COM90049 Project 2 Report

Name: Bing Xie
StudentID: 741012

1 Introduction

The purpose of this project is to train a system to make a prediction on geolocation for Twitter user based on their tweets which have been labeled. In this report, feature engineering will be introduced and conducted to find proper attributes based on methods, process and observation. Various machine learning algorithms will be applied to classify the geolocation. Finally machine learning algorithms will be evaluated on the basis of output performance report of weka.

2 Feature Engineering

The purpose of feature engineering is to effectively choose attributes with good performance. In this part, three methods are used, namely find feature by statistical words frequency after processing Twitter dataset, manual find feature associated with the geolocations and further feature selection

2.1 Find Feature By Statistical Words Frequency

2.1.1 Remove duplicate instance, case sensitive and non-alphabetic characters

It can bee seen that there exist numbers of duplicate twitter contend which can be regard as redundant instance. Hashset are used to eliminate all duplicate instances. Use regular expression to identify all-alphabetic characters and replace non-alphabetic characters with space line by line.

2.1.2 Stemming

The morphological variants of words may share the same meaning and those words can be considered as same word. For example, in the twitter dataset, care and careful can be considered as same word. With stemming, attribute datasets reduced. A good prediction model will be learned and constructed by weka with minimum attribute datasets.
Lovins stemmer is used to remove the words affixes because of its algorithm with good time complexity. Apart from removing double letters, irregular plurals can be also handled, such as mice.
After applying Lovins stemmer, short and meaningless words like ux, sd and dc can be merged. In this case, attributes like these could be removed from candidate attributes.

2.1.3 Removing stop words

Since words like the, a and already hold no meaning for geolocation prediction, words like these should be removed from instances and thus the efficiency of prediction model and geolocation prediction may be improved.

2.1.4 Removing user names and IDs

Majority of words begin with @ are username, which hold no meaning for constructing geolocation prediction model and should be removed. ID and punctuation similar to username also require to be removed.

2.1.5 Count the words occurrence frequency

In my java program, AttributeSelect class is used to count the word occurrence frequency in twitter dataset. However, count word occurrence frequency is unnecessary since Nave Bayes is a binary classifier. The purpose is to choose proper feature candidates for further selection.

2.2 Manually Find Features

Features are found based on conducting an investigation on Houston, Boston, San Diego, Seattle and Washington DC. This investigation is mainly focus on local history, climate, culture, landmark architecture and famous university. For example, the White House is the residence of president as a famous landmark of Washington DC. Then check these two words in twitter datasets and it can be found that they have a

high word frequency with respect to Washington DC.

2.3 Further Feature Selection

According to lecture, the basic feature selection methods are point wise mutual information (PMI), mutual information (MI) and Chi-square (χ^2).

2.3.1 Mutual Information versus Chi-square

According to reference as shown in table 1, it evaluates the F1 measure by comparing Chi-square and mutual information selector in Nave Bayes classifier. According to Table 1, it can be

FS	NB
	Min Avg Max
CHI	0.93 0.947 0.95
MI	0.27 0.59 0.90

Table 1: Table 1 Average F1 measure value [1]

concluded that Chi-square has a stable performance which is more suitable for this project.

2.3.2 Feature selection conformation

After finding feature by statistical words frequency and manual finding feature associated with the geolocations, the total number of candidate feature is 102.
Since weka has Chi-square feature selection methods API to choose to rank the attributes, 60 features is finally confirmed according to the rank order.

2.4 Vector Space Model and ARFF Files

It is impractical to train the system on the basis of whole twitter datasets. For geolocation prediction model, weka requires an ARFF file with data in vector spaces model. In my Java program, vector spaces model generated by using attributes as tokens to match and count attributes frequency in of tweet content. ARFF file generate by merging attributes, ID, vector spaces model, geolocation.

3 Applied Machine Learning

3.1 Sequential minimal optimization (SMO)

SMO is a fast algorithm for training Supported Vector Machine which has higher efficiency and better performance in dealing with sparse data, and thus SMO might be the best machine learning algorithm for this project. In addition, it can solve high dimensional problem and non-linear problem, and it can avoid minimum point and structure choice problem of neural network.

```
=== Summary ===

Correctly Classified Instances       20938               30.5522 %
Incorrectly Classified Instances     47594               69.4478 %
Kappa statistic                          0.0683
Mean absolute error                      0.3085
Root mean squared error                  0.4104
Relative absolute error                 96.9638 %
Root relative squared error            102.8609 %
Total Number of Instances            68532
```

Figure 1. The result of SMO

3.2 Nave Bayes and its derived Multinomial Nave Bayes

Nave Bayes model is constructed based on classical mathematics with solid mathematical foundation. It has high and stable classification efficiency in various applications, such as email spam detection, personal email sorting and language detection. It can be proved that the conditional independence assumption between attributes when implement this project. Based on this project, the derived formula for Nave Bayes as following:

$$P(class \mid Attributes) = \frac{P(Attributes \mid class)P(class)}{P(Attributes)} \quad (1)$$

Multinomial Nave Bayes is a derived specialized version of Nave Bayes that explicitly models the words counts and adjusts the Nave Bayes formula for dealing with text documents.[1] Nave Bayes and Multinomial Nave Bayes classifiers are implemented in weka package, and here is the classification result:

```
=== Summary ===

Correctly Classified Instances       19342               28.2233 %
Incorrectly Classified Instances     49190               71.7767 %
Kappa statistic                          0.032
Mean absolute error                      0.3137
Root mean squared error                  0.3964
Relative absolute error                 98.6234 %
Root relative squared error             99.343  %
Total Number of Instances            68532
```

Figure 2 . The result of Nave Bayes

```
=== Summary ===

Correctly Classified Instances       20594               30.0502 %
Incorrectly Classified Instances     47938               69.9498 %
Kappa statistic                          0.065
Mean absolute error                      0.305
Root mean squared error                  0.3916
Relative absolute error                 95.8634 %
Root relative squared error             98.1505 %
Total Number of Instances            68532
```

Figure 3 . The result of multinomial Nave Bayes

3.3 Zero-R

According to lecture, the classifier method of Zero-R is that classify all instances based on class with most frequency in the training data. Zero-R classifier is implemented in weka package, and here is the result:

```
=== Summary ===

Correctly Classified Instances       17929               26.1615 %
Incorrectly Classified Instances     50603               73.8385 %
Kappa statistic                          0
Mean absolute error                      0.3181
Root mean squared error                  0.399
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances            68532
```

Figure 4 . The result of Zero-R

3.4 One-R

According to lecture, the classifier method of One-R is that create one rule for each attribute in training data, and rule with smallest error rate is selected as its one rule. One-R classifier is implemented in weka package, and here is the result:

```
=== Summary ===

Correctly Classified Instances       15772               23.0141 %
Incorrectly Classified Instances     52760               76.9859 %
Kappa statistic                          0.0252
Mean absolute error                      0.3079
Root mean squared error                  0.5549
Relative absolute error                 96.8003 %
Root relative squared error            139.0822 %
Total Number of Instances            68532
```

Figure 5 . The result of One-R

3.5 Results Comparison different Machine learning algorithm

	Accuracy	Time
NB	28.2233%	18.4s
NBM	30.0502%	6.6s
SMO	30.5522%	6.02s
ZeroR	26.1615%	5.81s
OneR	23.0141%	5.8s

Table 2: Results

According to Table 2, SMO with 30.5522% accuracy and consuming time 5.81s is proved to be best suit for this project because of its higher efficiency and better performance in dealing with sparse data. NaiveBayes and OneR have same low accuracy 23.0141%, but the consuming time of oneR is much less than NaiveBayes. NaiveBayesMutional as a derived specialized version of Nave Beyes that explicitly models the words counts and adjusts the Nave Bayes formula for dealing with text documents, has a better accuracy and much less consuming time than NaiveBayes, and it almost catches up on SMO.

3.6 K-Nearest Neighbor(KNN)

KNN is one of the popular classification algorithms in machine learning. But the output is determined by the value of K as shown in figure 1, it can be seen that, the unknown point could be classified as blue if K=3, while it could be classified as red if K=9. In this project, the vector space model of majority tweets content is 0, which is labeled as non-geolocation. In this case, with K setting with large value, it can be sure that most of tweets will be classified as non-geolocation. To be specific, it is difficult to find a appropriate K value. In addition, KNN is a lazy learner since m train instance and n test instance will require m*n times of calculation.

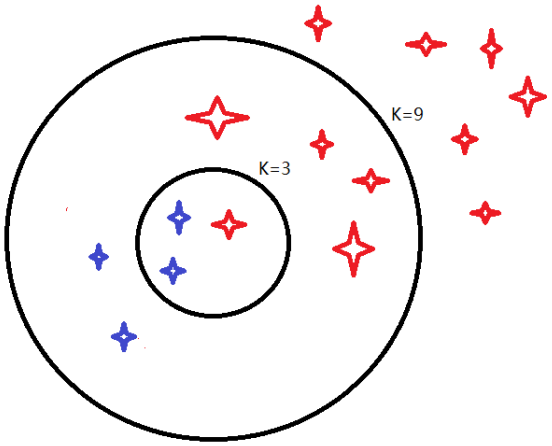

Figure 6. Sample space model

3.7 Decision Tree

When use decision tree to implement project, it can be found that tweets dataset need to be scanned and sorted more than once, which lead to low efficiency.

4 Error Analysis

Feature engineering is extremely important step in finding Attributes and thus constructing geolocation prediction model. If something wrong happens, it will bring catastrophic influence on geolocation prediction model.

Firstly, the method for vector space model split the potential attributes which is in forms of several words into separated single words, such as white house and obama care. Secondly, attributes are chose only according to the top 200 word occurrence frequency, and some words with high representative may be filtered and

thus reduce the efficiency of geolocation prediction model. Finally, the application of stemming and stop words could merge or eliminate some two-letter words may be a localism of a place, and thus reduce the efficiency of geolocation prediction model. Therefore, it is a tradeoff to use stemming and stop words.

5 Conclusions

In this project, Feature Engineering is implemented in two steps, namely find feature by Statistical Words Frequency and manually find Features. Tweets datasets is firstly processed by using removing duplicate tweets, stemming and stop words, then by counting words frequency and choose 10 words with high representative features for each class. After that, manually choose the feature based on conducting an investigation which mainly focuses on local history, climate, culture, landmark architecture and famous university for each city. After choose 102 candidate attributes, Chi-square is applied to filter these features. Finally, 60 features are successfully chosen and transform the tweet dataset into ARFF file for geolocation prediction model and classification.
In machine learning experiment part, SMO and f Multinomial Nave Bayes are proved to be the best suitable algorithm for text documents classification. Zero-R, One-R and Nave Bayes have poor performance in text documents classification, particularly Naive Bayes has low efficiency. KNN and decision tree are not suit for text documents classification.

6 Reference

[1] P. Domingos and M. Pazzani. On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning, 29:103130, 1997.