Rating for sub-properties using latent topics

Madhavi Yenugula
A53046539
CSE255 Assignment1
UCSD

1 Dataset

Tripadvisor is a travel website which provides user generated reviews for travel-related content. I used the Tripadvisor dataset available here here. The dataset has a total of 1621956 reviews about 12,773 hotels with an average of 126 reviews for each hotel. Hotel's information provided includes its name, location, price etc. A review for a hotel includes the author's location, date, rating and review text. The rating aspect of Tripadvisor dataset is particularly interesting as it provides ratings for other aspects/sub-properties of the hotel in addition to the overall rating. This information is helpful for a customer to take an informed decision about his choice based on his specific preferences.
From the number of ratings per month, we can see that there are particular months during which people take vacation - the region from Aug-December. It is consistent with the general knowledge that people take vacation during holidays. This trend is observed during all the years from 2003 to 2012 from 2. It can also be seen that the number of people using/rating places on Tripadvisor increased almost exponentially over the years.

Figure 1: Average number of ratings per month

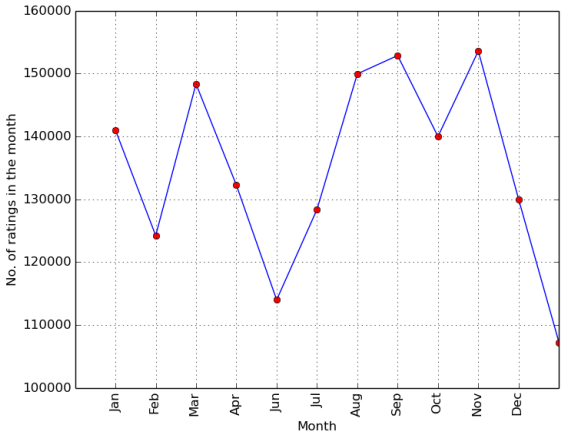
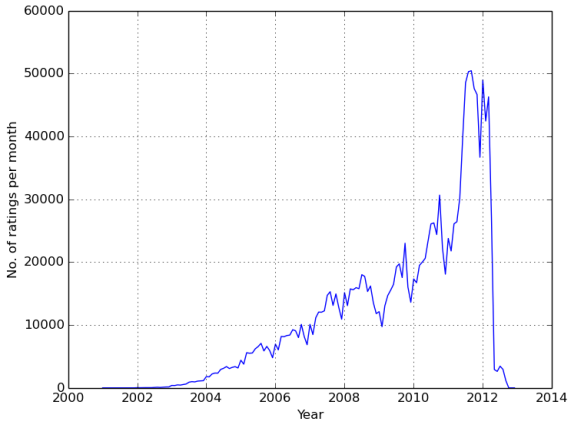


Figure 2: Average number of ratings from 2003 to 2012



One other interesting aspect of this dataset is the location information that is provided for both authors and hotels. It is interesting to note that the proportion of foreign visitors is different for different places as shown in 7. It would be interesting to investigate if being a local or visitor to a place actually has an effect on the overall rating.

The rating information includes a separate rating for Service, Business service, Cleanliness, Business service (e.g., internet access), Check in / front desk, Value, Sleep Quality, Rooms, Location. Though overall rating and rating for each of these factors is heavily correlated, it is not the same always. So, the goal of this project is to predict rating for each of these factors from the given data. Figure 8 shows the distribution of ratings for the given data. To the left of the blue line are the data points whose average rating for a particular sub-category is less than the average overall rating for that hotel by a person. To the right of the line are the points whose sub-category ratings are higher. Figure 9 shows the distribution of sub-category rating and overall rating. The vertical lines are the places where both of them agree. As we see from the figure, both the ratings agree at many points but there are also a significant chunk of points where they don't. The cases where the sub-category ratings

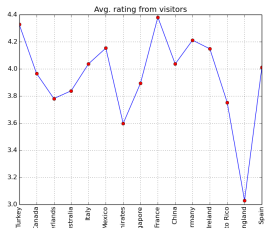


Figure 3: Avg. rating visitors

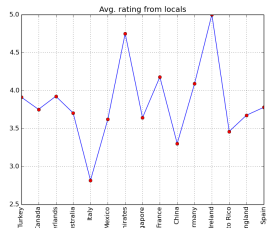


Figure 4: Avg. rating locals

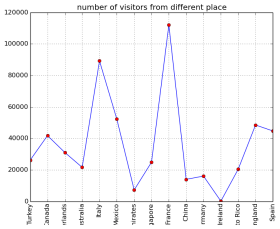


Figure 5: # of visitors

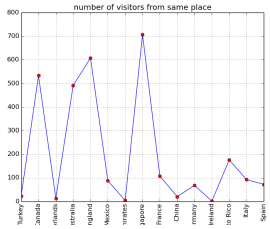


Figure 6: # of locals

Figure 7: Location related graphs

agree with the overall rating are higher when the overall rating is 5 as compared to other cases. This means that we can attribute an overall rating to the sub-category rating where the overall rating is very high but not so easily in the other cases. The last vertical line in Fig.9 signifies this. In both the graphs, different color scatter plots are used to represent each of the 10 sub-categories.

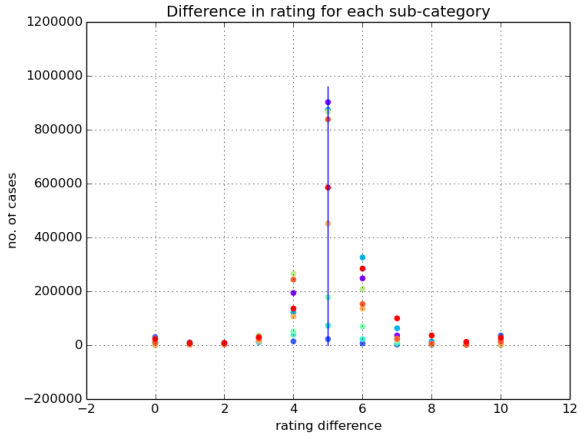


Figure 8: Distribution of Rating

2 Predictive Task

As seen from the data in the previous section, there is a high variance in the overall rating given by a user to a hotel vs. the other sub-category rating the user gives. The dataset is

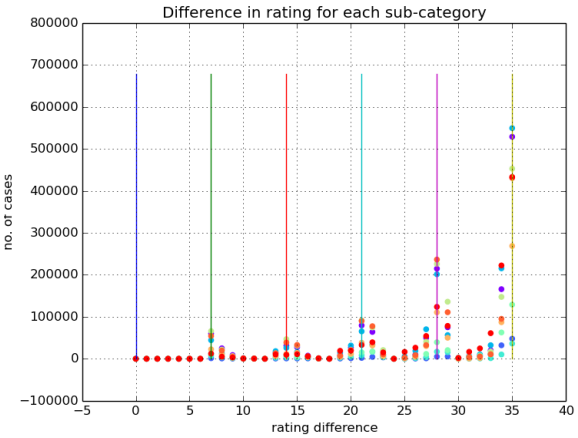


Figure 9: Difference in Rating

a bit noisy, users don't provide the sub-category ratings for all their ratings. The dataset is divided into 3 sets. Validation set, Test set and Training set. The first 2 reviews (which contain the rating information as well) from each of the hotel's reviews are added to the Validation set. The next 3 reviews are added to the Test set and the rest of the reviews are used to train the data.

The reviews in the Test set contain the overall rating and other sub-category ratings. So, I predict the rating for a given user and compare it against the rating originally given by the user. The measure used to decide the performance improvement is the RMSE which is calculated as shown in eq.1.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(predRating_i - actualRating_i)^2}{N}} \quad (1)$$

The baselines I'll be comparing against are the average with item and user bias (as used in [7]). The model I describe tries to enhance the rating prediction by compensating for the factors that the biases don't explain. In addition to using the topic distribution of a review, sentiment associated with a certain review and a certain sub-category attribute is also used to predict the rating. If the sentiment associated with a certain statement does affect the rating, it should be shown in the results as a minor improvement at the least.

2.1 Previous Work

A lot of work has been done using sentiment analysis for predicting the rating of a given review or some textual content [1], [3], [9] and [2] using Machine Learning methods. [4] studies the rating of sub-properties of Tripadvisor data. It is one of the first works to study the rating for a given scenario(aspect). They use a probabilistic rating regression

model to solve the problem. The input to their system is the review text and the overall rating information. Given this information, they predict the rating for a given sub-property. They use LDA to discover the latent topics in the reviews and use a Maximum Likelihood estimation method to infer the specific rating. This works assumes that there is a predefined set of aspects or sub-categories. This work incorporates the sentiment of the review into the measure by attributing a satisfactory rating wth respect to each of the subcategories. [5] is an improvisation of the previous work. This model doesn't assume any pre-specified keywords for the sub-categories. In addition to the topic distribution, they also learn the weights a reviewer associates with each of the aspects. As discussed in section 2.1 of [7], the baseline predictors for this work are similar to the ones discussed in this paper. According to [7], most of the collaborative filtering setting examples exhibit the property of having heavy item and user biases. This is the basis for [6] as well. It is an other work which uses the textual review data to model the latent factors between items and users to predict the rating. This model also uses LDA to model the topical distribution and this information is incorporated into the model. [4] and [5] have extensively discussed the aspect ranking and they have the best performance on this dataset. [10] has a similar rating prediction for BeerAdvocate dataset. The model I implement is a fairly simple model which verifies the claim that sentiment and topic distribution helps in predicting the rating for sub-categories by modeling the variance in the data with the help of latent topics present in the review and the sentiment it expresses. This model cannot compete with the sophisticated models described above as [4] already incorprates the sentiment and latent topic data into its model.

3 Features

Similar to the work in [7], it is assumed that rating given by a reviewer has some implicit bias that is associated with that particular reviewer and the item he is reviewing. For this part, we need the ratings that a user gives to each of the sub-categories. The average, user bias and item bias are calculated separately for each of the sub-categories. Apart from these biases there are factors that push a user towards a particular rating, which might be lesser or higher than the cumulative ranking given by the baseline predictor with biases. These factors can be learnt from the review text that is provided along with the user's rating. In addition to being able to predict the overall rating using the text, these features learnt using the latent factors are also used to predict the rating for each of the sub-categories mentioned in the Exploration section. Apart from the topic features, the sentiment associated with the review is also a very high indicator of the rating as shown in the figure 10. A person's review usually contains some aspect that pertains to the rating that he gives for a particular sub-category. The idea of using latent topics is to capture this relation and there by improve the rating prediction accuracy.

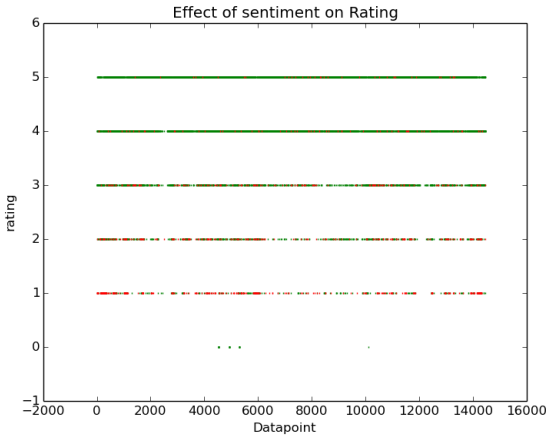


Figure 10: Effect of sentiment on Rating

Stop-words from the reviews are removed before discovering topics on the data. Apart from the common words, words which appear frequently in the data like hotel, room - which don't directly affect any of the sub-categories are also removed.

4 Model

The initial part of this model is based on [7]'s baseline prediction. Most of the ratings of the users can be explained by the biases of the users and items. The baseline rating r_{ui} can be calculated using the eq.2.

$$r_{ui,c} = \mu_c + b_{i,c} + b_{u,c} \tag{2}$$

Here μ is the overall average of the entire training data set. We try to solve the minimizaton function in eq.3.

$$\min_b \sum_{u,i,c} (r_{ui,c} - \mu - b_{u,c} - b_{i,c})^2 + \lambda (\sum_u b_{u,c}^2 + \sum_i b_{i,c}^2) \tag{3}$$

The obtained b_u and b_i values are plugged into eq.2 to obtain the rating for the new user.

But this equation doesn't consider the relation between user u and item i in the calculation of $r_{ui,c}$. From 10 we can see that sentiment in the review text plays a major role in identifying the rating given by a user. For this the standard deviation of the validation set σ_c of the rating $r_{ui}$is calculated. This is the value that needs to be corrected.

4.1 LDA

latent Dirichlet allocation (LDA), [8] a generative probabilistic model for collections of discrete data such as text

Topic	Words				
2	service	staff	great	time	restaurant
10	staff	wonderful	staff	service	stayed
28	service	view	beautiful	concierge	food

Table 1: Topics chosen to represent the sub-category 'Service'

corpora. LDA is a three-level hierarchical Bayesian model, in which each item of a collection is modeled as a finite mixture over an underlying set of topics. Each topic is, in turn, modeled as an infinite mixture over an underlying set of topic probabilities.
Each document in the LDA setup is the concatenation of all the review of a hotel of the Training set. All the stop words are removed and few very commonly used words in the hotels setup are also added to remove the bias in the topic distribution. LDA is run for two K values - K = 50 and K = 10. Results from both the setups are reported. LDA assigns a topic distribution for each of the documents (hotels) and also the composition of each of the topics. After this step, for each sub-category, few keywords are chosen. Presence or absence of these keywords decide whether that particular topic is related to the sub-category. For example, the topics with words service, staff are chosen to represent the sub-category service as shown in Table 1

For each of the sub-categories, few topics are selected based on few keywords. For each review in the test data set, a topic distribution vector is associated with it based on the model learnt on the training set. To quantify the presence of an aspect in a review, we find the magnitude of modified topic vector. The topic vector for a particular review for a particular sub-category is nothing but the original topic vector with the contributions of un-related topics zeroed out. If the original topic vector is $(x_1, x_2, ..., x_{50})$. For sub-category Service, the topics are 2, 10, 28. The modified topic vector is $(0, x_2, 0, 0, ..., 0, x_{10}, 0, ..., 0, x_{28}, ...)$ The magnitude for each of this topic vector is normalized with the square root of length of the vector. For calculating the overall rating, the topic vector is not modified and the value is calculated in the same way as described above. So, for each of the subcategory the LDA score $l_{ui,c}$ is calculated for each of the reviews for which ratings are to be calculated.

4.2 Sentiment Analysis

The sentiment expressed in the review plays a major role in explaining the deviation of the user's rating from the baseline value. The same keywords that are used to mine the topics for each sub-category are checked in each of the sentences. If a sentence contains any of those words, it is assumed to have some information about that sub-category.

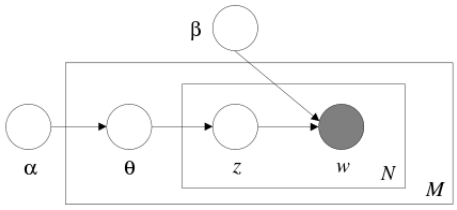


Figure 11: slab representation of the standard LDA

So the sentiment of all such sentences in the user's review is calculated and the average of those values is assigned the sentiment value $s_{ui,c}$ for the user's review for that sub-category c.

$$s_{ui,c} = \frac{\sum_{sentences_c} \text{sentiment value}}{\text{no. of sentences}} \quad (4)$$

sentence	sub-category	rating	sentiment
.. no internet service	internet access	1	-ve
Had a quite nice night sleep	sleep quality	4	+ve
.. a basic place to sleep peacefully	sleep quality	5	+ve

Table 2: Sentiment about sub-categories in review text

As shown in Table 2, many reviews have some text supporting the good/bad rating a certain aspect receives.

4.3 Update

The equation 2 to calculate the rating is modified to the following equation

$$r_{ui,c} = \mu + b_{i,c} + b_{u,c} + \sigma_c * s_{ui} * l_{ui,c} * \beta \quad (5)$$

The parameter β is determined heuristically on the validation set. On experimentation it was understood that it is easier to detect negative sentiment statements and the accuracy is higher in their detection. So, the β parameter for statements with negative sentiment is higher than the ones with negative sentiment. As seen in the exploratory section, sentiment has a direct impact on the rating. Also, the reviewers comment on sub-categories in their reviews. So, the aim of this model is to test the impact of these two factors. There are other methods as mentioned in the previous work section [4] and [5], which train the model using the latent parameters. This is external training which might not be as strong as the models that have the latent factors inherently. As the dataset is huge, training time was quite high especially the LDA part. Tuning the parameter using the validation set was time taking as well.

sub-category	no. of cases	M1	M2, K = 10	M2, K = 50
Service	22941	1.5260	1.5050	1.4878
Business Service	3931	3.4428	3.4338	3.4399
Cleanliness	22860	1.4689	1.4013	1.4359
Internet access	187	1.6737	1.6430	1.6227
Front Desk	4225	2.5471	2.5312	2.5248
Overall	23746	1.2977	1.2146	1.2696
Value	22908	1.5066	1.5035	1.5013
Sleep quality	16512	1.2599	1.2292	1.2219
Rooms	21968	1.4520	1.3861	1.4199
Location	21990	1.4805	1.4640	1.4693

Table 3: Results with LDA+sentiment

5 Results and Conclusions

I calculated RMSE for 2 methods.

- M1: Baseline Classifier with no sentiment/lda terms
- M2: BaseLine Classifier + LDA + sentiment

The dataset is not complete - few reviews do not contain ratings for all sub-categories. So, the 2nd column in the table 3 shows the number of cases where this method is evaluated. Topic modeling is done using K = 50 topics and K = 10 topics. The performance is slightly better in the case where K=10. The sparse ness of the topic distribution might be a reason for this behavior. From the table 3, it can be seen that using topic modeling and sentiment analysis to predict rating improves rmse by 4% and 2.5% in K=10 and K=50 case respectively.
The highest improvement in RMSE is in the case of overall value. For calculating this prediction, the sentiment of the whole review is considered. As of now, a very simple method is employed to calculate the sentiment of a subcategory in a review. The prediction for the other subcategories might improve if the sentiment is calculated in a better way. The baseline method doesn't consider the interactions between users and items at all. It computes the user and item bias independently. So, the extra term that is being added captures the missing interaction which should directly improve the accuracy of prediction. The performance of this model is not superior to the other models especially [4] and [5] which achieve an average MSE of ≤ 1. The model is to verify the impact of sentiment of the review on a sub-category rating.

Acknowledgments

References

[1] Dave, Kushal, Steve Lawrence, and David M. Pennock. "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews." Proceedings of the 12th international conference on World Wide Web. ACM, 2003.

[2] Devitt, Ann, and Khurshid Ahmad. "Sentiment polarity identification in financial news: A cohesion-based approach." (2007).

[3] Cui, Hang, Vibhu Mittal, and Mayur Datar. "Comparative experiments on sentiment classification for online product reviews." AAAI. Vol. 6. 2006.

[4] Wang, Hongning, Yue Lu, and ChengXiang Zhai. "Latent aspect rating analysis without aspect keyword supervision." Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2011.

[5] Wang, Hongning, Yue Lu, and Chengxiang Zhai. "Latent aspect rating analysis on review text data: a rating regression approach." Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2010.

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

[7] Koren, Yehuda, and Robert Bell. "Advances in collaborative filtering." Recommender systems handbook. Springer US, 2011. 145-186.

[8] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." the Journal of machine Learning research 3 (2003): 993-1022.

[9] Maas, Andrew L., et al. "Learning word vectors for sentiment analysis." Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. Association for Computational Linguistics, 2011.

[10] McAuley, Julian, Jure Leskovec, and Dan Jurafsky. "Learning Attitudes and Attributes from Online Reviews." (2012).