Comp30027 Report

Anonymous

1. Introduction

This project is about to choose appropriate machine learning strategy to predict star ratings for reviews on restaurants. To achieve which, a dataset extracted from real-world customers' reviews is provided. Based on the dataset, systematic analysis is required to performed, in order to predict potential star ratings for each line of review data. In this project, Long Short-Term Memory (LSTM) model is taken into consideration, regarding of other basic ML techniques (NB, SVM, KNN etc.), which finally helps achieve better overall performance.

This report is delivered with following sections: related work, methodology, result analysis and my conclusion.

2. Related Work.

The problem addressed in this project could be concluded as a classification problem. In this subject, previously we have discovered NB[1], SVM[2], KNN[3]. To counter contextual language processing, as the one in this project, models with capability of 'memorizing'information, among many of which, LSTM[4] is the most famous one, which could be a better choice to solve the project.

3. Methodology

3.1 Dataset

Three types of files [5][6] have been provided which are used for training, testing and word embedding.

- Training Data:

review_meta_train.csv, which contains the meta data and label for training instances.

review_text_train.csv, which contains the raw text data for training instances.

- Test Data:

review_meta_test.csv, which contains the meta data for test instances.

review_text_test.csv, which contains the raw text data for testing instances.

- Word Embedding:

review_text_features_doc2vec(50,100,200).zip Each of them contains a matrix of Doc2Vec representation of the review text for training data with different features.

3.2 Method

3.2.1 read file

The dataset used in this project could be divided into 2 types of files, one is meta data included in review_meta_train.csv for training and review_meta_test.csv for predicting. Both of these two files contain data columns as 'id', 'date', 'review ids', which are apparently not useful to produce the prediction, so that they are removed and only the vote columns are remained at this stage. The other is review text, where three 'document to vector' files are adopted for training and testing in this project, each of them is of the same origin reviews file, but with different vector dimension provided.

From doc2vec files, review vectors of 50, 100, 200 dimensions could be read and from meta files, vote vectors of 3 dimension could be read.

3.2.2 Feature Engineering

3.2.2.1pre-processing on doc2vec files.

One of the important process at this stage is combining 2 types of vectors into 1, producing a data vector = review vectors + vote vectors. For instance, assuming for now we're using doc2vec50, for each of its line, [vec1, vec2, …, vec50] is followed by its corresponding vote tags [vote1, vote2, vote3], combining result would be [vec1, vec2, …, vec50, vote1, vote2, vote3]. Bearing the same principle, the rest of vectors and votes are processed and finally producing a list of vectors. Similarly, in this project, more other data vectors are produced—with 103 and 203 dimensions of vector, from doc2vec100 and doc2vec200, originally.

Before implementing models that running on such data vectors, another process called Normalization is adopted, aiming at reducing the side-effects of irregular data. For the dataset provided, the values inside any review vector are within the range -10 to 10, whereas the values in votes would be 0 to 30 and even higher. The mismatching between such values would affect the models running onto them leaning unbalanced between reviews and votes. Hence, method of normalizing data is used—normalize from sklearn.preprocessing package.

3.2.2.2 Pre-processing on raw text files (used for LSTM Model)

1. Tokenize the text. Ignore the spaces, punctuations and other non-numeric & non-alphabetic characters. In order to simplify content prior to the next step of processing.

2. Sentence padding. Regarding of LSTM's inherited data requirement that is all input data need to be the same topology (length, vector dimensions). Find the length of the longest sentence (max length), and for each sentence, compensate it with a padding string (e.g. CONST_PADDING = '<PADDING>') until the padded sentence reached the max length.

3. Word embedding. LSTM cannot directly understand real world words. So that we need to find a path to "translate" them into a format that the model can understand. There are bunch of ways to do word embedding. In this project, *Word2Vec* which is imported from *gensim.models* has been chosen to do the task.

3.2.3 Models

In this project, several classifier models are taken into consideration, running on the same set of data vector for validation and comparisons, shown as following:

1) Gaussian Naïve Bayes—the most basic model that doing classification task, result produced would be use as the basic comparison against other more complicated models.

2) Support Vector Machine—using the separate panel as its approach to perform classification, demonstrated as the most effective way to classify among a set of

instances.

3) K-Nearest Neighbour—considering the number of K nearest instances' classes as the ones in hand, the value of K used would potentially affect the model's strength.

4) LSTM Model -one of the most effective solution to deal with sequence prediction problems. This model has a property of selectively remembering patterns for long durations of time.

Moreover, I want to discuss more for my attempt of LSTM.

Practically, a working LSTM can be visualised as following graph.

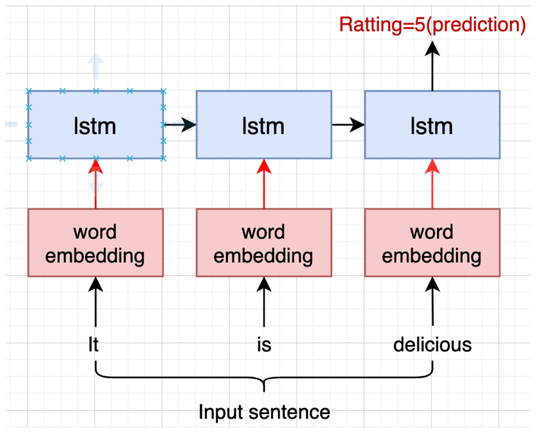

Figure 1: LSTM by example

consider we have a sentence of three words "It is delicious", and supposing the model by far learns knowledge as : 1) At the beginning of each sentence giving it a prediction as 1 ; 2)Encountering a positive word (like delicious), the prediction of rating will be scaled up; 3)Encountering a negative word (e.g. bad) , the rating will be scaled down.

Go back to the example, the model keeps reading the words, Both "it" and "is" are neutral, when it meet "delicious", the model knows it is a strong positive word and scale up the rating by 2 and make the final prediction to be 5.

In terms of using LSTM, we need to split all the training instances into groups with a certain batch size. Using every 19 batches to train, and using the next batch to do the evaluation, namely calculating the accuracy of prediction. By doing so, we can evaluate the model evenly across all training data instances Moreover, by changing the batch size, the batches that used to do the evaluation can be generated more randomly

3.2.4 Evaluation

Evaluating the effectiveness of each model within the training set would give the most information that whether the current model is useful or not and more or less effective when compared to others. Hence, several groups of evaluation would be performed, with variations in different model, data vector sets (53. 103 and 203), and model settings. Within each group, as the precision, recall and accuracy have the same trend under variation of the group setting as above, only the accuracy is taken into account for simplification.

4 Result and Analysis

4.1 Same Training Data, Various Models（NB, SVM, KNN）

Accuracy	Naïve Bayes	SVM	KNN (K = 5)

53 data vec	78.35	80.84	77.21
103 data vec	79.34	82.26	74.93
203 data vec	78.35	84.05	74.22

Table 1: model accuracy over datasets

The table above could be visualized as following graph:

(x-axis: vector-dim; y-axis: accuracy)

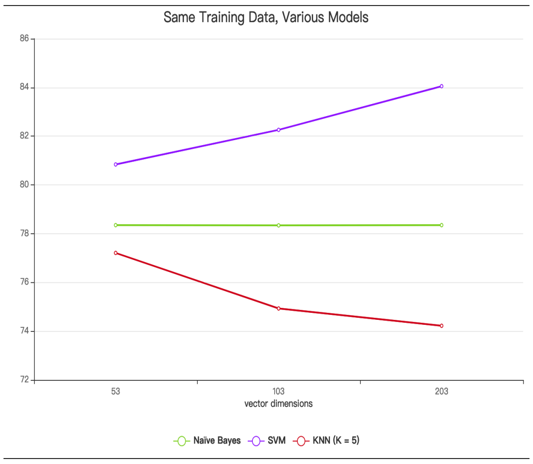

Figure 2: visualisation of table

From the graph above we can investigate that SVM model always has a better performance than the two other models. And the accuracy of SVM will increase with higher vector dimensions, since SVM is sensitive with the amount of data.

Naive Bayes treats all features of the dataset to be independent, which is not true in this project's case. So, the wrong assumption will influence the performance of NB model. However, SVM maintains the interactions between the features. From a theoretical point of view, NB is based on the probabilistic and SVM is based on geometric. So that SVM can do better classification for this case whose features have a lot of interactions.

It is obvious to see from the graph that KNN (with K=5) has the worst performance. Principally, KNN classifies the test input according to the majority class of the K nearest training instances. which intrinsically not using all embedded features in the datasets –only concerning on the K nearest instances and does not take care of outliers. Moreover, the time computation needed by KNN is much larger than other models.

4.2 Attempt on LSTM

In my original assumption, LSTM is very suitable to deal with this case since this model is always a great tool for anything that has a sequence. Since the meaning of a word depends on the ones that preceded it. Principally, LSTM would likely to produce a good performance, whereas in this project, the number of training instances are not big enough. So that the final performance is not as good as we expect, reasons as following:

1.The model cannot learn much due to the small size of training data so it cannot predict very well. Deep learning is always not efficient with small datasets.

2.If we try to reuse the training data (e.g. Iteratively running on the same small dataset) in order to let LSTM learn more, the problem of overfitting would be very serious. Even though we tried to reduce the number of hidden units to alleviate the problem, the final performance of LSTM is still not good, the accuracy is just above %75.

Further work could be conducted once we have sufficient data, to prove whether LSTM can outperform or not when comparing to other models.

5. Conclusion

I use three simple models (NB, KNN, SVM) to do the prediction, and SVM is the best choice and has the highest accuracy. Additionally, according to the trend of the graph above, better performance can be achieved by SVM if more data provided (e.g. use 250,300+ dimensions in training vector).

Unexpectedly, LSTM underperforms under the project dataset settings. The reason behind could be the size of dataset is rather small which could be potentially rectified if larger dataset provided.

6. Reference

[1] Hastie, Trevor. (2001). The elements of statistical learning: data mining, inference, and prediction: with 200 full-color illustrations. Tibshirani, Robert., Friedman, J. H. (Jerome H.). New York: Springer.

[2] Cortes, Corinna; Vapnik, Vladimir N. (1995). "Support-vector networks" (PDF). Machine Learning. 20 (3): 273–297.

[3] Altman, Naomi S. (1992). "An introduction to kernel and nearest-neighbour nonparametric regression" (PDF). The American Statistician. 46 (3): 175–185.

[4] Sepp Hochreiter; Jürgen Schmidhuber (1997). "Long short-term memory". Neural Computation.

[5] What Yelp fake review filter might be doing? A. Mukherjee, V. Venkataraman, B. Liu, and N. S. Glance, *ICWSM*, 2013.

[6] Collective Opinion Spam Detection: Bridging Review Networks and Metadata.Shebuti Rayana, Leman Akoglu, ACM SIGKDD, Sydney, Australia, August 10-13, 2015