Reproducing Paper: Latent Aspect Rating Analysis

Chengmin Huang
Ge Yu
Xuehao Wang

Introduction and Overview

In this paper, we will introduce the steps of building Latent Aspect Rating Analysis(LARA) in order to mining the opinion ratings on topical aspects of given certain services type(Hotel reviews, restaurant reviews etc.). Specifically, it will infer the opinion ratings and relative weights focusing on the different aspects based on the reviews from the website such as Yelps, Trip Advisor, Amazon. We'll be using Trip Advisor as our dataset. The basic model of LARA will be based on the pre-defined aspect keywords, whereas the advanced LARA model won't need the supervision of predefined aspect keywords.

Since it's the time where the data is everywhere , LARA is helpful for users to digest a larger amount of the online reviews about a specific entity of tropic. In our project, we use the hotel reviews dataset provided by the Trip Advisor. Nowadays, most websites already decompose the overall rating into different specific aspects. For example, the hotel reviews might have such values, rooms, cleanliness and other categories. Since different users emphasize different aspects, it might still not be informative. Our LARA model can infer the relative emphasis placed by a reviewer on different aspects by digging into their specific reviews. LARA takes review texts about an entity as an input, and will produce output as 1) the ratings on a set of predefined aspects 2) the relative weights that the user placed based on their review texts.

For our implementation, we divide the LARA model into three stages: 1) Data Processing to process the raw data into the format for further processing 2) A Bootstrap algorithm to identify the aspects and segment of the processed review content 3) A Latent Rating Regression model to infer the aspect rating and weights in a review. The specific implementation will be introduced in the next section.

Implementation and Documentation

Data Reading and Processing

In order to process the data, we developed several functions. First, we read the initialized aspect words and stop words. We downloaded the stop words from the nltk library. Then we read the reviews from the json file downloaded from the database, and call the stemming Stop Removal() function to 1)tokenize the reviews into sentences and words 2) Remove the stop words to improve the accuracy of the model 3)Add words to the vocabulary list 4) Make the sentence objects and corresponding review objects. Also, since there are words that have less frequency but could be affecting the overall results as outliers, we developed a function to remove the words that have the frequency less than 5. After the processing step for the data, we call the functions in BootStrap.py to generate the processed word lists as local files.

Bootstrap Algorithm

As mentioned in the paper, the main usage of bootstrapping algorithms is to identify the aspects and segment the review content. We use a bootstrapping algorithm to generate the keywords. This is the code that we used from others and changed some details in order to make it satisfy our own goal . The assignAspect function: this is basically just assigning aspects to sentences. The chiSq and calcChiSq: these two functions are used to generate the chi-square value which is used to tell you how much difference exists between the observed date and the data you would expect to get. populateLists: this function is used to generate the word list. The bootStrap function: it is used to execute the algorithm, it basically implements all the functions I mentioned above. And saveToFile function is just simply saving the file.

All the files are saved in the modelData folder. wList.json is a list of words and their frequency matrix, ratingsList.json is list of ratings dictionary belong to review class, reviewIdList.json is list of review IDs, vocab.json is the list of all the vocabularies that being selected and the aspectKeywords.json is the file that contains the keywords that we obtain using the bootstrapping algorithm. Then we applied a linear rating regression model with these keywords.

Linear Rating Regression

After identification of aspects and segments in review content, the authors applied the Latent Rating Regression(LRR) model to complete the prediction. The LRR model mainly consists of two steps. The input of the LRR model is a list of words and their frequency in the review content. At the beginning of LRR model, the word list is separated into two subset i.e. training set and testing set. The original code separated 75% word frequency data into training set and the rest of data into testing data. In the E-step of training step, the model constrained posterior inference. To be specific, it estimated the updated states using the current parameters. In the

M-step, it updated the parameter estimation and maximized the log-likelihood of the whole corpus.

The model is using the “Overall” rating of each review as the true value and calculating the likelihood between these values and prediction values. However, review text might not be directly related to the overall rating values. One possible improvement is to replace the “Overall” rating with the average of all aspects numbers such as “Service”, “Cleanliness”, and “Location”. Another possible improvement is that since the original model didn’t use the validation dataset during the training step while the validation set could help with the optimization and convergence of parameters.

Further improvement

The results of this model are promising and meaningful. Even though there are some large numbers in the prediction values, the overall trend and relevant values are almost consistent with the actual values. It can be told that the prediction numbers of actual 5.0 rating is greater than the prediction numbers of actual 3.0 rating in the order of magnitudes. We think the possible reason for these large numbers might come from the bag-of-words assumption. This assumption limited the model’s aspect segmentation capabilities. At the beginning of E-step and M-step, the calculation parameters Mu and Sigma are set to the large range of numbers which leads prediction values to increase cumulatively. Even after many iterations steps, it would be difficult to lower these parameters down to the reasonable range. Further improvement can focus on the normalization of these prediction values to the [0, 5] range. In this way, it would be more clear to compare these two kinds of values and evaluate the performance of this model. Furthermore, the initialization of these parameters might also help with the improvement of accuracy.

Also, like mentioned in the paper, we successfully implemented the proposed method of using LRR as the model with the aspect keywords as supervision. However, we fail to implement the advanced model which has a better performance without using predefined aspect keywords as supervision. In the future, we will implement the improved model and compared with our model using bootstrap algorithm and LRR model.

Usage of software

This paper proposed a generative LARA model and used the model to infer the opinion rating on topic aspects. Also it improves the model by eliminating the use of pre-defined aspects of keywords. The software can be downloaded from https://github.com/alany9552/CourseProject. To run the software, users need to make sure they have installed the Python3 environment on their device. Also, the software uses *nltk* stopwords so users should use *import nltk,* nltk.download(‘stopwords’), and nltk.download(‘punkt’) to download the necessary dictionaries. After the completion of installation, users can run the software using python3 ReadData.py, python3 BootStrap.py, and python3 LRR.py sequentially. Then, the running results would show up. The results will list the “ReviewId”, “Actual OverallRating”, and “Predicted OverallRating” respectively. Also, there is a simple classification at the end of prediction that the review would be positive when the “Predicted OverallRating” is greater than 3.0 or negative when it is smaller than 3.0.

The software running can also be customized by users in terms of ratio of training dataset and testing dataset. In the line 46 LRR.py file, the users can change the percentage of the training set. Currently, the training dataset and testing dataset are in 3:1 ratio. In addition to the training ratio, users can also specify the maximum interaction steps and coverage threshold in line 370. Moreover, if they want to change the maximum interaction steps much lower, the changing of line 339 is also needed.

This model is applied to predict the review score of hotels and restaurants based on the review text. Therefore, it can be generalized to the prediction of most opinion tasks. Elimination of predefined aspects of keywords enables this model to be applied in various areas. For example, as mentioned in the paper, it can be applied to reviewer behavior analysis, topic opinion prediction, and personalization recommendations.

Contribution of each team member in case

It is hard to say who contributes to which part of the project since we are basically all doing work that overlapped. We first all read and understand the paper by ourselves, then we gathered and shared our understanding together. After uniting the idea, the team leader (GeYu) gave each of us different work. To be more specific, Chengmin Huang and Ge Yu contributed more in coding, and Xuehao Wang contributed more in testing and processing the data.

Citations and Contributors

Original Implementation of the authors:
http://www.cs.virginia.edu/~hw5x/Codes/LARA.zip

Since the implementation of the LRR and bootstrap algorithm, we extend existing models from the web as our model per the instructor's instructions and directions, and did some changes to fit our models better:
https://github.com/redris96/LARA
https://github.com/seanliu96/LARA
https://github.com/biubiutang/LARA-1

Data Sources:
http://times.cs.uiuc.edu/~wang296/Data/

Overall implementation and Ideas:
"Latent Aspect Rating Analysis on Review Text Data: A Rating Regression Approach",
http://sifaka.cs.uiuc.edu/~wang296/paper/rp166f-wang.pdf

"Latent Aspect Rating Analysis without Aspect Keyword Supervision",
http://sifaka.cs.uiuc.edu/~wang296/paper/p618.pdf