Multi-class Classification on Textual Information with Generative Model & Discriminative Model

Sarah Zhou
shi.q.zhou@mail.mcgill.ca
Ziqing Liang
ziqing.liang@mail.mcgill.ca
Zuchan Fang
zuchan.fang@mail.mcgill.ca

Abstract

Multi-group classification is often used in the classification of text information. In this project, we investigated the performance of a generative model (Naive Bayes) and a discriminative model (Logistic Regression) on predicting the sentiment of two distinct textual datasets: 20 news group dataset (20News)(Scikit-learn) and IMDB movie review dataset (IMDB) (Maas et al., 2011). We found that selecting the most independent features of text information can help Naive Bayes improve classification accuracy. However, when logistic regression is trained, it can find the optimal parameters regardless of whether there is a optimization between the features, and thus perform better. Compared to Naive Bayes, Logistic Regression will not be limited to feature engineering when applied. Indeed, through our learning, the use of feature selection has led to a slight decrease in the classification accuracy of Logistic Regression.

1 Introduction

The goal of this project is to conduct multi-class classification of 2 large-size textual datasets: 20 Newsgroup dataset with 18k newsgroups posts on 20 topics and IMDB movie reviews dataset with 50k full length review. To tackle this problem, we first investigated different modifications over the two text datasets to make them more suitable for digitization. For the purpose of extracting more valuable information from long texts and reducing running time at the same time, we selected the top 30% of the data with the most independent features from the tens of thousands of features in the dataset. In terms of classifers, we mainly implemented and examined two individual models: Naive Bayes (NB) and Logistic Regression (LR). For Naive Bayes model, we gained around 0.68 5-fold cross-validated accuracy for the 20 Newsgroup dataset and and 0.85 5-fold cross-validated accuracy for the IMDB dataset with the test prediction accuracies of 0.637 and 0.839 respectively. As for Logistic Regression (LR), we approached 0.74 5-fold cross-validated accuracy for the 20 Newsgroup dataset and and 0.89 5-fold cross-validated accuracy for the IMDB dataset with the test prediction accuracies of 0.674 and 0.879 respectively.

We found that the NB performance was improved under the premise of independent data features, while the LR overall performed better than NB on both datasets. At the mean time, their performance were all limited, regardless of whether they were generative or discriminative. In order to break this limit, some extension attempts and further optimization directions were also discussed by the end of this study.

1.1 Related Work

In 2013, a research was conducted to increase the performance of Naive Bayes classifier for text information (Mohanapriya and Jayabalan, 2013). The authors combined feature selection, negation handling and word n-gram techniques to improve the performance, in which they reached their best result of 88.8% .
The 20 Newsgroups dataset is a typical case when deploying textual information classification. Previous studies have used this dataset to assess the performance of multiple models. In 2017, the author IRJET team found that through all the experimental models, the SVM gave the best performance.(Pradhan, 2017). Coincidentally, Rennie and Rifkin compared the performance of NB and SVM on multi-class text classification and found that Support Vector Machine (SVM) outperformed NB by approximately 10% to 20% back in 2001(Wang and Manning, 2012). As for the IMDB dataset, a research in 2019 applied eight classifiers, including Naive Bayes, Random Forest, Decision Tree, Support Vector Classifier, K-Nearest

Neighbours, Ripple Rule Learning, Bayes Net and Stochastic Gradient Descent(Yasen and Tedmori, 2019). In their experiment, Ripple Rule Learning was found to give the worst results, whereas Random Forest outperformed other classifiers. At the same time, in 2019, a study observed that SVM and LR were efficient and yielded great performance (Gamal, 2019).

2 Dataset

2.1 20 Newsgroup Dataset (20 Newsgroup)

The 20 Newsgroups data set is a collection of approximately 18,000 newsgroup documents, which are partitioned approximately evenly across 20 different topics. Some of the newsgroups are very closely related to each other (eg. comp.sys.ibm.pc.hardware/comp.sys.mac.hardware), while others are highly unrelated (eg. misc.forsale /soc.religion.christian) (Jason, 2008). A true lable distribution of this data set is shown in figure 1.

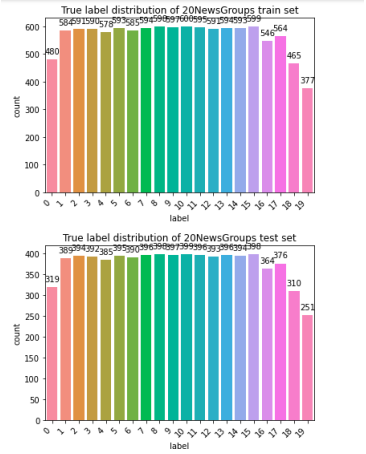

Figure 1: True labels distribution for the 20Newsgroup dataset

2.2 IMDB Dataset

IMDB Reviews dataset includes 50,000 full-length reviews, of which each was labeled with binary values, where one indicates that a review is positive, vice versa. The dataset was split into 25,000 training and 25,000 testing samples primarily. We used 5-fold cross-validation during training over the 25,000 training samples; the same process was applied during hyper-parameter tuning.

2.3 Preprocessing

Using functions from the nltk library (Nltk, 2020), we preprocessed the two datasets in the same manner. We tokenized, lemmatized, and finally, removed stop words. An alternative exploration for lemmatization is to stemmatize the words instead.

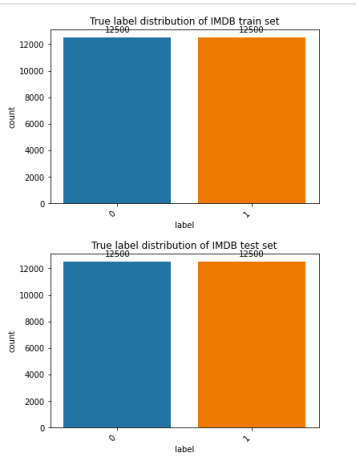

Figure 2: True labels distribution for IMDB dataset

2.4 Feature Extraction

The suggested feature extraction method is to use CountVectorizer provided by the scikit-learn library, which vectorizes each text using the raw count of words (Scikit-learn, 2020a). We also explored the TfidfVectorizer, which uses a TF-IDF transformer on top of the raw counts of CountVectorizer. TF-IDF stands for term frequency, inverse document frequency. It normalizes the weight given to each word, and scales down the impact of tokens that occur very frequently that are hence empirically less informative than features that occur in a small fraction of the training corpus (Scikit-learn, 2020c). In general, TF-IDF vectorized version of the corpus performs better on text classification tasks than the raw counts to construct the feature matrix.

2.5 Feature Selection

As the feature matrix is rather large (25000x24363 for the IMDB dataset and 11314x16282 for 20 Newsgroup) and we observed overlaps/dependency between features (Figure 3), we additionally performed feature selection to have a comparison in the results. Using the scikit-learn library, we selected 30% of the most independent features using the provided mutual information classifier function (Scikit-learn, 2020b).

3 Models

In this study, we implemented the Naive Bayes model with the multinomial likelihood from scratch

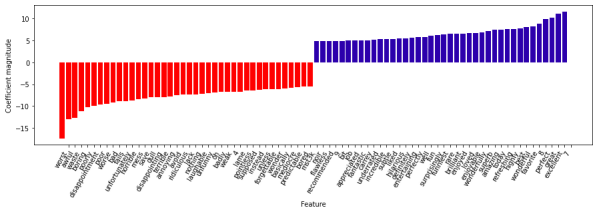

Figure 3: Top 40 features for IMDB

and directly implemented the Logistic Regression from the scikit-learn package within Python.

3.1 Generative: Multinomial Naive Bayes

Naive Bayes is based on Bayes' theorem, where the adjective Naïve implies that features in the dataset are mutually independent. In the context of text classification, the probability that a document d_i belongs to a class c is calculated by:

$$P(c|d_i) = \frac{P(d_i|c)P(c)}{P(d_i)} \quad (1)$$

We adopted the multinomial likelihood and implemented the Multinomial Naive Bayes algorithm to convert the conditional probability into calculating the number of times each word appears in a certain type of document (word frequency).

3.2 Probabilistic Smoothing

In this study, we introduce the smoothing hyperparameters for the prior probability and posterior probability of Multinomial Naive Bayes: α and β, which prevent the probability result from reaching 0. Selecting the best α and β that led to the highest average 5-fold CV accuracy of the model helped achieve The purpose of tuning hyperparameters and optimizing the model.

3.3 Discriminative: Logistic Regression

LR incorporates a statistical approach to assign discrete labels to data-points and finds any decision boundary that separates classes using the following cost function (Figure 4)(Sperandei, 2014)

$$J(w) = \sum_{n=1}^{N} y^{(n)} \log\left(1 + e^{-w^T x}\right) + \left(1 - y^{(n)}\right) \log\left(1 + e^{w^T x}\right),$$

Figure 4: LR cost function

where N is the number of instances or data-points, y(n) is the label for data-point n, x is the matrix of instances, and w is the weight vector.

Using GridSearchCV from scikit-learn, we run the experiment with 5-fold cross-validation, to search the best hyperparameter C, the inverse regularization strength. For model training, the best accuracy is retained. Since regularization has a known role in reducing overfitting(Druck, 2008), the hyper parameter was chosen to discover the amount and type of regularization that are suitable for the classification tasks.

4 Results

4.1 TF-IDF vs. Raw Count

As expected, the TF-IDF normalized version of the texts achieved higher training accuracy than the raw counts. After performing 5-fold cross-validation of the Naive Bayes model, the accuracy of improved from 0.681 to 0.740 for the complete the news dataset, and from 0.840 to 0.863 for the complete IMDB dataset. Therefore, for all subsequent experiments we applied TF-IDF normalization.

4.2 Naive Bayes vs Logistic Regression

In figure 5,6 and 7, the trend of LR and NB shows are similar. The NB accuracy trend grows as the training set size grows, which follows our expectation. In the news dataset, the accuracy range is roughly 10% from 20% to 100% of the training set size. However, strangely for IMDB, 20% of the training set would already ensure more than 80% accuracy. We calculated the f1 score to examine the potential flaws, but obtained high values (Table 1). We conclude that this phenomenon may be related to the very distinct words for positive/negative sentiment (Figure 3); some models need not a large amount of data to perform binary classification.

On average, logistic regression performs better on both datasets (Table 1). We found an optimal inverse regularization strength of 10 for both datasets. As smaller values specify stronger regularization, our datasets did not need much regularization.

4.3 Selected Features vs All Features

Applying the same method on the subset of selected features (top 30% of the original), on the IMDB dataset, the accuracy increased by 0.018 for NB. On 20News with NB, we found that the training accuracy lowered by 0.059, and test accuracy lowered by 0.028. The 2.8% difference is significant, but considering that only 30% of the features were used, we can conclude that more features would ensure the machine learning model to better capture

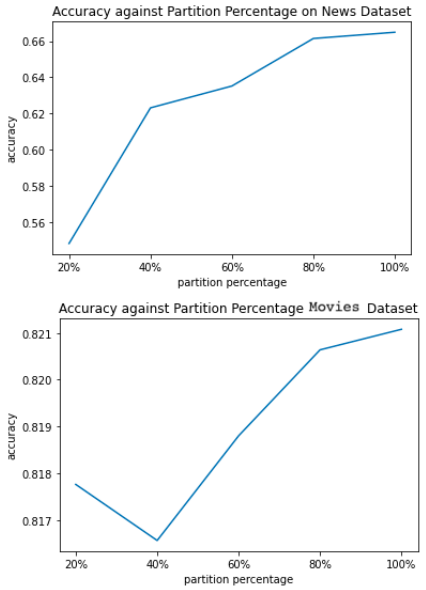

Figure 5: NB Accuracy Trend vs Training Data Size

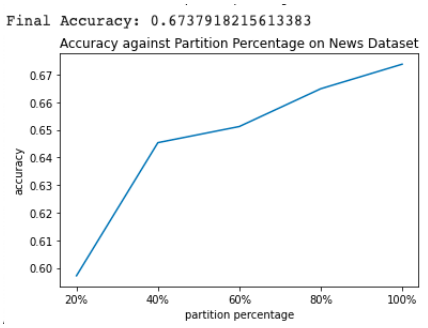

Figure 6: LR Accuracy Trend vs Training Data Size (20 News)

the subtle distinctions, but independent features have far more contribution than dependent ones in the case of Naive Bayes.

4.4 Classification Analysis on NB

Based on our current result, the prediction accuracy of NB is slightly worse than that of LR. To dig deeper into the reasons behind, we compared the prediction results of NB with the standard data on each class of the two datasets.

As observable in figure 8, in the 20News data, the classification results of NB in most of the groups are consistent with the standard results. However, only a few groups show large errors, namely for group 11(rec.sport.hockey), 16(soc.religion.christian) & 20(talk.religion.misc). In these three groups, the NB model believes that the data attributable to groups 11 and 16 is far more than their actual data volume; whereas for group 20, the model hardly has enough data to be classified under this category. Taking a closer look at the main features under the 3 groups, it is obvious to find that group 16 & 20 are both topics related to religion, in which there are many correlated features under both groups. Recall that an important assumption that NB is making is that the features in the dataset are mutually independent. Hence it is reasonable to suspect that the model confuses the features of these two groups and mistakenly classified most of the data originally belonging to group 20 into group 16. In the same way, group 11 mainly collects hockey-related news groups in the sports category, and there are other sports-related groups whose counts are lower than true labels that can be confused with group 11 (e.g., group1: alt.atheism & group 10: rec.sport.baseball).

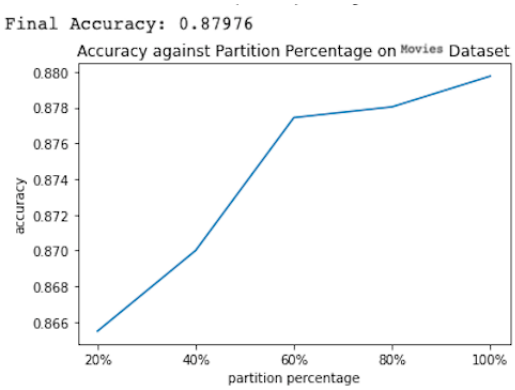

Figure 7: LR Accuracy Trend vs Training Data Size (IMDB)

5 Additional Experimental Attempts

5.1 SVM

Based on the existing results, we observed that the prediction accuracy of the discriminative classifier is better than that of the generative classifier. However, simply comparing NB and LR is not sufficient to prove this inference. In order to further verify our conjecture, we introduced another powerful discriminative classifier, the Support Vector Machine (SVM) model into the experiment. This method is built on the concept of hyper planes, namely to consider each pair of data points until it finds the closest pair that are in different classes and places a separation hyper plane midway between them. We will use the scikit-learn functions for this experiment.

We tested a few kernels (the hyperparameter). The best kernel is the Radial Basis Function kernel with inverse regularization strength of 10 that gives an

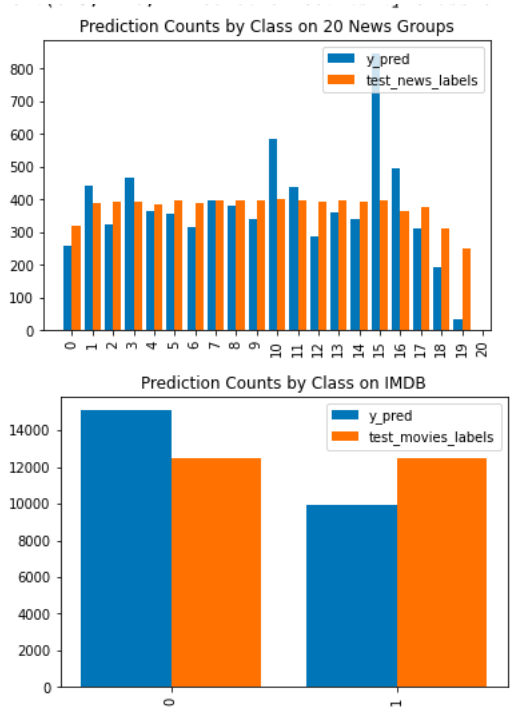

Figure 8: NB Prediction Counts by Classes

prediction accuracy of 0.633 for news dataset and 0.885 for movies. This kernel has infinite dimensions and allows to neatly separate the data. Due to time constraint, we did not perform a exhaustive hyper parameter search. One could expect a better accuracy by further tuning.

5.2 Ensemble of Generative Model and Discriminative Model

Up to now, we have only studied the classification of two datasets on a single model (generative or discriminative). We have roughly reached a rudimentary conclusion: the performance of a single model is limited on reaching high accuracy. A related research published in 2014 stated that model combination has been shown to perform better than any individual technique. The ensemble best benefits from integrating models that are complementary(Mesnil et al., 2014). Being inspired by this idea, we further attempted to improve the classification accuracy on the IMDB dataset by combining the generative component and the discriminative component to a NB-LR model.

Our specific implementation method roughly aligns with the previous work(Mesnil et al., 2014). Since a generative model defines a distribution over the input, by training a generative model for each class, we can then use Bayes rule to predict which class a test sample belongs to. In particular, we used two Naive Bayes models to train the binary classes of "positive" and "negative" reviews, so that the positive model will be more sensitive to positive unknown reviews, and same for the negative ones. Two NB models will respectively obtain the maximum likelihood corresponding to the positive and negative evaluations for a data, and then the ensemble model will use the ratio of the two maximum likelihood values as one of the reference features for the next step of the discriminant classifier, which consequently makes the final discrimination on the data.

Aligning with our expectations, the prediction accuracy of the model on the IMDB dataset is higher than that of NB and LR alone, reaching 0.916 as the 5-fold CV accuracy and 0.912 as the best prediction score.

6 Discussion and Conclusions

From the experimental results, the performance of SVM and Logistic Regression are similar, and moderately better than NB. This further confirmed our conjecture that the discriminative classifiers perform better overall. Moreover, the generative model and the discriminative model combination has been shown to perform better than any individual technique.

A flaw in our implementation is that the vectorizers used are based on words occurrences and do not consider word types. In reality, an adjective may be way more important in sentiment analysis than a determinant for instance. A direction for further improvement is to do POS tagging or add a weight function.

One can also attempt to improve the accuracy of the experiments using deep learning methods and libraries such as TensorFlow. These methods use neural networks, which is a powerful tool in non-linear learning and NLP.

7 Statement of Contributions

All authors contributed equally to the report and coding components of this assignment. Author 2 developed the Naive Bayes model. Author 1 was in charge of data prepossessing and Author 3 contributed to the Logistic Regression experiments.

Model	Dataset	F1 Score	Accuracy Score	
			Train	Test
Naive Bayes	20N		0.740	0.665
	20N (selected features)		0.681	0.637
	IMDB	0.800	0.863	0.821
	IMDB (selected features)		0.852	0.839
Logistic Regression	20N		0.743	0.674
	20N (selected features)		0.730	0.640
	IMDB	0.878	0.888	0.880
	IMDB (selected features)		0.878	0.880
SVM (kernel = rbf)	20N (selected features)			0.633
	IMDB (selected features)			0.885
NB-LR	IMDB		0.916	0.912

Table 1: Reported accuracy for model variants.

References

G.; McCallum Druck, G.; Mann. 2008. Learning from labeled features using generalized expectation criteria. page 595–602.

M.; El-Horbarty E.; M.Salem A. Gamal, D.; Alfonse. 2019. Analysis of machine learning algorithms for opinion mining in different domains.

Jason. 2008. 20 newsgroups.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Grégoire Mesnil, Tomas Mikolov, Marc'Aurelio Ranzato, and Y. Bengio. 2014. Ensemble of generative and discriminative techniques for sentiment analysis of movie reviews.

M Mohanapriya and Lekha Jayabalan. 2013. Fast and accurate sentiment classification using an enhanced naive bayes model. *Intelligent Data Engineering and Automated Learning – IDEAL*, 8206.

Nltk. 2020. Nltk 3.5 documentation.

N.A.; Dixit-C.; Suhag M. Pradhan, L.; Taneja. 2017. Comparison of text classifiers on news articles. int. res. j. eng. technol. (irjet). page 2513–2517.

Scikit-learn. Fetch20newsgroups.

Scikit-learn. 2020a. Countvectorizer.

Scikit-learn. 2020b. Mutual information.

Scikit-learn. 2020c. Tfidfvectorizer.

S. Sperandei. 2014. Understanding logistic regression analysis. page 12–18.

Sida Wang and Chris D. Manning. 2012. Baselines and bigrams: Simple, good sentiment and text classification.

Mais Yasen and Sara Tedmori. 2019. Movies reviews sentiment analysis and classification. *IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*.