Author's gender identification by classify twitter text

ZHILONG ZHENG (29499496)
SONGYUAN LI (29439205)
XIUCONG XIE (29778115)

JUNE 14, 2020

Abstract

In the age of big data, it is found that we can identify the personality features of specific groups by studying people's language. Here, we created three models to identify the author's gender by classifying Twitter text which post by author, which are SVM, KNN and logistic regression. In this report, we use word frequency method to extract features from twitter text and find the SVM model is the best model to predict the author's gender. We hope to explore the relationship between People's language and features through machine learning.

Keywords: Gender identification; Classifier; SVM; Feature; Twitter

Table of Contents

1 Introduction 3

2 Pre-processing and feature generation 3

2.1 Pre-processing 3

2.2 Feature generation 4

3. Models 4

3.1 Logistic regression model 4

3.2 Support Vector Machines (SVM) 5

3.3 k-nearest neighbors (KNN) 5

3.4 Discussion of model difference(s) 5

4 Experiment setups 6

5 Experimental results 6

5.1 optimal extraction features 6

5.2 optimal model parameters 7

6. Conclusion 8

References 8

1 Introduction

Authorship analysis is to determine some features by studying the language and text used by specific types of people, then using these features to identify the author's personality, gender, age and language etc (Author Profiling 2017, n.d.). In this project, it is required to extract features related to data by using twitter content posted by authors, and to create three different classifiers with machine learning technology to solve the problem of author gender recognition. Specifically, the data included the id of 3,600 authors and 100 tweets from each author. Of those 3,600 authors, the gender of 3,100 authors was known, while the gender of the remaining 500 authors was unknown. We need to extract features from the twitter context of 3,600 authors and select classifiers to train the model based on 3,100 author data of known gender, so as to predict the gender of other 500 authors.

At the same time, in this project, our goal is to select the appropriate feature, then adjust the parameters of the classifier and choose the best model, so that the prediction results of the classifier can be highly similar to the correct results, and the accuracy is expected to remain at around 80%.

In this program, all members of our group participated as part of data preparation and feature extraction, and then each member was required to find a classifier method in machine learning algorithm, to be involved in adjusting models, data prediction and accuracy evaluation. Finally, finish this report together.

2 Pre-processing and feature generation

The data set has a labelled file about authors and the information of their gender. And it contains many twitter texts which are collected from social media, so the structure of these data is not suitable for us to analyse directly. It is required for us to do some data pre-processing. After that, we will generate the feature based on extracted texts.

2.1 Pre-processing

The function of ElementTree is a convenient method to deal with xml files. First, we use this method to load each file and get roots in it. In our data, each file has one root <author> and the root has a node named <documents>. The text that we want to extract is in subnodes named <document>. Finally, we put the extracted data in an empty dictionary. A sample of the dictionary is:

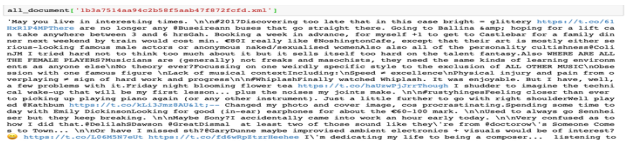


Figure 1. A sample of extracted data

We can see from figure 1 that there are some meaningless words for generating our feature, like the web link and some usernames.

For web links, we prefer to change all links to a word ‘URL’ rather than delete them. Because one web link represents a picture and putting pictures in twitter text is a habit, which is useful for us to generate features.

For usernames, we would like to use the word ‘username’ to replace all real names. The number of ‘username’ can show how many times the author interacts with others.
After filtering the text, we put results in a dataframe, like:

```
df_without_feature.sample(5)
```

	id	content	feature
2290	991f5a4c639813c58764ede843f9b9f7	Eased down winner 2.5l. 💰💰💰 URL The voodoo mag...	NaN
1565	af6e2052d66eba20031298039 8abf45	Highlighter for scale 😄 URL not including the ...	NaN
2446	73873ae2fbf82446d90115b8d902fbea	Whitney Houston - How Will I Know (Beave Remix...	NaN
3451	c222c5b205fbb531e78c07b9ba9971ab	@username I look forward to seeing your new sk...	NaN
826	94bdb7d46163215d094666ce3c75ca33	What a birthday! Would have been nice take the...	NaN

Figure 2. Dataset after pre-processing

2.2 Feature generation

Then we tokenize the sentences by using LemmaTokenizer and vectorize the text.
There are lots of capitalized words so convert all characters to lowercase before tokenizing.
Frequency control: Set 'min_df=0.05' and ‘max_df=0.95’ is in order to filter out words with low and high frequency.
Stopwords: Remove the stopwords in package ‘english’.
N-grams: We set the range of n-grams from 1 to 3. Because the length of phrases is mostly less than 3 words.
Then we put the result in the previous dataframe and merge this dataframe with the training set to get information about gender for each ID.
The final training set is:

	id	gender	content	feature
329	f018af7405a2be33c56a2d9d9c056c59	1	Fall in #love today. And tomorrow. And every d...	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
1862	22d4568344e014a64baa40f33f0bec31	0	had the worst chicken nuggets. mcdonalds wtf i...	[0.635564772326453, 0.4395131418311368, 0.2364...
3035	ff26ace44cfced9f9b329afc23aaeda6	0	Just commented on @username Ireland is expecte...	[0.005916611760677986, 0.0, 0.0, 0.0, 0.0, 0.0...
698	a8c233deb8093cf3d8dfd45e017b51d2	1	Dear @username have you used getdns to send LO...	[0.02255693985692344, 0.0, 0.0, 0.0, 0.0, 0.0,...
874	b7a354ce032c38376715f0c8361d5b98	1	Who's geeky enough to go to this with me 😂😎🤹 U...	[0.29395037414040637, 0.039459455103892344, 0....

Figure 3. Sample of training set

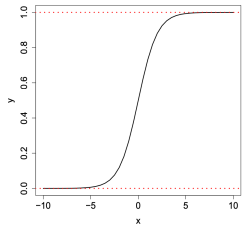
3. Models

3.1 Logistic regression model

Logistic regression is a machine learning method used to solve binary classification (0 or 1) problems, used to estimate the probability of a certain thing. For example, the possibility of a user purchasing a product, the possibility of a patient suffering from a certain disease, and the possibility of an advertisement being clicked by the user (Swaminathan, 2018).
Logistic regression model is based on the logistic function. The formula of logistic function is

$$f(x) = \frac{e^x}{1+e^x} = \frac{1}{1+e^{-x}}$$

logistic regression formula



logistic regression figure

The method of logistic regression model is to calculate x in the formula through feature, weight and intercept, and calculate the probability of each feature through the formula. This probability is the probability that a feature belongs to one of the labels. These probabilities can be used to deduce which label each feature should belong to (Brownlee, 2019).

3.2 Support Vector Machines (SVM)

A support vector machine (SVM) is a supervised machine learning model that uses classification algorithms for two-group classification problems. After giving an SVM model sets of labeled training data for each category, they're able to categorize new text.
A support vector machine takes these data points and outputs the hyperplane (which in two dimensions it's simply a line) that best separates the tags. This line is the decision boundary: anything that falls to one side of it we will classify as blue, and anything that falls to the other as red (Stecanella, 2017). The algorithm will find the best hyperplane that maximizes the margins from both tags with the formula of calculating margins (d):

$$d = \frac{|\omega^T x + \gamma|}{\|\omega\|}$$

3.3 k-nearest neighbors (KNN)

K Nearest Neighbour is an algorithm that stores all the available cases and classifies the new data or case based on a similarity measure. It is mostly used to classify a data point based on how its neighbours are classified (Subramanian, 2019).
In the classification setting, the K-nearest neighbor algorithm essentially boils down to forming a majority vote between the K most similar instances to a given "unseen" observation. Similarity is defined according to a distance metric between two data points. The distance is calculated by Euclidean distance method:

$$d = \sqrt{\sum_{i=0}^{n} (xA_i - xB_i)^2}$$

3.4 Discussion of model difference(s)

Difference between these three models:
KNN is a non -parametric model, whereas Logistic regression and support vector machines is a parametric model. SVM and KNN can handle non-linear solutions whereas logistic regression can only handle linear solutions (Varghese, 2018).
Advantages and disadvantages of each model:

- Support vector machines

1. Advantages: SVM handles outliers better than KNN and Logistic regression and outperforms KNN when there are large features and lesser training data.
2. It is difficult to train large-scale data and sensitive to missing data.

- Logistic regression

1. Advantages: θ parameter explains the direction and intensity of significance of independent variables over the dependent variable. It also can be used for multiclass classifications.
2. It cannot be applied on non-linear classification problems. Proper selection of features and good signal to noise ratio is required.

- K-nearest neighbors

1. Advantages: It is an easy and simple machine learning model and has few hyperparameters to tune.
2. Disadvantages: K should be wisely selected and Large computation cost during runtime if sample size is large.

4 Experiment setups

For the model, we used the SVM model, logistic regression model and KNN model.

For the SVM model, we used the SVM in the Sklearn library. For this model, it has many parameters. We adjusted some of the parameters to get an optimal model.
The parameters are

- C is the Regularization parameter.
- kernel is the type of kernel used by this algorithm.
- Gamma, the value of gamma we set is auto.

For the logistic regression model, we used the LogisticRegression function to create the logistic model. We modified C, max_iter and fit_intercep in the logistic model model.
These parameters represent

- C variable means Inverse of regularization strength
- max_iter means the maximum iteration times of logistic model
- fit_intercept means whether to add intercept to the model.

For the knn model, we used the KNeighborsClassifier function to create the model. We modified n_neighbors, this parameter means number of neighbors to use.

We used the cross-validate setup method, and for this method I used the function GridSearchCV. This method divides the train dataset into 5 parts and takes one of them as test data. The rest is used as a training dataset to train the dataset. We set a variable called parameters. This variable stores all the parameters in the model that need to be changed. A model is built by combining these parameters randomly, and a set of parameters with a higher accuracy is selected.
The accuracy formula is

$$Accuracy = \frac{number\ of\ correct\ predictions}{number\ of\ all\ predictions}$$

5 Experimental results

5.1 optimal extraction features

The first experiment is optimal extraction of features to improve model accuracy.
The feature we use is the word frequency feature. In the method of extracting the feature, we use the two functions CountVectorizer and TfidfTransformer to convert the data, which will convert all words into numbers in the form of word frequency. Because we have two methods for processing data before, we convert the results obtained by the two methods into features according to the word frequency method and bring them into the same model to calculate the accuracy. We used svm model, knn model and logistic model, and the parameters of all models are default values. We will call the method of converting only url and @username as Method 1. On this basis, we will continue to refer to the processing method of converting words such as “$amp”， “&gt” , “&lt” and etc as Method 2. The accuracy rate is shown in the following two tables.

Method 1		Method 2	
model	**Accuracy**	**model**	**Accuracy**
SVM	0.804	SVM	0.788
KNN	0.65	KNN	0.64
Logistic regression	0.806	Logistic regression	0.8

table 1. The accuracy of different methods and models

Through these tables, we can know that the first method has a high accuracy rate for the model, but the parameters of these models are the initial values.

5.2 optimal model parameters

The second experiment we consider optimal model parameters to improve model accuracy. We adjust the parameters of the model of each method. We use the cross-validation method and use the GridSearchCV function. Through this method we can calculate the results of all models after optimization.

Method 1		Method 2	
model	**Accuracy**	model	**Accuracy**
SVM	0.81	**SVM**	0.798
KNN	0.7	**KNN**	0.682
Logistic regression	0.806	**Logistic regression**	0.802

table 2. The accuracy of different methods and models

From these accuracy analysis, method 1 SVM will get higher accuracy, this is because when the first method is used for processing, the total number of data features will be more than method 2, because in method 1 A lot of special symbols will be reserved, so that the number of features in method 1 will be more than that in method 2, which leads to the accuracy of the SVM model in method 1 is greater than the logistic model, the accuracy of the logistic model in method 2 It is larger than the SVM model, because SVM is more suitable for data sets with a large number of features.

6. Conclusion

In a nutshell, in our experiment, the SVM model was the best classifier to identify the gender of a tweet writer as accurately as possible, with an accuracy of 81%.

In the process of this project, we believe that there are two key points to improve the accuracy of the prediction model. The first is the extraction method of features. The feature we used is the word frequency feature, in which we used two functions, CountVectorizer and TfidfTransformer, as well as the selective use of some text transformations. The second point is the selection of models and the adjustment of parameters in models. SVM model, KNN Model and Logistic Regression Model are considered, and the GridSearchCV Function is used to adjust their parameters respectively.

Through the exploration and learning of this project, we get the idea of how to classify features by language, have a further understanding of the algorithm of machine learning, and realize the difficulty of parameter adjustment. If it is allowed in the future, we hope to make more efforts to find a more suitable model, so that the accuracy can reach 90%.

References

Author Profiling 2017. (n.d.). Retrieved from https://pan.webis.de/clef17/pan17-web/author-profiling.html

Brownlee, J. (2019, August 12). *Logistic Regression for Machine Learning*. Retrieved June 2020, from https://machinelearningmastery.com/logistic-regression-for-machine-learning/

Stecanella, B. (2017, June 22). *An Introduction to Support Vector Machines (SVM)*. Retrieved June 2020, from https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/

Subramanian, D. (2019, June 8). *A Simple Introduction to K-Nearest Neighbors Algorithm*. Retrieved June 2020, from Towards data science: https://towardsdatascience.com/a-simple-introduction-to-k-nearest-neighbors-algorithm-b3519ed98e

Swaminathan, S. (2018, March 15). *Logistic Regression — Detailed Overview*. Retrieved June 2020, from Towards data science: https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc

Varghese, D. (2018, December 7). *Comparative Study on Classic Machine learning Algorithms*. Retrieved June 2020, from Towards data science: https://towardsdatascience.com/comparative-study-on-classic-machine-learning-algorithms-24f9ff6ab222