 to 52 are learned through supervised linear regression. We performed linear regression with a squared error loss function with regularization.

6.2 Collaborative Filtering Model Using only Bias Terms

In this model we define a prediction function for restaurant rating, using a collaborative filtering model. We find user and item biases towards rating. These biases are then used with a global bias offset to compute rating predictions. The model for rating is:

$$f(u,i) = \alpha + \beta_u + \beta_i \quad (3)$$

where:
α = offset paramenter
β_u = bias for user u
β_i = bias for item i

(a) Location and Service (b) Breakfast

(c) Drinks (d) Vietnamese Cuisine

(e) Japanese Cuisine (f) Indian Cuisine

(g) Fast Food (h) Sandwiches

(i) Opinion (j) Kid Friendliness

(k) Staff (l) Ambience

Figure 1: Sample Topics

This model is trained using iterative update rules on the training dataset.

6.3 Latent Factor Model

We explore implicit feedback from the reviews and user-item interaction by discovering their latent factors. In addition to item and user biases, this model captures implicit item features and user preferences. For example, latent factors for restaurant can capture the type of cuisine, service offered and latent factors for user can capture preferences of that user towards these attributes of a restaurant. The predictive task in this model is:

$$f(u,i) = \alpha + \beta_u + \beta_i + \gamma_u.\gamma_i \quad (4)$$

where:
$\gamma_u = K$ dimensional latent factors for user u
$\gamma_i = K$ dimensional latent factors for item i
The other features are the same as in the previous collaborative filtering model. This model is updated using iterative update rules for the features.

7. PERFORMANCE COMPARISON

Figure 2 compares the performance of our regression model against our baseline collaborative filtering models. Performance is measured with mean squared error. Each plot looks at the performance of the three models on the training, validation and the test data subset. In particular, we focus our attention on the performance of the three models on the validation and the test set.

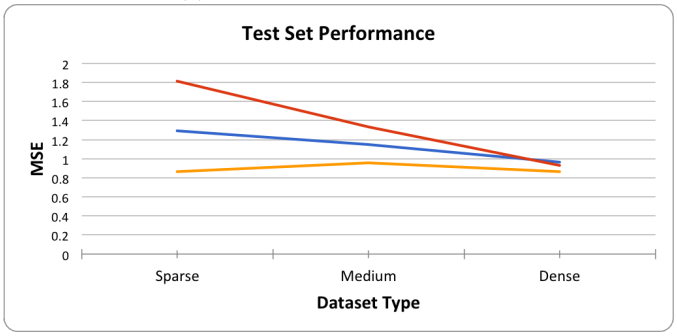
Under sparse conditions, our regression model significantly outperforms the baseline collaborative filtering models. As we move from left to right along the x-axis, the density of the data increases. We notice that the gap in performance between the models is bridged with increasing density in the dataset.

In figure 2b and 2c, we notice that as the density in the data increases, there is a improvement in performance of the latent factor model (with latent factors gamma) while the performance of the regression model has flattened. This trend is expected as the a dense data set allows to learn rich latent factors which enable the latent factor models to outperform the regression model. Due to lack of time, we have not been able to test on datasets that are more dense than the once we have examined. Extrapolating the graph further, we can expect the latent factor model to outperform the regression model. Similar results have been explained with the HFT model described in [3]. Again due to lack of time, we have not been able to compare the performance our our model with the HFT model. Since our model requires such little user item density, it would be interesting to compare our model against the HFT model under sparse data conditions.

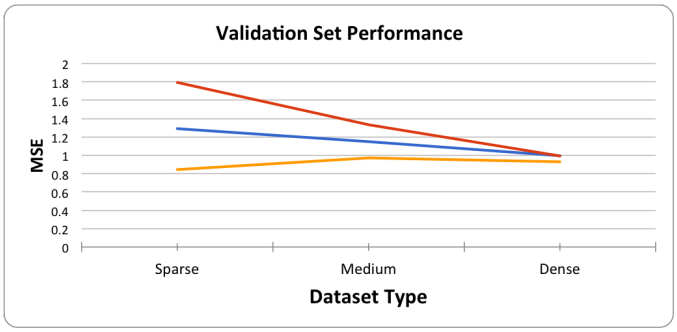
8. CONCLUSIONS AND FUTURE WORK

We have presented a linear regression model that predicts ratings for a given user item pair. We have shown that our model outperforms collaborative filtering methods under sparse data conditions. As expected the performance gap reduces with increasing density in the data. Based on the trends observed, we expect the latent factor model to outperform the regression model when the dataset is sufficiently dense. Our model can help address the problems faced by

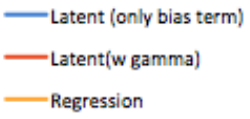
(a) Performance on Training Set



(b) Performance on Test Set



(c) Performance on Validation Set



(d) Legend

Figure 2: Performance (MSE) of the linear regression model and baseline collaborative filtering models on the Sparse, Medium and Dense Datasets

traditional collaborative filtering methods under cold start conditions when a new restaurant or a new user enters the system. Due to lack of time we were not able to compare our model with the HFT model described in [3]. This would make an interesting comparison between the complex HFT model and our simple linear regression model under sparse conditions. We would also like to explore a collaborative filtering model that incorporates the features of our linear regression model as additional features and measure its performance.

9. ACKNOWLEDGMENTS

We would like to thank Professor Julian McAuley for his guidance throughout the quarter.

10. REFERENCES

[1] Yelp Dataset Challenge. http://www.yelp.com/dataset_challenge.

[2] E. J. James Huang, Stephanie Rogers. Improving restaurants by extracting subtopics from yelp reviews.

[3] J. McAuley and J. Leskovec. Hidden factors and hidden topics: Understanding rating dimensions with review text. *RecSys*.