Final Project

Tuesday 2nd May, 2017
To be send to Maolaaisha.Aminanmu@etu.unige.ch

1 Aim

In this project you will work on predicting user score about the items on the amazon website based on their review. The database you will experiment with is Amazon review dataset. It provides different users' review about the specific items and also the score which ranging from 1 to 5 representing the strength of positive and negative feeling of the reviewers about the item. You can formulate it as a multi-class classification problem where you have 5 class and each instance belongs to one class or a regression problem where your output is taken value from $\{1, 2, 3, 4, 5\}$.

2 Data Description

- amazonreviews: the source of the data (real data before extracting the information) which includes reviews for product from ALWAYS, GILLETTE, ORAL-B, PANTENE, TAMPAX. This is provided just to let you get better understanding the real problem.
- datasets: Each dataset includes the extracted data from the source, include id-review, date, product-id, user-rating, date-other-format, user-id title, and review about products from each brand. What is important is the user-rating (the score) and review.
- stop-words.txt: This file includes stop words which you should use to remove stop words from your feature space (remove the stop words from your dictionary of the text).
- word2vec.txt[1]: This file contains 300-dimensional vector representation for 3 million words which came from google news database. Each word is uniquely represented by a vector. This vector representation keep word's semantic meaning so that words which have similar meaning would have smaller distance while different words have large distance.

3 Processing the data

You need to clean the dataset which included in the file called dataset by removing stop words and do stemmin[2] so that you have most relevant words left in your review which would help to predict the score. After cleaning the data, you need to transform your data in the form of bag-of-words text representation, which is basically each text is represented by the frequency of the words appeared in the document. The words are considered as features and how many times the word appeared in the text is the value of the features. Therefor the feature dimension is the size of your dictionary (total number of unique words appeared in the whole Document). You could also use other methods to represent the document, for example frequency inverse document frequency (TF-IDF) The vector representation of the words(give in word2vec) gives you some additional information on your features, since bag of words doesn't consider the semantic meaning of the words, if you just use bag of words you will lose information like good and perfect is similar words, and they both implies positive feeling of the reviewers towards the product. Therefor, you could try to incorporate the additional information of the features in your model instead of only using the features which is the case in standard learning approach. For that you need to filter vector representation of the words appeared in your dictionary from that file called word2vec. Those words which you don't have vector representation (not included in the file word2vec.txt) you can ignore.

4 Preliminaries

In the following, we will denote:

- $X \in R^{n \times d}$: n reviews, where each raw is representing one review. d is your dictionary size if you use bag of words document representation.
- $Y \in R^{n \times 5}$ scoring for n review if you see it as classification problem where only one column of each raw is non zero representing which score the reviewer has given or $Y \in R^n$: n if you see it as regression problem where each raw is a scalar which is the score the reviewer has given.
- $Z \in R^{d \times 300}$: vector representation for d words where d is your dictionary size (unique words appeared in the whole review text).

We will look for a function f which can map the instances x_i to the corresponding output y_i by using all the information available by X and Z matrix.

5 Model

The final model we would expect is a model that can work on universal data, which means it can give a reasonable prediction on all kinds of the products. Here we have 5 different types of products, so you supposed to train the model

in such a way that if we see an instance which is coming from totally different type of products than we have seen during training phase, the model still be able to predict a reasonable score based the review, and we also expect you are able to incorporate the feature additional information (word vector representation) in the standard learning procedure to help prediction.

5.1 Base Model

Make a base model which uses bag of word representation for the review and do simple prediction of the score (classification or regression, depend on which one you have used in your model) by simply mapping X to Y without using any other additional information.
To compare your model's performance with the base model's, you will have to do a significant test on the number of times your model beats a baseline or not. To do that, you can use the McNemar's test on a 2×2 contingency matrix where the off-diagonal will contain the number of time algorithm X beats algorithm Y and the number of time algorithm Y beats algorithm X, and the on-diagonal is zero.

6 Project Summary

- Prepare the data, get the matrix X, Y,Z.
- Write an algorithm to do the score prediction by using all information contained in X, Y, Z
- Build the base model with using only information containing in X and Y, then compare the performance of your model to the base model.
- Report your approach and results; we want a full picture of what you have done exactly and how. You should also discuss the different performances you have with your methods and why these work or not. What is important is to show us that you have a good understanding on the problem and on how to model it, what are the problems you encounter and how you solve them.

We will be available to help you on technical questions and problems. However you should manage yourself to do the experiments appropriately. Note that this is an open project, you can try many different approach as long as it make sense to get the best performance, creative ideas are always welcomed.

7 Reference

[1]: https://code.google.com/archive/p/word2vec/
[2]:https://en.wikipedia.org/wiki/Stemming