Project Documentation

Reproducing the LARA paper

Kevin Yu - December 12, 2020

Introduction

This section is overview of overall introduction of functions for this project. I am Kevin Yu from China, and I formed a single person team due to some personal reasons when the project start.

Github Url: https://github.com/yukuo78/CourseProject
CMT Url: https://cmt3.research.microsoft.com/CS410Expo2020/Submission/Summary/124

The plan of this project is to reproduce a paper called *Latent Aspect Rating Analysis on Review Text Data: A Rating Regression Approach.* And here is the quotation of the papers abstract:

> *"In this paper, we define and study a new opinionated text data analysis problem called Latent Aspect Rating Analysis (LARA), which aims at analyzing opinions expressed about an entity in an online review at the level of topical aspects to discover each individual reviewer's latent opinion on each aspect as well as the relative emphasis on different aspects when forming the overall judgment of the entity. We propose a novel probabilistic rating regression model to solve this new text mining problem in a general way. Empirical experiments on a hotel review data set show that the proposed latent rating regression model can effectively solve the problem of LARA, and that the detailed analysis of opinions at the level of topical aspects enabled by the proposed model can support a wide range of application tasks, such as aspect opinion summarization, entity ranking based on aspect ratings, and analysis of reviewers rating behavior."*

I used **Python** as the implementation language. I used python3 and numpy and nltk library.

Dependancies:

Library	Description
numpy	For vector and matrix calculation.
nltk*	For some text preprocessing features.

** Note: you'll need to download specific components of nltk after pip installation. Please follow the error message if any, and use nltk.donwload("xxx") with a python shell.*

Implementation

This section introduces the implementation of this project, I divided the code into three main part, namely preprocessing, aspect segmentation and LRR model regression. There are some documentation text build into the code with certain level of detail.

1. Preprocessing (preprocess.py)
loading data from files and do some basic processing, generate vocabulary.

In this section performed below tasks:

- Load hotel review data from file
- Perform preprocessing tasks: lower case, remove stop word, stemming.
- Build a vocabulary and a vocabulary dict.

2. Aspect Segmentation (aspect_segmentation.py)
Use a bootstrapping method to find out aspect terms and distribute all sentences into different aspects.
In this section performed below tasks:

- Load the minimum initial aspect terms, we predefined 7 aspects.
- Calculate Chi square parameters as a K*V matrix

$$\chi^2(w, A_i) = \frac{C \times (C_1 C_4 - C_2 C_3)^2}{(C_1 + C_3) \times (C_2 + C_4) \times (C_1 + C_2) \times (C_3 + C_4)}$$

- Add terms into aspects according to max Chi square value of each aspects.
- Loop 10 iterations

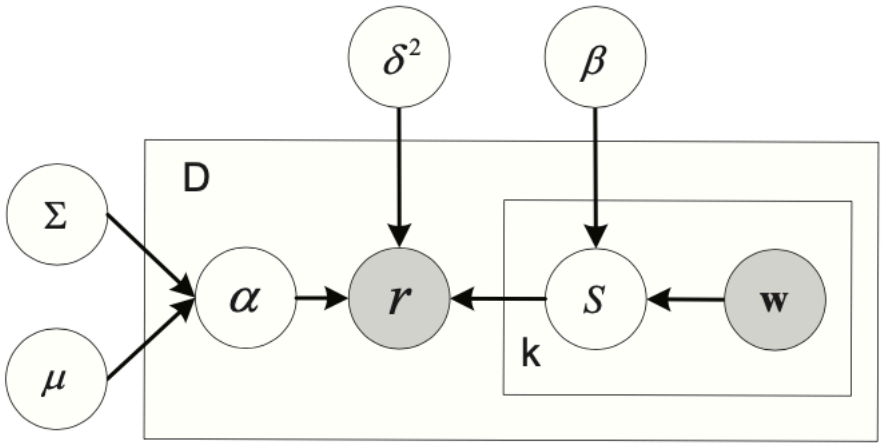
The output of this phase is a considered the ground truth for the LRR model in the next step. I collected all needed data into a class named “Data”.

Algorithm: Aspect Segmentation Algorithm

Input: A collection of reviews $\{d_1, d_2, \ldots, d_{|D|})\}$, set of aspect keywords $\{T_1, T_2, \ldots, T_k\}$, vocabulary V, selection threshold p and iteration step limit I.
Output: Reviews split into sentences with aspect assignments.
Step 0: Split all reviews into sentences, $X = \{x_1, x_2, \ldots, x_M\}$;
Step 1: Match the aspect keywords in each sentence of X and record the matching hits for each aspect i in $Count(i)$;
Step 2: Assign the sentence an aspect label by $a_i = argmax_i\ Count(i)$. If there is a tie, assign the sentence with multiple aspects.
Step 3: Calculate χ^2 measure of each word (in V);
Step 4: Rank the words under each aspect with respect to their χ^2 value and join the top p words for each aspect into their corresponding aspect keyword list T_i;
Step 5: If the aspect keyword list is unchanged or iteration exceeds I, go to ***Step 6***, else go to ***Step 1***;
Step 6: Output the annotated sentences with aspect assignments.

3.LRR Model Regression (LRR.py)
With restrictions of time and considering I only worked on this project alone, this section is only partially finished.

- Perform E step
- Perform M step



Usage

This section I will introduce the usage of my project code. The implementation can be triggered simply from the command line:

> *python preprocess.py*
> *python aspect_segmentation.py*
> *python LRR.py*

Or it can be viewed with more detail with a Jupiter notebook called Demo.ipynb