BBC News Classification

Group 9: Hsiangwei Chao, Rui Wang, Xuexian Li, Qi Zhang

Authors

Hsiangwei Chao: first-year MS in DS
Rui Wang: : first-year MS in DS
Xuexian Li: first-year MS in DS
Qi Zhang: second-year MS in CS

Introduction

News classification is a classic text classification problem. The objective of our project is to explore some effective methods and create some classifiers to predict the types of news based on statistics methods and supervised machine learning models.

A. Objectives

We want to implement three supervised machine learning models: Naive Bayes, Softmax Regression and Support Vector Machine (SVM), then compare and evaluate their performances on the validation set. Choose the best model for this problem according to their precision, recall, time consumption and so on.

Additionally, in reality, the true training set we have sometimes is very limited in some time and we need to predict the types of a large amount of articles. So we want to try a few ideas to improve prediction accuracy under this situation. And the application set is used for the test of these methods. The mean precision is the main approach to evaluate these methods.

B. Data description

The whole dataset includes 15,963 articles crawled from BBC news website. There are five categories: 2868 articles belongs to Entertainment & Arts category, 6510 in Business , 1137 in Science & Environment , 790 in Health and 4658 in Technology. For each article, the attributes include title, content and date. Divide the data set into three parts, training set, validation set and application set. All of them are sorted in chronological order.

C. Data preprocessing

1) Word segmentation: what we get from the website at first are HTML pages. We need to extract contents, date and titles, and split these strings into separate words.

2) Remove unprintable words: the set of articles still contain unprintable words, like some punctuations and unknown characters. So we need to remove these words.

3) Remove stop words: there are many words with really high frequency, like ‘a’, ‘be’, ‘can’, ‘the’ and so on. These words are useless and make no contributions to the news classification. Those stop words will affect the calculation of TFIDF weights if they are not removed.

4) Stem Words: the final step of data preprocessing is stemming. Due to the grammar, one word may have different forms. For example, ‘dog’ and ‘dogs’ are both represent dog and ‘dog’ and ‘dogs’ are actually the same thing. However, the computer cannot understand this, so those words will be handled separately. That is why we need to stem all words. In this step, we use ‘nltk’ package to stem words.

D. Data exploration

To prove our project is feasible, we want to visualize the data to see if the dataset is separable. To reduce the computation time, apply PCA(Principal Component Analysis) first and reduce the variable space of train set to 50 dimensions.

Then use t-SNE to embed data to two-dimensional space. t-SNE is short for T-distributed stochastic neighbor embedding. This is a non-linear dimensionality reduction algorithm, which is used for embedding high-dimensional data into a space of two or

three dimensions. We will not cover much details of this algorithm because it is just a useful tool for our data visualization here.

Figure 1 is the visualization of training set in two-dimensional space using the PCA and t-SNE methods. In the figure, each color represents one label and we can see that the data are indeedseparable. This means that our objective is meaningful and the three supervised machine learning models are applicable for this news classification problem.

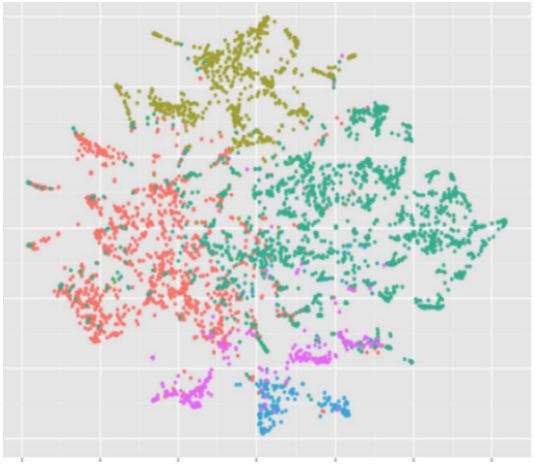


Figure one. Training set Visualization

Methods

A. TFIDF

TFIDF is short for term frequency-inverse document frequency, which is the product of term frequency and inverse document frequency. It reflects how important a word is to a document in a corpus. *D* represents the whole articles set, *d* is the document that term *t* belongs to.

$$tfidf(t,d) = tf(t,d)idf(t,D).$$

The simplest term frequency is the number of times that a term appears in a document and for our problem we will use augmented frequency.

$$tf(t,d) = 0.5 + 0.5\frac{f_{t,d}}{max\{f_{t',d}: t' \in d\}}$$

Inverse document frequency basically is logarithmically inverse fraction of the documents that contain the word. Inverse document frequency can both reduce the weight of terms that have high frequency and increase the weight of rare terms. The three models we use are based on the weights of each word calculated by TFIDF.

$$idf(t,D) = log\frac{N}{|\{d \in D: t \in d\}|}$$

B. Naive Bayes

Naïve Bayes classifiers are a family of simple probabilistic classifiers based on the Bayes' conditional probability theorem with strong (Naïve) independence assumptions between predictors. In practical applications, parameter estimation for the Naïve Bayes models uses the method of optimizing maximum likelihood. Only a small amount of training data is required to estimate the parameters for classification is one of the advantages of naïve Bayes classifier.

$$p(C_k|x) = \frac{p(C_k)p(x|C_k)}{p(x)}$$

The goal of Naïve Bayes model is to maximize the following likelihood function.

$$\hat{y} = \underset{k\in\{1,\dots,K\}}{argmax} p(C_k)\prod_{i=1}^{n} p(x_i|C_k)$$

.

C. Support Vector Machine

The Support Vector Machine classifies an observation depending on which side of a hyperplane it lies. The hyperplane is chosen to correctly separate most of the training observations into different classes. The solution of SVM is to optimize the following equations.

$$maximize\ M$$

$$\text{subject to } ||\beta||^2 = 1$$

$$y_i(\beta^T x_i + \beta_0) \geq M(1-\varepsilon)$$

Since we have larger number of predictors than number of observations, we chose to

use linear kernel since it should be much faster than non-linear kernel. One vs all scheme was used for creating the decision boundaries. For each observation to predict, the signed distance between it and each boundary, which is called confidence score, is calculated, and the category with the highest confidence score was chosen.

D. Softmax Regression (Multinomial Logistic Regression)

Softmax Regression is a kind of multinomial logistic regression classification method that generalizes binary logistic regression to multiple classification problem. we used the Softmax function in this model:

$$P(y = j|z^{(i)}) = \phi_{softmax}(z^{(i)}) = \frac{e^{z^{(i)}}}{\sum_{j=0}^{k} e^{z_k^{(i)}}}$$

The exponent exaggerates differences among $z^{(i)}$. The value would be close to one when $e^{z^{(i)}}$ is the max of all values, close to zero otherwise.

We decide to use mini batch gradient descent to compute coefficients in Softmax Regression. Mini Batch gradient descent is an optimization method to compute coefficients in order to minimize the cross-entropy-loss function. Because for the large-scale dataset in our project, it is every inefficient to use batch gradient descent, which need to go through whole training set in every iteration. Also, stochastic gradient descent is unstable and hard to converge. Compared with these two methods, mini-batch gradient descent is more effective and accurate.

E. Retrain with prediction outcome

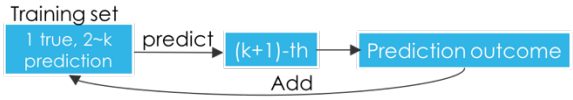
In reality, sometimes we may only know the true categories of a small proportion of data and try to predict a large amount of data or there is a long time interval between the training data and the newly generated data that we need to predict. To solve these problems, we want to try several ideas to improve the prediction accuracy under this situation.

The first thing is to partition the application set into one hundred parts and they sorted in a chronological order. For example, all the articles in the first part are from 2000 and most the articles in last part were posted in 2017. To test our methods, we assume the first part which is also the first one percent is the only true set we have and use it as our training set and consider the rest fifty-nine parts as category unknown articles.

The simplest way is just using the first one percent true data to predict the rest ninety-nine percent data.

Obviously, it is very inaccurate because there are many new words coming out every month. We need to keep updating the corpus.

The first idea we came up with to solve this problem is iteratively adding the prediction outcome to the current training set. That means we use the first part as training set to predict the categories in the second part and add the prediction outcome to the original training set, then use the new training set to predict the third part and so on. In this way not only can the corpus keep updating but also reinforce the original true data. The model may perform worse eventually due to the accumulation of errors but it can still be able to perform well in first several phases. There is another problem that this method is time-consuming since the size of training set would grow linearly.



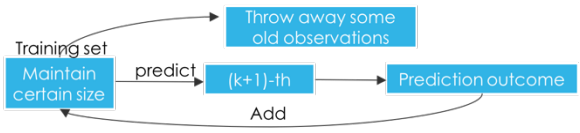
F. Reservoir Sampling

It is natural to come up with another idea that we can throw away some old observations or old words so that we can maintain the training set to certain size. This way can highly improve the computation efficiency and it may not affect the accuracy

a lot as well. We decided to use Reservoir Sampling to implement this idea.

Reservoir Sampling is a randomized algorithm that when we need to randomly pick k items from a very large list and we do not know the length of the list in advance. Specifically, when the i-th ($i > k$) item arrives, keep the new item and replace an old one randomly with a probability of $\frac{1}{i}$ and keep all the old items with $1 - \frac{k}{i}$.

The reason why we use the Reservoir Sampling to implement our idea is its time and space efficiency. For example, in i-th iteration, we need to add the $(i+1)$-th prediction outcome to the current training set. The training set is supposed to be randomly picking certain number of items from all previous data including the true data or previous prediction outcome and the cur-rent prediction outcome. But what we have at this stage is the current training set and some old observations have been removed. This method can avoid going through the old ones again and still make sure each element is picked with equal probability. Also, we don't need to know how many more articles we need to predict, which means we can stop at any time. In a word, what we do is that in each iteration we throw away some old observations and add new data with equal probability with Reservoir Sampling method to maintain the size of training set.



Results

The whole dataset was divided into three parts: 30% training set, 20% validation set and 50% application set. Both training set and validation set were randomly shuffled, while application set was sorted by chronological order. The best model is selected from the validation set, and is used to be applied to the application set. Table 1 and Table 2 display the number and proportion of each categories in training set and validation set.

Training set		
Category	**Count**	**Proportion**
Technology	1421	29.7%
Entertainment & Arts	872	18.2%
Business	1941	40.5%
Health	227	4.7%
Science& Environment	327	6.8%

Table 1. Number and proportion of articles in training set.

Validation set		
Category	**Count**	**Proportion**
Technology	960	30.1%
Entertainment & Arts	591	18.5%
Business	1262	39.5%
Health	171	5.4%
Science& Environment	209	6.6%

Table 2. Number and proportion of articles in validation set.

A. Performance on validation set

We evaluated the performance of three different models on validation set using their precision, recall and f1-score. Since the data is imbalanced, and categories with smaller proportion should not be omitted, we considered both micro average and macro average.

The difference between micro and macro average is that micro average gives equal weights on each article, while macro average gives equal weights on each category. A model with high micro average and low macro average performs bad on categories with small proportion. A good model in text classification should perform well on both averages.

		Precision	Recall	F1-score
Naïve Bayes	Mirco Avg	0.861	0.848	0.836
	Macro Avg	0.907	0.698	0.741
		Precision	**Recall**	**F1-score**
SVM	Mirco Avg	0.934	0.932	0.932
	Macro Avg	0.930	0.927	0.931
		Precision	**Recall**	**F1-score**
Softmax	Mirco Avg	0.932	0.931	0.931
	Macro Avg	0.930	0.933	0.932

Table 3. micro and macro average

B. Coefficients Interpretation

Besides performance, it is also essential to interpret the results of these three models. Actually, their coefficients can reflect the importance or weight of each word to each category. Words with higher coefficients are more important predictors for predicting the category of an article. Table four, five, six, seven and eight show the words with top 5 highest coefficients for each category picked by three different models, Naïve Bayes, SVM and Softmax regression. Keep in mind that these words are preprocessed with stemming.

Technology			
No.	**Naive**	**SVM**	**Softmax**
1	user	googl	firm
2	firm	site	googl
3	googl	game	site
4	compani	devic	devic
5	devic	firm	game

Table 4

Business			
No.	**Naive**	**SVM**	**Softmax**
1	bank	busi	busi
2	compani	bank	bank
3	busi	compani	compani
4	market	tradit	mr
5	growth	mr	tradit

Table 5

Science&Environment			
No.	**Naive**	**SVM**	**Softmax**
1	space	scienc	climat
2	scientist	rocket	scienc
3	earth	climat	rocket
4	climat	speci	engin
5	scienc	engin	scientist

Table 6

Health			
No.	**Naive**	**SVM**	**Softmax**
1	patient	nh	nh
2	nh	care	patient
3	health	health	care
4	hospit	cancer	health
5	cancer	patient	hospit

Table 7

Entertainment & Arts			
No.	**Naive**	**SVM**	**Softmax**
1	patient	nh	nh
2	nh	care	patient
3	health	health	care
4	hospit	cancer	health
5	cancer	patient	hospit

Table 8

C. Reservoir Sampling Performance

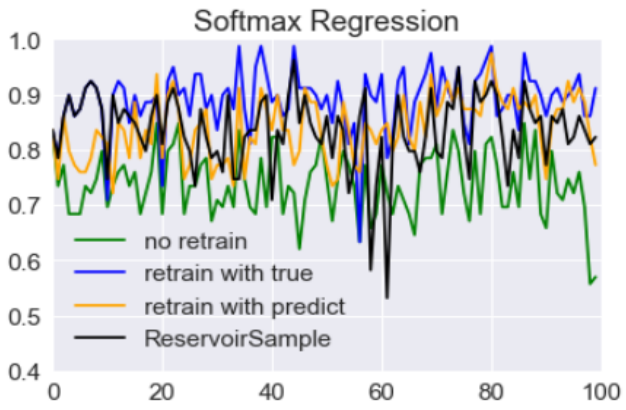


Figure 2. Precision

According the performances on validation set of three models, we decided to use Softmax Regression to try out our ideas.

In the figure two, X-axis is the number of part of data and Y-axis is the precision starting from 4%. The blue line represents the performance of the most ideal case that we always retrain with first $(i-1)$-th parts of true data to predict the current i-th part. The green line is the performance of only using the first 1% true data without retraining. These two lines are for comparison. The yellow line is the performance if we always retrain the model with prediction outcome. The black line is the performance of the model after using Reservoir Sampling to maintain the size of training set. Table 9 shows the mean precision of each line.

Methods	Retrain with true data	Reservoir Sampling	Retrain with prediction	No retrain
Precision	0.907	0.845	0.801	0.725

Table 9

Figure 3 shows the comparison of execution time when the method with and without Reservoir Sampling.

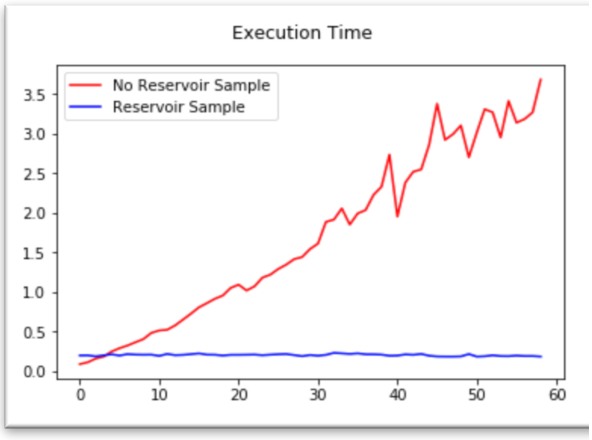


Figure 3. Execution Time

Discussion

A. Performance

Naïve Bayes performs fine if consider only micro averages. Its macro average for recall, however, is substantially low. This might be because Naïve Bayes model largely favors categories with larger proportions. A low macro average disables the possibility to use prediction to predict, since eventually it will predict only categories with larger proportion.

Both SVM and Softmax regression have similar effectiveness. Not only do they perform well on precision and recall, but also on micro average and macro average. These are positive signs for a good model in text classification.

B. Coefficients

At first, we tried to apply PCA for variable space reduction to save the time of training models. However, it turned out the precision is lower if we use the training set processed with PCA. Additionally, if we want to use PCA, we need to do SVD decomposition to choose the appropriate number of principle components, which is very time-consuming. So we decided not to use PCA. Also, when implementing the Softmax Regression, we found out lasso did not improve the accuracy on validation set, which means the model does not have overfitting problem. That was partly due to removal of low frequency words during TFIDF.

Since we did not do dimension reduction for Naïve Bayes, SVM and Softmax Regression, it is much simpler to interpret the fitted models by observing the words with highest coefficients for each category. Words with higher coefficients are more important predictors for predicting the category of an article. Words with highest scores in SVM and Softmax regression are highly identical. They have almost the same words with only slightly differences in order. We may conclude that SVM and Softmax regression

had resulted in very similar models. This might also explain why they have similar precision and recall.

C. Model Selection

The performance of Naïve Bayes model is much worse than those of Support Vector Machine and Softmax Regression, which means we should choose the best model between SVM and Softmax Regression. These two models, however, not only have similar effectiveness, but also have similar models while analyzing words with higher coefficient. But there is a slight difference between their macro average recall on the validation set. More importantly, since we used Mini Batch Gradient to train the Softmax Regression, Softmax Regression has the advantage of time efficiency over SVM. Hence, we are supposed to select Softmax Regression as the best model for this dataset. And it would be used in the study of retraining with prediction outcome.

D. Reservoir Sampling

From the figure 2, we can see that Retrain with true data(blue line), not surprisingly, gives the best prediction result. Prediction without retraining(green line) doubtless has the worst precision among all. We use this two lines for comparison.

If we keep retraining the model with the new prediction outcome(yellow line), the whole accuracy is improved. Not only because does the prediction reinforces the original limited true data, but also the corpus keeps updating. But it gets worse eventually because of the accumulation of the errors.

Also, we found that not only Reservoir Sampling(black line) improves the accuracy but also the line does not have strong trend to decline in the end. This is possibly due to some faulty prediction are abandoned. What's more, from Figure 3, we can see that since it always have constant training set, it largely decreases the execution time for training the model. In a word, our method that retrain with prediction outcome and maintain the size of training set can improve the mean accuracy.

Therefore, we concluded that this is a possibly effective approach to improve the prediction performance under the situation that the true data is very limited.

References

McCallum, A., & Nigam, K. (1998, July). A comparison of event models for naive bayes text classification. In AAAI98 workshop on learning for text categorization (Vol. 752, pp. 4148)

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. Machine learning: ECML98, 137142.

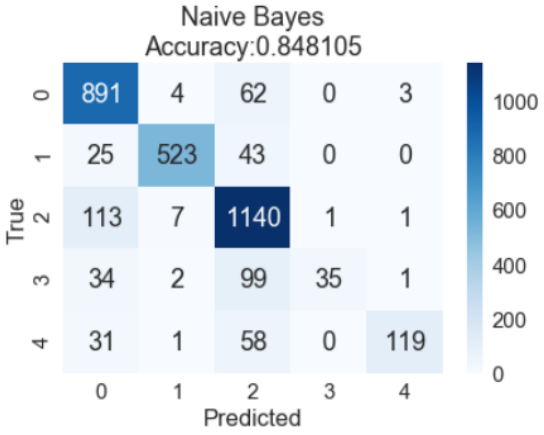
Trevor, H., Robert, T., & Jerome F., (2008). The Elements of Statistical Learning 2nd Edition,Springer.

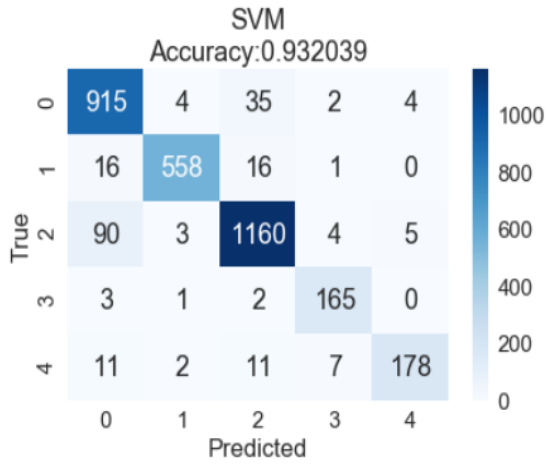
Appendix

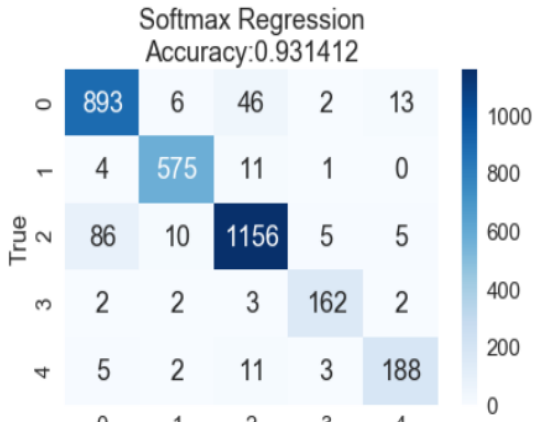
Github (code):

https://github.com/chao-h/DS5220

Confusion Matrices:







Statement of Contributions:

Qi Zhang: Model selection, model evaluation

Xuexian Li: Data Prepressing, model evaluation

Hsiangwei Chao: Naïve Bayes and SVM implementation, Retrain model

Rui Wang: Softmax Regression implementation, Reservoir Sampling