Movie genres classification
COMP90049 Report

Anonymous

1 Introduction

Genre classification and sentiment analysis are the two most mainstream application scenarios of movie classification technology. The sentiment analysis focus on movie review comments to get objective audiences' attitudes. Genre classification technology is used not only to judge the movie ratings to find the suitable age group but also to define movie categories, such as action, comedy, horror and so on. It's a very challenging classification problem that labelling movies according to their corresponding genre.

In this paper, the research process can be mainly divided into two steps: i) feature selection and representation, ii) model training and prediction. The former relies on term frequency–inverse document frequency (TF-IDF), while the latter bases on the Naive Bayes and the Support Vector Machine (SVM). The re- port shows the changes in the prediction results under different features or different models and further analyzes these changes.

2 Related Work

Over the years, researchers have proposed various automatic genre classification methods for text and general video data. Some methods rely on textual features, such as movie titles or summary user tags. Other methods rely on audio- visual features.

Rasheed et al(2010). compute four video features: lighting key, motion content, average shot length and colour variance to predict movie gen- res. But they only divided movies into four categories: comedy, action, drama, and horror. Obviously, this broad classification method is no longer applicable. In addition, since certain categories are very similar visually such as documentary and comedy, using just visual features are not enough. In this scenario, the audio or text extracted from audio is necessary for genre classification. One hybrid method was proposed by Huang and Wang(2017) combining both low-level visual features and audio information.

Huang(2017) applied three methods to predict 10 most popular genres based on the dataset only containing 16,000 movie titles. Overall, the SVM achieves the highest F1 score of 0.55 in his paper. Lei(2009) explores Naive Bayes and Recurrent Neural Networks for text classification, and his model performs well in multi-label problems.

3 Feature Selection

3.1 Data Set

The data set used in this report is derived from a larger database script by Deldjoo et al(2018). and F. Maxwell et al (2015).

The data set is partitioned into three sets, a training set, a valid set and a test set. The training set and the valid set contain two files, the features data and the genres data. The training set is used to train and fit the model, and the valid set is used to test and analyze the differences between various models. The test set has only one file containing features data. We need to use a trained model to predict the genres of the test data. The prediction result is used for kaggle competition. Table 1 below shows the structure information of different files.

Corpus	Features	
Training set	Train_features	5240 instances
	Train_lables	5240 instances
Vaild set	Valid_features	299 instances
	Value_labels	299 instances
Test set	Test_features	235 instances

Table 1: The structure of the data

3.2 Data Pre-processing

For the special model, the input data should have the same structure. Therefore, we should perform the same operation on train

features, valid features and test features. Below I will only explain the processing of train features.

Preprocessing the original data is necessary and effective to promote the training of the model, because there are NaN and invalid data in train features. Abnormal data can be divided into three categories:

Since MovieId and YTId do not contain any information related to the movie category, eliminating them from the training set does not affect the accuracy of the model

The year represents the release time of the movie. But the data of year in some instances is missed. Compared with title and tags, the year contains very little movie information, although movies of similar types may be released at the same time. I choose to exclude the year column.

In all instances, the values of avf31, avf32 and avf104 are the same. I have reliable rea- sons to judge that these three data do not rep- resent any characteristics of the movies. There- fore, I can eliminate these three columns.

3.3 Term Frequency–Inverse Document Frequency

Since the ”title” and ”tags” are textual data, We cannot directly know which word is important. Term Frequency–Inverse Document Frequency (TF-IDF) is a numerical statistic that is intended to reflect how important a word is to a document in the corpus.

Calculate the TF at first. The weight of a term that occurs in a document is simply proportional to the term frequency. In other words, in one instance, the more times a word appears, the more important the word. Therefore we can define:

$$tf(t,d) = f_{t,d}$$

$f_{t,d}$: the number of times that term t occurs in document d

Then, we need calculate the IDF. The specificity of a term can be quantified as an inverse function of the number of documents in which it occurs. In other words, A word is not important if it appears in most examples. So we can define:

$$idf(t,D) = \log(D/n_t)$$

D: the number of instances

N_t: the number of instances containing t.

Finally, we can get the tf-idf:

$$\text{Tf-idf} = tf(t, d) * idf(t, D)$$

The greater the tf-idf, the more important the word.

3.4 Standardization or Normalization

Comparing different columns, we can find that the values between the columns are very different. For example, *avf1* is generally less than 0.5, but a*vf15* is generally greater than 10. If we directly train this data, when *avf1* and *avf15* have the same coefficient, the impact of *avf15* will be much larger than that of *avf1*. However, the data of different columns should have the same weight. Therefore, we need to standardize or normalize data.

By calculating the value of the Gaussian error function for each column, we can find that most data does not conform to the normal distribution. For example, *avf10* has only four discrete values. If we use standardization, the error of the training result may increase. There- fore, I use normalization to process the data, and the range of each column is [0,1]:

$$X' = (X - X_{min})/(X_{max} - X_{min})$$

4 Methodology

Because this is a classification problem, in a large number of machine learning models, we should first consider the classification algorithm, such as Naive Bayes, SVM, K-nearest neighbours and Decision tree. I

verified these four models in the code and found that Naive Bayes and SVM performed better. This article will focus on the in-depth analysis of Naive Bayes and SVM.

4.1 Naive Bayes

The naive Bayes method is based on the conditional independence assumption with the decision rule, *maximumaposteriori* (MAP), to classify an instance. It is a mainstream spam classification algorithm and has great advantages in processing textual data. In the original data, both the title and the lag are textual data, and naive Bayes may have an ideal training effect.

Naive Bayes can be roughly divided into three categories: Gaussian naive Bayes, Multinomial naive Bayes, and Bernoulli naive Bayes. When dealing with continuous data, one common assumption is the data is distributed according to a normal distribution. Therefore, we cannot calculate the exact probability of a giving point since the data is not discrete. However, we can get the mean and variance of the sample

to calculate the density function of normal distribution. Using the density function to replace the probability is the kernel of Gaussian naive Bayes.

When the data is discrete and finite, samples (feature vectors) represent the frequencies with which certain events have been generated by a multinomial. Multinomial naive Bayes is an event model commonly used for document classification, where events represent the occurrence of words in a single document.

Like the multinomial model, the Bernoulli model is also suitable for discrete features. The difference is that the value of each element in the Bernoulli model can only be 1 and 0. If we only need to judge whether the keyword appears, Bernoulli naive Bayes is the best model.

4.2 Support Vector Machine (SVM)

SVM is supervised learning models with associated learning algorithms. This method con- structs a maximum-margin hyperplane which aims to lower the generalisation error of the classification. It can be used for both regression and classification tasks. Usually, it is widely used in classification labels.

With the help of kernel function, SVM can handle both linear and non-linear data classification. The most important theory is mapping the feature vectors in low-dimensional space to high-dimensional space where we can easy to find linear separability. Besides, when we ap- ply the kernel trick to maximum-margin hyperplanes, we can get a nonlinear classifier with curved borders.

5 Results

5.1 Different TFIDF parameters

The Multinomial naive Bayes model is used here to determine the effect of different TF-IDF parameters on the accuracy of the model. Above all, stop words have no meaning, so we use *stop words =′ english′* to eliminate all stop words. Besides, there are two important parameters: *min_df* and *max_df.*

min_df is a cut-off parameter since every word in word bag must be contained in at least min df instances in the whole data set. If float in the range of *[0.0, 1.0]*, the parameter represents a proportion of documents. It can effectively avoid the impact of extreme data that has only appeared once or twice on the model. *max_df* have the opposite effect that can delete the common words.

There are more than 5,000 data in the *title* words bag. Without filtering, the difficulty of model training will greatly increase. At the same time, too many features may cause the curse of dimensionality. However, the *tag* words bag only have 206 words. It's easy to handle hundreds of features. Therefore, we can use max df = 0.3 to delete the most common words in *title* and *tag*.

max_df. is 0.3, *min_df.* expands from 1 to 9, the accuracy rate increases from 24% to

38%. The word bag size is reduced to 149 from 5819. When *min_df* expands from 9 to 20, the accuracy rate drop to 36%. Since then, continue to expand the *min_df*, the accuracy rate has been maintained at around 36%.

we can have a conclusion that when min df is very small, the samples in the training set are very large, and our model cannot effectively learn the information from data. As the training set decreases, the accuracy rate increases steadily. However, when min df is greater than 9, the training sample becomes smaller, and the accuracy rate begins to gradually decline, indicating that the features cannot effectively reflect the information of *title.* When min df is greater than 20, the data extracted from *title* has become insignificant and has no significant effect on the model. We can make bold predictions, 36% accuracy rate is the model ability to predict without *title* columns.

5.2 Different Naive Bayes models

Now we stipulate *min_df = 9* and *max_df = 0.3*. Let us observe the effect of different Bayes models on accuracy. Before testing, I predict that the Multinomial Naive Bayes will perform best, while the Gaussian Naive Bayes and Bernoulli Naive Bayes are very poor because most of our features are discrete data and do not satisfy the binomial distribution or Gaussian distribution.

However, the accuracy of Multinomial Naive Bayes is 38.4%, the accuracy of Bernoulli Naive Bayes is 31.6% and the accuracy of Gaussian Naive Bayes is 7.6%.

Why is Bernoulli Naive Bayes much better than Gaussian Naive Bayes in our project? I analyzed the training data. Many raw data are randomly distributed. However, the data extracted by tfidf about *title* and *label* usually only have two or three specific values. These data cannot fully satisfy the binomial distribution, but the data characteristics are very similar to the binomial distribution. When processing such data, Bernoulli Naive Bayes realizes much better than Gaussian Naive Bayes.

5.3 Linear or Nonlinear SVM

Now, I chose one popular nonlinear SVM, Ra- dial basis function (RBF), to compare with sample linear SVM. The linear and RBF kernel are simply different in case of making the hyperplane decision boundary between the classes. Usually, RBF can map the feature vectors in low-dimensional space to higher high- dimensional space than linear SVM.

The accuracy of Linear SVM is 39.2%, but the accuracy of RBF SVM is 41.4%. In my opinion, since RBF mapping the vectors to very high-dimensional, it's easiler to find separability than linear SVM. Therefore, it is more accurate. However, my computer has been running RBF SVM for more than 1 minute, and running B only took less than 15 seconds. Finally, the mapping process from low size to high size takes a lot of time. And because the RBF SVM data is more sparse, the risk of overfitting a single training set will increase.

5.4 Naive Bayes vs SVM

In this part, we just compare the Linear SVM with Multinomial Naive Bayes. The prediction accuracy of the two models is very close, and Linear SVM is only 0.8% higher than Multinomial Naive Bayes.

I mentioned above Naive Bayes is a mainstream spam classification algorithm. This model usually performs well on textual data. However, SVM can handle both textual and numerical data. In our raw data, only a few columns are text, while more than 100 columns are numbers. Maybe it's the reason why SVM is better than Naive Bayes in this problem.

6 Conclusions

To summarize, this report evaluates the effectiveness of Naive Bayes and SVM on the classification of film genres. Based solely on the given data set, the overall accuracy of SVM is better than Naive Bayes. In addition, it is very important to modify the appropriate parameters for every model.

Nowadays, more and more movies have more than one genre, and multi- label movie genre detection is the future improvement direction.

References

Deldjoo, Yashar and Constantin, Mihai Gabriel and Schedl, Markus and Ionescu, Bogdan and Cremonesi, Paolo. MMTF-14K: A Multifaceted Movie Trailer Feature Dataset for Recommenda- tion and Retrieval. Proceedings of the 9th ACM Multimedia Systems Conference, MMSys 2018, Amsterdam, The Netherlands, June 12-15, 2018

F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19 (December 2015)

Tun Thura Thet, Rasheed, Jin-Cheon Na, and Christopher SG Khoo. Aspect-based sentiment analysis of movie reviews on discussion boards. Journal of information science, 36(6):823–848, 2010.

Huang Wehrmann and Wang C. Barros. 2017. Movie Genre Classification: A Multi-label Approach based on Convolutions Through Time. Applied Soft Computing 61 (2017),973–982. https: //doi.org/ 10.1016/j .asoc.2017.08.029

Lei Tang, Suju Rajan, and Vijay K. Narayanan. 2009. Large Scale Multi-label Classification via Metalabeler. In Proceedings of the 18th International Conference on World Wide Web (WWW '09). ACM, New York, NY, USA, 211–220. https: //doi.org/10.1145/1526709.1526738