COMP30027 Project 2 Report

1. Introduction

Machine learning is the art of prophecy. By training a model with a huge amount of data, the system can accurately predict the behaviour of an instance, based on given features. And the key step is increasing the predicting accuracy among all types of data.

In this Project, the methods to improve predicting accuracy of sentiment analysis problem is investigated. The goal of sentiment analysis is to automatically identify and extract polarity (e.g. positive, negative, or neutral) from reviewer text.

In our project, we have collected some data of review text and the corresponding rating with some other data of reviewers. And we want to predict star ratings for reviews on restaurants, by building and critically analysing some supervised Machine Learning methods.

2. Model selection

2.0 Input data analysis

For each instance prediction, a paragraph of review text is provided in a csv file with another csv file contains corresponding meta data, including rating of the review. The concept is ordinal data with only rating 1, 3 or 5, approximately 68.72% of ratings are 5, 8.32% are 1 and 22.96% are 3. This suggests the rating of review is mostly polarized. This makes sense, as the review recorded are generally either praising or criticising with strong emotion. Customers with moderate feeling of a restaurant tend not to write down any feeling. The instances are text, we pre-processed the data by vectorize the review paragraph. This vectorized dataframe has high dimension. Since we also obtain part of testing data set, we can select model though testing the test dataset and figure out its accuracy. As the test data set is a part of real-world data, the model accurately predicts the test data set would hypothetically be accurate on whole real-world data as well.

2.1 SVM

Among all types of SVM models (RBF, linear, Polynomial), we chose the linear SVM which applies a linear support vector to classify clusters.

As the training data contains frequency of the word, and each word usually has a typical tendency in diction. Each vocabulary has positive or negative meaning, the review with a lot of positive words would generally obtain a higher rating. As a result, the frequency of vocabulary used is linearly correlated with the rating. Although using a “RBF” or polynomial kernel enriches feature space to add more power of explanation, they may cause overfitting problems which increase the model variance. (Gori 2018)

Moreover, SVM with a Gaussian or polynomial kernel is generally good at training instances with low dimension but huge number of instances. But SVM with a linear kernel is generally good at training instances with high dimension regardless the number of instances.

Therefore, we choose to use SVM with a linear kernel.

2.2 Neural Network

Neural network is deep learning technique that aggregates simple logic relationship between each layer to figure out the predicting system.

We chose hyper parameters based on the input data, such as activation function, number of layers, number of nodes in each layer.

Since ReLU has better time complexity and will not accelerate the gradient descent, we chose it as our hyper parameter.

After testing several hidden layer sizes, we achieved balanced time complexity and classification power by obtaining 3 layers [80, 100, 100].

2.3 Random Forest (Bagging)

Random forest models involve instance manipulation techniques, so we have decided to use bagging type of random forest rather than a boosting method. Because boosting involves iterative sampling to minimise the

instance bias, it tends to have higher model variance due to over focus on some samples.

In practice, if we limit the max depth of each tree, we observed that the performance of system was very poor (69%). Therefore, we use cross validation to fully examine each set without limitation.

In addition, we chose to use doc2Vec50(with entropy) as input data frame rather than simple count Vector (with Gini coefficient). This would help investigate the interaction between vocabularies further.

2.4 Stacking

Stacking is a feature manipulation method, that can theoretically reduce model bias and variance simultaneously, by combining different models.

In stacking, we have to choose different types of classifiers to classify the raw data to get a meta classifier. In generating metadata, we also have to decide the number of folds to do cross-validation.

While implementing, we divide the data set into 5 folds and use cross validation method to generate features in the meta classifier. make predictions of rating in the fold. Finally, we trained a logistic regression model in classifying the data. As logistic regression applies gradient descent to find a classifying function by optimising a convex set of errors in predicting the rating (Rodrigo, 2019).

Due to previous testing of different classifiers, we finally select four most effective types of models as those base classifiers: Gaussian Naive Bayes, Random forest, Linear SVM, MLP Neural network. In addition, we also train another Linear SVM information provided in meta data including “vote funny”, “vote cool” and “vote useful”, as we reckon these features are highly corelated with final rating.

SVM and Neural network are the best performing systems with single classifier. Random forest and Stacking are combinational classifiers based on instance manipulation or feature manipulation.

We finally trained a classifier to predict rating based on meta data of classifier. In testing phase, we tested both 5 folds and no fold model.

3. Critical analysis

In evaluation of a system, we focused on the performance of a system in both accuracy and runtime efficiency. Theoretically, an accurate model with extremely high time complexity is deficient to use in practice, it is unfriendly to software development. Accuracy is also significant as an inaccurate model is not trustworthy.

3.1 SVM

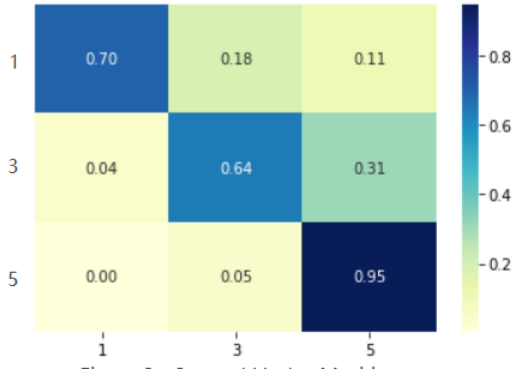

Figure 1-SVM

In practice, we tried support vector machine classifiers use a linear, RBF or polynomial kernels. The testing result aligned with our hypothesis that the linear kernel is the most powerful classifier, achieving accuracy of 85% by cross-validation. This implies linear SVM has balanced model bias and model variance; it is neither underfitting nor overfitting the data set too much. (Figure 1)

Suppose the number of instances in the data set is n. The time complexity is $O(n^3)$. It grows at least like n^2 when C, the bound of the support vector, is small and n 3 when C gets large. (Bottou, 2007) The space complexity is $O(n^2)$. The model is overall very slow, although the accuracy is high.

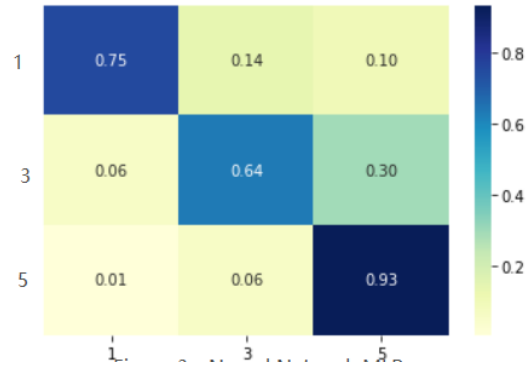

Figure 2-Neural Network

3.2 Neural Network

{'fit_time': array([86.64887404, 89.47489405, 89.27473116, 88.80780244, 8.8344965]),
'score_time': array([0.43709898, 0.42960143, 0.43709922, 0.42709661, 0.42809653]),
'test_score': array([0.85289403, 0.84324902, 0.84609904, 0.84642794, 0.68727726])}

Figure 3-MLP Score of Cross validation

The predicting accuracy of neural network system is 81.5%, averaging 5 cross validation sets.

However, neural network demonstrated great deficiency in predicting middle class rating =3 (fail rate=0.36), based on random hold out (Figure 2).

The time complexity is $O(nt(ij + jk))$ and space complexity $O(i + j + k)$, where n is number of instance, t is number of iterations, i, j, k are number of nodes in first 3 layers, which are 80, 100, 100 in the model built.

During training, its model accuracy is sometimes very high during cross validation (Figure 3). This suggests this system may have instance bias.

As the predicting accuracy of model varies a lot for different cross validation set, the model variance is comparatively high.

3.3 Random Forest (Bagging)

Random forest has time complexity $O(nlog(n)dm)$ and space complexity O(dm), where d is max depth of the trees and m is the number of trees, n is the number of instances. The algorithm is faster as well.

The model variance is high, as it is very inaccurate in predicting instances from rating=1&3, although it predicts the majority class rating =5 accurately. (Figure 4) It suggests neural network is not sensitive with minority class of data. It focuses too much on improving overall accuracy by improving the accuracy of majority class labelling, sacrificing the predicting accuracy of the minority class.

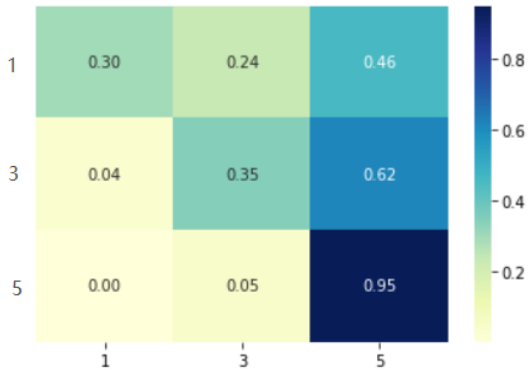

Figure 4-Random forest

3.4 Stacking

Time complexity of Gaussian naive bayes is $O(nd)$, space complexity is $O(nk)$, where n is number of instances and k is number of class labels. Time complexity of Linear SVM is $O(n^2)$ while space complexity is $O(1)$. Time complexity of other instances with Linear SVM is $O(n^2)$ with space complexity $O(1)$. Random forest has time complexity $O(nlog(n)dm)$ and space complexity $O(dm)$, where d is max depth of the trees and m is the number of trees. Neural network has The time complexity is $O(nt(ij + jk))$ and space complexity $O(i + j + k)$, where n is number of instance, t is number of iterations, i, j, k are number of nodes in first 3 layers, which are 80, 100, 100 in model built. The final logistic regression step obtains time complexity of $O(nd)$, and its space complexity is $O(d)$, where d is dimension=5, n is number of instances. (Kumar, 2019) Therefore, the whole stacking system has time complexity is $O(f(n\text{^}2 + nd + n\text{^}2 + nlog(n)dm + nt(ij + jk)) + 5n) = O(f(nlog(n)dm + nt(ij + jk)))$, as they are the most dominated term, where f is the number of folds applied in the end. The run time complexity is very slow compared to just implement a single system.

And the space complexity is $O(1 + nk + 1 + i + k + j + d) = O(nk)$. Overall, the memory space it takes is not very large.

In practice we tested no fold version at first. In practice, the stacking system has the best accuracy. The accuracy of no-fold is satisfyingly 84.7%, the feature manipulation method successfully reduces bias and variance of the model in practice. (Figure 5)

However, the accuracy of 5 folds stacking is only 67%. This demonstrates although cross-validation has reduced model variance, the model bias is increased somehow. After the splitting, our model tends to predict more dominated class rating=5, resulting in inaccurate metadata. The accuracy in classifying data in each other groups is largely reduced. Finally, the stacking model is underfit with great instance bias.

Overall bias od model is reduced, it predicts more accurately than any singular model.

Variance is reduced as different types of classifiers supplementarily handle instance prediction correctly; successfully cover the mistakes each other. These made the final prediction model more accurate. By applying n-fold cross-validation, we further reduce the model variance by allowing models build on proportion of data set to classify different instances, rather than just the training set. But this did not avoid overfitting. The instance variance is indeed increased.

Furthermore, we tested some other stacking models, with only SVM, Neural network, metadata. However, this model does not performed well in practice too.

4. Future improvement

Firstly, we should try to collect balanced data set without any dominated class label. There shall be more instances from rating=1 and 3.

Secondly, we can try to decrease the dimension of vectorized dataset by PCA. Ideally, the dimension reduction would eliminate overfitting and minimise the instance variance in predicting the data.

Thirdly, we can try different model combinations in building up a stacking system, such as Multinomial NB, KNN. Or reducing or aggregating different models together, in different combinations.

Fourthly, we can implement additional user-orientated system to identify general favour orientation of users. Combined in predicting the final rating.

Finally, we shall try different models with different numbers of folds to investigate the ideal number of folds of data to balance the model. Perhaps, 2 folds or 3 folds would be the ideal solution for stacking system.

Model	processing	BestScore	KaggleScore
MultionmialNB	CountVect & TfidfTransformer	84.14%	82.66%
GaussianNB	doc2vec50	67.02%	\
Neural Network	CountVect & TfidfTransformer	84.69%	83.42%
SVM	CountVect & TfidfTransformer	82.66%	84.66%
Randomforest	CountVect & TfidfTransformer	75.35%	73.92%
Stacking	LogeticReg	\	77.86%
Stacking	Randomforest	\	82.85%
Stacking	SVM	\	84.70%

Figure 5 Accuracy table of different models

5. Conclusion

Among all systems implemented and tested, stacking system without any folds is the most accurate one. However, the time complexity of the system is very high, we may find is inefficient in practice.

On the other hand, Linear SVM system obtains almost the same accuracy level, but its time efficiency is much higher. It may be a better solution.

Although Neural network obtains almost the same accuracy as Linear SVM in predicting, its accuracy is not stable. We do not recommend this system based on its big instance variance.

Although we tried to implement multiple

classification techniques, even the highest accuracy 84.7% of a system build is still not satisfying enough. In addition, we fail to generate a system that is good at predicting rating = 3 class. All our models have a polarized tendency with categorizing a review with rating 1 or 5.

In summary, English is a very complicated language, it is relatively hard to computationally predict rating just based on some text, more dimension of data is required.

6. References

Bottou, L. & Lin, C.-J., 2007. Support Vector Machine Solvers. *Large-Scale Kernel Machines*. Available at: https://leon.bottou.org/publications/pdf/lin-2006.pdf.

Gori, M., 2018. *Machine learning: a constraint-based approach*, Cambridge, MA: Morgan Kaufmann Publishers.

Kumar, P., 2019. Time Complexity of ML Models. *Medium*. Available at: https://medium.com/@paritoshkumar_5426/time-complexity-of-ml-models-4ec39fad2770 [Accessed May 27, 2020].

F MELLO, RODRIGO. ANTONELLI PONTI, MOACIR, 2019. *MACHINE LEARNING: a practical approach on the statistical learning theory*, S.l.: SPRINGER.

Mukherjee, A., Venkataraman, V., Liu, B. & Glance, N. What Yelp fake review filter might be doing? 7th International AAAI Conference on Weblogs and Social Media, 2013.

Rayana, S. & Akoglu, L. Collective opinion spam detection: Bridging review networks and metadata. Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015. 985-994.