Letter Recognition Data Analysis

Xiaoyi Long
Wuhan University
2018302100026@whu.edu.cn

***Abstract*— The Data Mining Process is a powerful technique that helps not only in decision making based on the data that is available but also helps in predicting the potential change or result that might occur in the future. Data Mining Technique can come up with various useful predictions that usually cannot be interpreted using the graphical reporting. Classification is one of the data mining techniques that help in classifying the items according to the items with predefined set of classes. This paper will include evaluation of three sudifferent algorithms(Naïve Bayes, KNN, Multilayer Perceptron) using WEKA. In this project, I have implemented the Classification technique in predicting the 'lettr' attribute based on the other relevant fields.**

Keywords—data mining, Weka, evaluation

I. Probelm Statement

The objective is to identify each of a large number of black-and-white rectangular pixel displays as one of the 26 capital letters in the English alphabet.I have trained on the first 16000 items and then use the resulting model to predict the letter category for the remaining 4000.

II. Data Set

A. Data Set Information

The character images were based on 20 different fonts and each letter within these 20 fonts was randomly distorted to produce a file of 20,000 unique stimuli. Each stimulus was converted into 16 primitive numerical attributes (statistical moments and edge counts) which were then scaled to fit into a range of integer values from 0 through 15. Dataset was obtained from UCI Website.

https://archive.ics.uci.edu/ml/datasets/Letter+Recognition

B. Attribute Description

1. lettr capital letter (26 values from A to Z)
2. x-box horizontal position of box (integer)
3. y-box vertical position of box (integer)
4. width width of box (integer)
5. high height of box (integer)
6. onpix total # on pixels (integer)
7. x-bar mean x of on pixels in box (integer)
8. y-bar mean y of on pixels in box (integer)
9. x2bar mean x variance (integer)
10. y2bar mean y variance (integer)
11. xybar mean x y correlation (integer)
12. x2ybr mean of x * x * y (integer)
13. xy2br mean of x * y * y (integer)
14. x-ege mean edge count left to right (integer)
15. xegvy correlation of x-ege with y (integer)
16. y-ege mean edge count bottom to top (integer)
17. yegvx correlation of y-ege with x (integer)

III. Pre-Progressing Steps

Data preprocessing transforms the data into a format that will be more easily and effectively processed by the algorithm. Real world data are incomplete, noisy, and inconsistent. There are attributes which are false predictors and has missing values, noise, error, other data discrepancies.

A. Trasformation of initial data set

Data preprocessing is one of the most critical steps in a data mining process which deals with preparation and transformation of the initial data set.

Step1. Download *letter-recognition* in .data and .names format, Open it in notepad++, organizing them as one file.

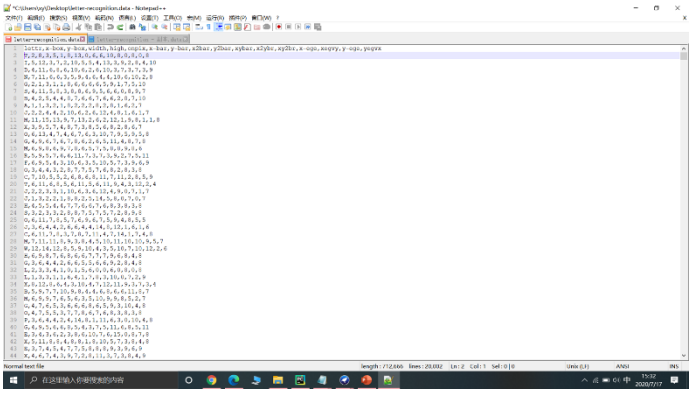


Fig. 1. Letter-recognition.data

Step2. Open it in Excel, and Save it as csv format

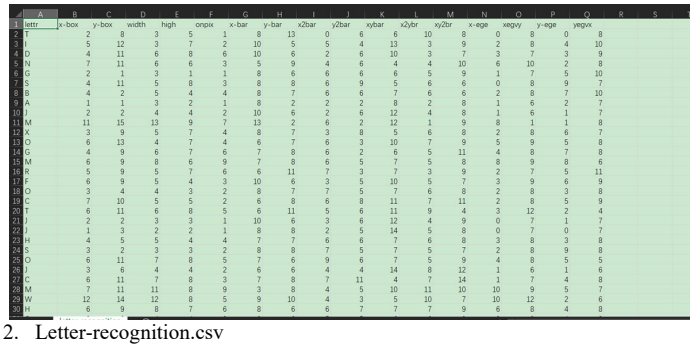


Fig. 2. Letter-recognition.csv

Step3. Open it in Weka, as csv format, then save as arff format.

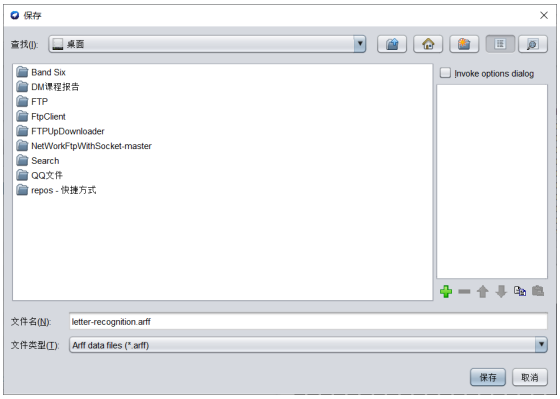


Fig. 3. Save as .arff

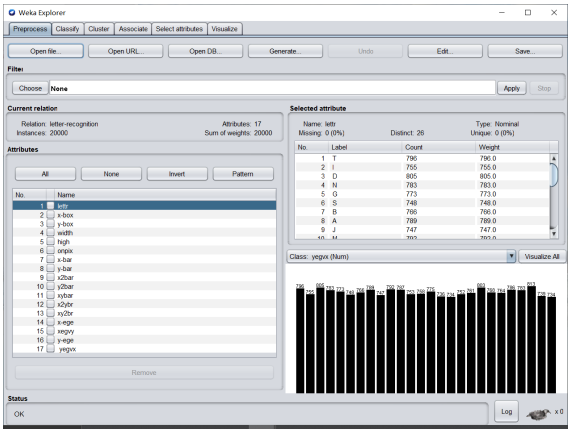


Fig. 4. Open it in Weka

B. *Split the data set into Training set and test set*

- training set—a subset to train a model.
- test set—a subset to test the trained model.

slicing the single data set as follows:

Fig. 5. A horizontal bar divided into two pieces: 80% of which is the training set and 20% the test set.

Make sure that the test set meets the following two conditions:

1. Is large enough to yield statistically meaningful results.

2. Is representative of the data set as a whole. In other words, don't pick a test set with different characteristics than the training set.

I have split it in Python, here is the code.

```python
import sklearn
import pandas
from pandas import DataFrame
from sklearn.utils import shuffle

df = pandas.read_csv("letter-recognition.csv")
rate = 0.8

df = shuffle(df)
size = df.shape[0]
train_size = int(size * rate)
train_df = df[:train_size].reset_index(drop=True)
test_df = df[train_size:].reset_index(drop=True)


train_df.to_csv("train.csv")
test_df.to_csv("test.csv")
```

Fig. 6. Python code, Slicing a single data set into a training set and test set.

IV. DATAVISUALIZATION

Data visualization is a general term that describes any effort to help people understand the significance of data by placing it in a visual context such as patterns, graphs, trends and correlations that might go undetected in text-based data but can be recognized easily with data visualization softwareword.

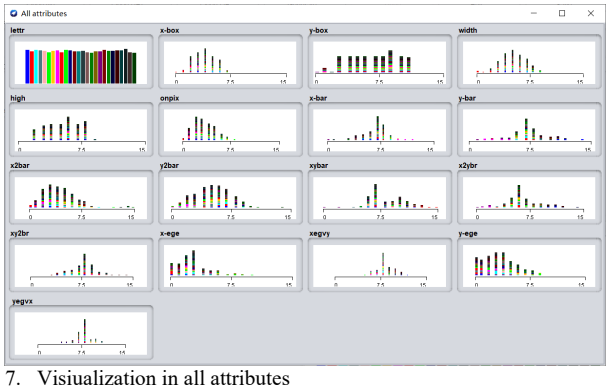


Fig. 7. Visiualization in all attributes

V. EXPERIMENT

A. *Naïve Bayes Classifier*

The naïve Bayes classifier is one of the simplest approaches to the classification task that is still capable of providing reasonable accuracy. Bayesian inference, of which the naïve Bayes classifier is a particularly simple example, is based on the Bayes rule that relates conditional and marginal probabilities. There are two major approaches in applying Bayesian inference to the classification task: model-probability inference and class-probability inference. The naïve Bayes algorithm may need some minor enhancements before it is ready to work using real-world data; the chapter reviews the most important practical issues that need to be taken care of. Missing attribute values are likely to decrease model quality for any modeling algorithm, when occurring for training instances, or classification accuracy, when occurring for classified instances. The not-so-naïve versions of the naïve Bayes classifier substantially increase the computational complexity of model creation and prediction.

1) Cross-validation Folds:10

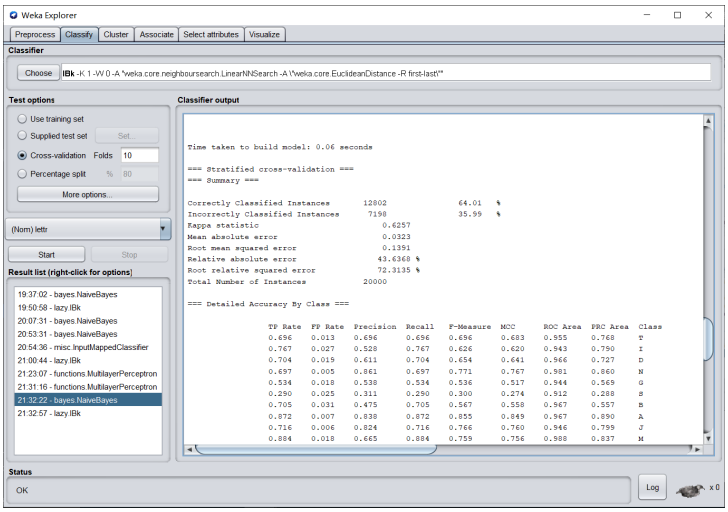


Fig. 8. Classifier output

2) 80%traning set and 20% test set

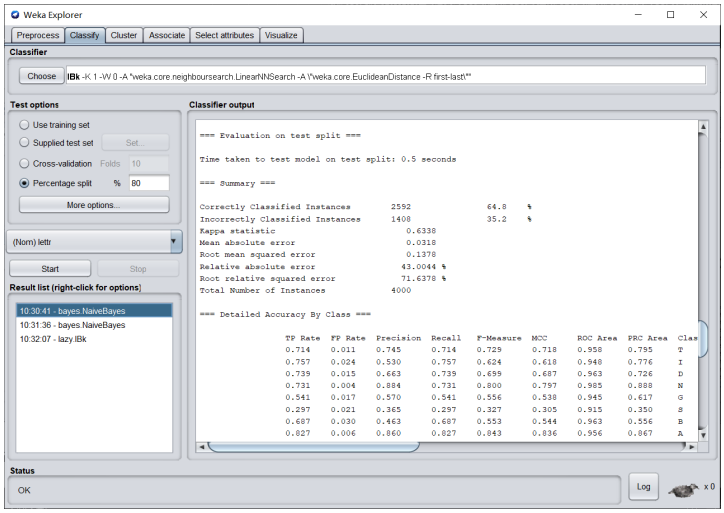


Fig. 9. Classifier output.

B. KNN

In pattern recognition, the k-nearest neighbors algorithm (k-NN) is a non-parametric method proposed by Thomas Cover used for classification and regression.

In k-NN classification, the output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor.

1) Cross-validation Folds:10

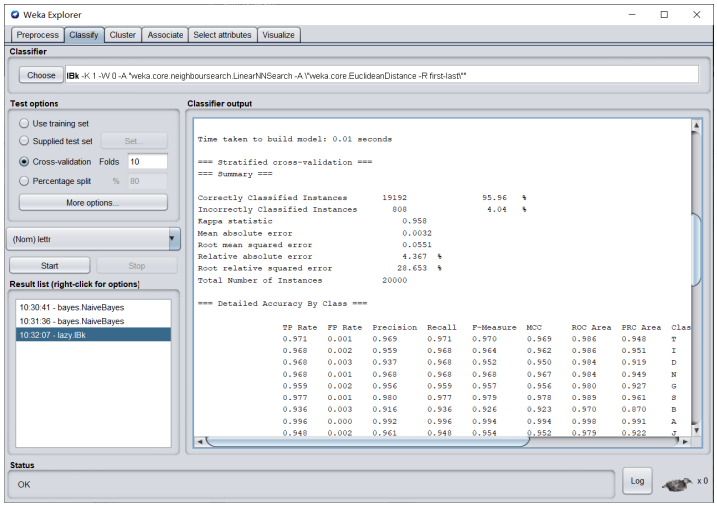


Fig. 10. Classifier output

2) 80%traning set and 20% test set

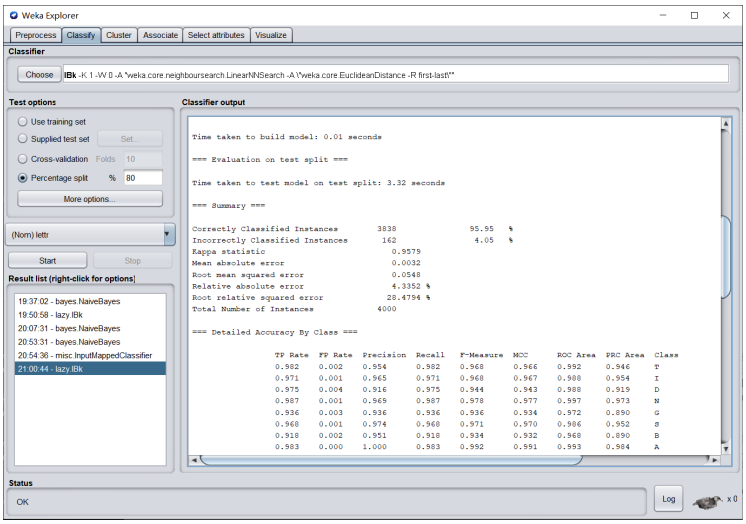


Fig. 11. Classifier output

C. Multilayer Perceptron

A multilayer perceptron (MLP) is a class of feedforward artificial neural network (ANN). The term MLP is used ambiguously, sometimes loosely to any feedforward ANN, sometimes strictly to refer to networks composed of multiple layers of perceptrons (with threshold activation); Multilayer perceptrons are sometimes colloquially referred to as "vanilla" neural networks, especially when they have a single hidden layer.

An MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training.Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable.

1) Cross-validation Folds:10

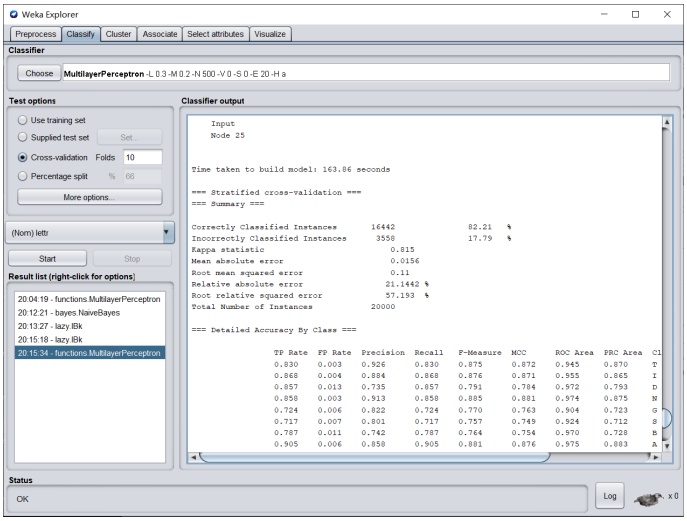


Fig. 12. Classifier output

2) 80%traning set and 20% test set

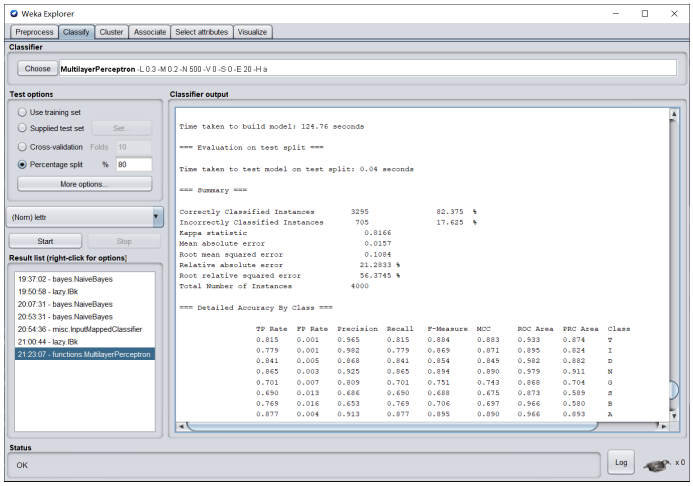


Fig. 13. Classifier output

VI. Conclusion

After many experiments, I found that this data set has the best classification effect on the KNN model. Here are the classified results using KNN.

TABLE I. Training Results

Algorithm	Accuracy on data set	
	Cross-valition Fold:10	***80%traning set and 20% test set***
Naïve Bayes	64.01%	64.8%
KNN	95.96%	95.95%
MLP	82.21%	82.375%

This experiment is to do some analysis on the Letter Recognition Data Set data. Through this data mining experiment, I learned the concepts and knowledge of data mining, understood the purpose and use steps of data mining, and further studied WEKA open source data. How to use mining tools in data mining learning. Through this experiment, I also realized that after data mining explores a large amount of data, it can reveal the hidden regular content, and further form a model analysis method. Different types of models can be established for the whole or part of a certain business process, can describe the current situation and regularity of development, and can be used to predict the situation that may occur when conditions change. This can provide a better support basis for subsequent research. This experiment went smoothly, which gave me a deeper understanding of how to perform classification analysis in Weka. I also mastered the Bayesian algorithm, K nearest neighbor algorithm, and Multilayer Perceptron algorithm for classification and analysis in Weka. I also deeply realized the importance of data preprocessing for data mining. Correct data preprocessing is the basis of data analysis. In addition, the process of data analysis requires solid basic knowledge and careful experimentation. This course experiment has benefited me a lot.

References

[1] *Data pre-processing*
http://www.cs.ccsu.edu/~markov/ccsu_courses/DataMining-3.html

[2] *Data Cleaning and data pre-processing*
http://www.mimuw.edu.pl/~son/datamining/DM/4-preprocess.pdf

[3] *Data Pre-Processing*
http://searchsqlserver.techtarget.com/definition/data-preprocessing

Rumor Detection on Social Media

A Lierature Reiew on Rumor Detection

Xiaoyi Long
Wuhan University
2018302100026@whu.edu.cn

***Abstract*—Social media platforms have been used forinformation and news gathering, and they arevery valuable in many applications. However, they also lead to the spreading of rumors andfake news. Many efforts have been taken to detect and debunk rumors on social media by analyzing their content and social context using machine learning techniques. This paper gives an overview of the recent studies in the rumor detection field. It provides some solutions to detect rumor on Twiter.**

***Keywords*—rumer dection, heterogeneous graph, user embedding**

I. Introduction

Rumors sometimes may spread very quickly over social media platforms, and rumor detection has gained great interest in both academia and industry recently. Government authorities and social media platforms are also taking efforts to defeat the negative impacts of rumors. In the following sub sections, I will introduce the rumor detection definition, the problem statement, solution and conclusion.

II. Related Work

A. Rumor Dection

Different publications may have different definitions for rumor. It is hard to do a head-tohead comparison between existing methods due to the lack of consistency. In this survey, a rumor is defined as a state ment whose truth value is *true*, *unverified* or *false*. When a rumor's veracity value is *false*, some studies call it "*false rumor"* or *"fake news"*. However, many previous studies give "*fake news*" a stricter definition: fake news is a news article published by a news outlet that is intentionally and verifiably false. The focus of this study is rumor on social media, not fake news. There are also different definitions for *rumor detection*. In some studies, rumor detection is defined as determining if a story or online post is a rumor or non-rumor, and the task of determining the veracity of a rumor (*true*, *false* or *unverified*) is defined as rumor verification. But in this survey paper, *rumor detection* is defined as determining the veracity value of a rumor.

B.Graph Neural Networks

The topic of graph neural networks has aroused widespread concern. Some researchers had extended mature neural network models,such as CNN, to apply to regular grid structures (twodimensional grids or one-dimensional series) that can be used for graphics of arbitrary structures. Based on their groundbreaking work, Kipf and Welling proposed a simple graph eural network model, called the Graph Neural Networks GCN), which achieved better results than the benchmark method in the graph datasets. GCN has also been applied to various of applications, e.g., text classification relation extraction , image classification molecular fingerprints nd protein interface prediction . For the unstructured data in the task, the researchers sed the graph structure data of the external resource or assumed the relational structure in the task, and then used the graph convolution networks to directly operate on the graph. In this paper, they consider the integration of user behavior elationship network information into the rumor detection method, and a user behavior relationship network can be represented by a graph. Based on the excellent performance of graph convolutional networks in the representation of graph structure data, they propose to use graph convolution networks to model user for rumor detection.

C. Traditional Machine Learning Methods

The majority of early detection methods for rumors were based on statistical machine learning, which manually extracts effective features to identify rumor from the text contents, user profiles, and propagation patterns of the rumors. One class of these approaches combined different types or different types of temporal variations of features to detect rumors. Castillo et al. [3] proposed a series of features consists of message-based, user-based, topic-based, and propagationbased for rumor detection, and Ma et al. [4] explored a novel approach to capture the temporal characteristics of social context features based on the time series of rumor's lifecycle. Another class exploited the topological-structure features extracted from the source tweet propagation of rumors to detect rumors. Ma et al. presented an SVM classifier with a tree-based kernel function calculating the similarity of the propagation topological tree to identify rumors.

However, these methods are time-consuming and laborintensive due to the features manually extracting from text contents, user profiles, and propagation patterns. And these features depend on the datasets and sometimes are impossible to be extracted.

D. Deep Learning Methods

Deep learning has been successfully applied in many realworld tasks, such as natural language processing. Researchers also begin to work on the deep neural networks for detecting rumors. One group of researchers exploited neural networks to capture the temporal information or topological-structure information of the source tweet propagation of rumorsfor rumor detection. Liu et al. [5] modeled the source tweet propagation as a sequence of user characteristics and

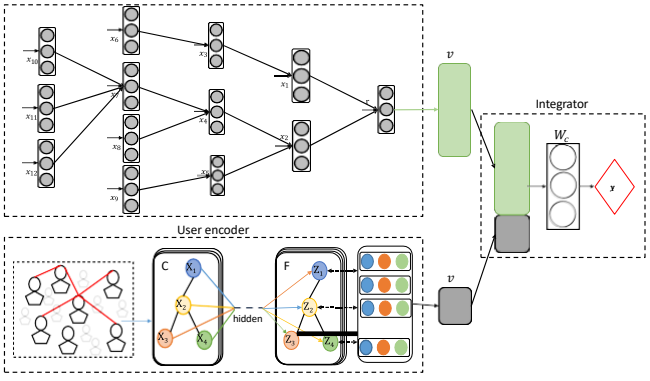
proposed a propagation path classifier composed of recurrence and convolutional networks to detect rumors. Yuan et al. [6] posed a global-local attention network (GLAN) to capture the global structure features of the source tweet propagation topological tree for rumor detection. Another group of researchers explored a framework consists of three modules to capture the information from text contents, user profiles, and propagation patterns for rumor detection. Huang et al. [7] utilized a graph convolutional network to model the user graph formed by user behaviors and combined the powerful user representation with the representation of the propagation tree for rumor detection on twitter.

However, these methods ignore the global semantic relation in the text content of rumors, and its integration with the information involved in the source tweet propagation effectively not been solved. In this paper, they construct a heterogeneous tweet-word-user graph according to the text content and the source tweet propagation of rumors and propose a new metapath based heterogeneous graph attention network framework to capture the global semantic relation in text contents and effectively integrate it with the information involved in source tweet propagations for rumor detection.

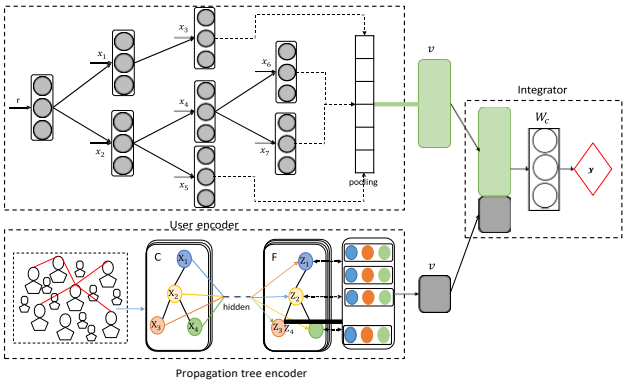
III. Proposed Modle

A.Deep Structure Learning for Rumor Detection on Twitter

In the paper,They proposed a model. This model consists of three parts, namely user encoder, propagation tree structure encoder with content semantics and integrator (see Fig. 2). User encoder obtains the user representation using graph convolutional networks to model the user graph. Propagation tree structure encoder encodes the propagation tree by a tree-based recursive neural network. And then integrator combines these features to a fully connected layer for rumor detection.



(a) hybrid model with bottom-up RvNN



(b) hybrid model with top-down RvNN

Fig. 2: An illustration of the proposed hybrid neural model

Given a rumor propagation tree structure and the poster of root node, the user encoder gets the user vector representation, the propagation tree encoder gets the propagation tree vector, and the integrator cascades above two vectors to identify rumors.

B.Heterogeneous Graph Attention Networks for Early Detection of Rumors on Twitter

We present a heterogeneous graph attention network framework to solve the problem of rumor detection on a heterogeneous graph. As shown in Fig. 3, the framework includes a subgraph attention network and subgraph-level attention. The subgraph attention network exploits an attention mechanism similar to graph attention network for capturing the global relation information of nodes. The subgraph-level attention introduces an attention mechanism to fuse the source tweet representation in different subgraphs for rumor detection.

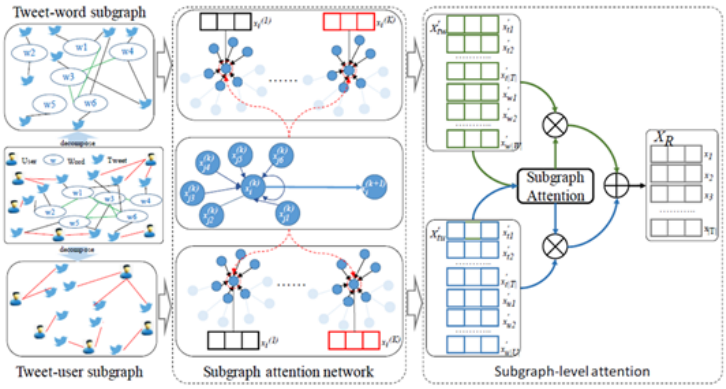


Fig. 3: The architecture of our heterogeneous graph attention networks for rumor detection.

Ⅳ. *RUMOR CLASSIFICATION PERFORMANCE*

A.Deep Structure Learning for Rumor Detection on Twitter

TABLE II: Results of rumor detection.(NR: non-rumor; FR: false rumor; TR: true rumor; UR: unverified rumor)

(a) Twitter15 dataset

Method	Acc.	NR	FR	TR	UR
		F_1	F_1	F_1	F_1
DTR	0.409	0.501	0.311	0.364	0.473
DTC	0.454	0.733	0.355	0.317	0.415
RFC	0.565	**0.810**	0.422	0.401	0.543
SVM-TS	0.544	0.796	0.472	0.404	0.483
SVM-HK	0.493	0.650	0.439	0.342	0.336
SVM-TK	0.667	0.619	0.669	0.772	0.645
GRU-RNN	0.641	0.684	0.634	0.688	0.571
BU-RvNN	0.708	0.695	0.728	0.759	0.653
TD-RvNN	0.723	0.682	0.758	0.821	0.654
BU-Hybrid	0.738	0.796	0.713	0.773	0.663
TD-Hybrid	**0.752**	0.699	**0.773**	**0.831**	**0.709**

(b) Twitter16 dataset

Method	Acc.	NR	FR	TR	UR
		F_1	F_1	F_1	F_1
DTR	0.414	0.394	0.273	0.630	0.344
DTC	0.465	0.643	0.393	0.419	0.403
RFC	0.585	0.752	0.415	0.547	0.563
SVM-TS	0.574	**0.755**	0.420	0.571	0.526
SVM-HK	0.511	0.648	0.434	0.473	0.451
SVM-TK	0.662	0.643	0.623	0.783	0.655
GRU-RNN	0.633	0.617	0.715	0.577	0.527
BU-RvNN	0.718	0.723	0.712	0.779	0.659
TD-RvNN	0.737	0.662	0.743	0.835	0.708
BU-Hybrid	0.735	0.706	0.735	0.860	0.636
TD-Hybrid	**0.773**	0.716	**0.756**	**0.870**	**0.756**

As shown in Table Ⅱ, our proposed model basically achieved better performance than the other methods on the two datasets via capturing the user embedding with graph convolutional networks. It is observed that the performance of four baseline methods in the first group based on manual features are very poor, varying between 0.409 and 0.585 in accuracy. DTR ranks the candidate rumor cluster which collected related tweets matching a set of regular expressions to identify rumors. While only a few posts match these regular expressions, it preforms the worst. Compared to the DTC, the SVM-TS and RFC take into account the time-series structure to model variation of handcrafted features, so their effect is better than DTC. When observing the performance of the method considering the propagation of structural features, except for SVM-HK, other methods are superior to methods based on handcrafted features. And TD-RvNN achieves the best performance among all the baselines. This is because posters will correct some inaccurate information during the propagation of information in social media and TD-RvNN captures this self-correcting information effectively. However, all baseline do not adequately extract user information, especially the behavior information of user. Our model uses a graph convolutional network for learning user representations to model user features and behaviors. The experimental results show that they have learned a valid user representation for rumor detection, and graph convolution networks captures the node attributes and structural attributes in the graph well. For only the non-rumor class, our method is worse than the RFC and SVM-TS on the datasets. Compared to RFC and SVM-Ts, they not only consider the statistical features of user, but also consider the user behavioral network information. Since the users group behavior is more likely to spread rumors, so when they consider the user behavior information, it can improve the accuracy of rumor detection. Meanwhile, it bring some interference to the detection of non-rumor.

B.Heterogeneous Graph Attention Networks for Early Detection of Rumors on Twitter

Ⅴ. CONCLUSION

A.Deep Structure Learning for Rumor Detection on Twitter

In this paper, they present a hybrid neural network model for rumor detection on twitter. This is the first work that model user with graph convolutional networks for rumor detection. The model consists of three modules: a user encoder module modeling the graph formed by user behaviors based on graph convolutional networks, a propagation tree structure encoder represents the propagation tree using a recursive neural network, and an integrator to integrate feature representations. Our model considers three aspects of rumor detection: contents, users, and propagation. Experiments on real-worlddatasets show that our method is more efficient than previous methods. In future work, they plan to give weights to the edges in the user graph based on the stance of interaction between users.

B.Heterogeneous Graph Attention Networks for Early Detection of Rumors on Twitter

Semantic relations among the text contents of rumors are ignored by the majority of existing rumor detection methods. To address this challenge, they constructed a heterogeneous tweet-word-user graph based on the text contents and the source tweet propagations of rumors. A meta-path based heterogeneous graph attention network framework was further proposed to learn the global semantic relations of text contents and effectively integrate them with the information related to source tweet propagations for rumor detection. Specifically, they first decomposed the heterogeneous graph into a tweet-word subgraph and a tweet-user subgraph according to the tweet-word and tweet-user meta path. Then the subgraph attention network was exploited to model the subgraphs for obtaining the representation of nodes with global structure information. Finally, they utilized an attention mechanism to integrate the representations of tweet nodes in different subgraphs for rumor detection. Experiments on two real-world Twitter datasets demonstrated that our method has better performance than the state-of-the-art baselines in accuracy and has a comparable ability on the early rumor detection task.

REFERENCES

[1] Huang, Qi, Junshuai Yu, Jia Wu, and Bin Wang. (2020). *Heterogeneous Graph Attention Networks for Early Detection of Rumors on Twitter*.

[2] Huang Q, Zhou C, Wu J, Wang M, Wang B. Deep Structure Learning for Rumor Detection on Twitter. *2019 International Joint Conference on Neural Networks (IJCNN), Neural Networks (IJCNN), 2019 International Joint Conference on*. July 2019:1-8. doi:10.1109/IJCNN.2019.8852468.

[3] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on twitter," in WWW, 2011, pp. 675–684.

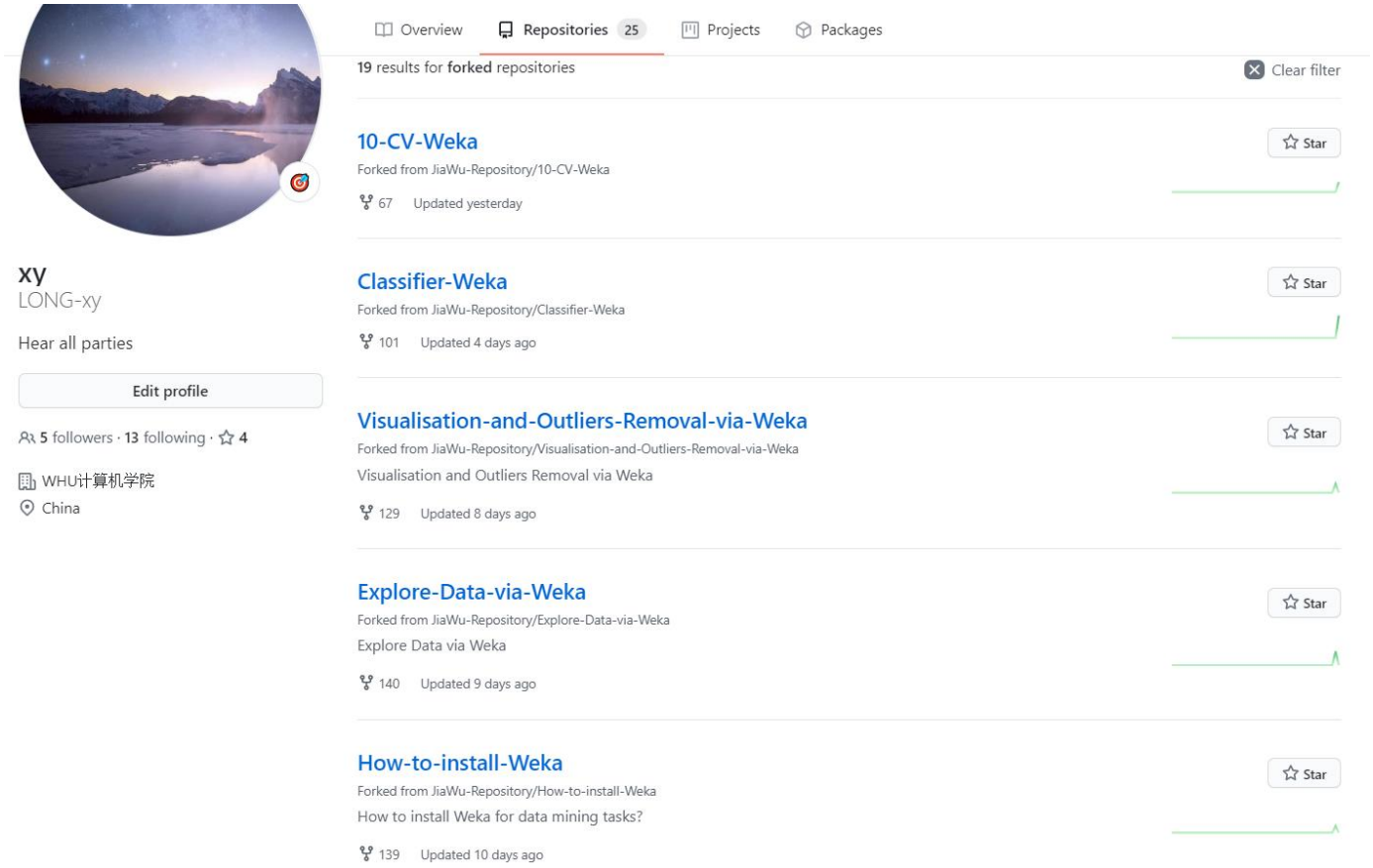
[4] J. Ma, W. Gao, Z. Wei, Y. Lu, and K.-F. Wong, "Detect rumors using time series of social context information on microblogging websites," in CIKM, 2015, pp. 1751–1754.

[5] Y. Liu and Y.-F. B. Wu, "Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks," in AAAI, 2018.

[6] C. Yuan, Q. Ma, W. Zhou, J. Han, and S. Hu, “Jointly embedding the local and global relations of heterogeneous graph for rumor detection,”.arXiv preprint arXiv:1909.04465, 2019.

[7] A. Zubiaga, M. Liakata, R. Procter, G.-W.-S. Hoi, and P. Tolmie,“Analysing how people orient to and spread rumours in social media bylooking at conversational threads,” PloS one. vol. 11, e0150989, 2016.

Github 截图（关于课程部分的仓库）



COURSE REFLECTIONS（课程感言）

After 8 days of Computer Frontier theory courses, I systematically learned the relevant knowledge of data mining, including the definition and methods of data mining, the use of Weka software, probability theory and mathematical statistics, and neural networks. On the last day, two PHD students from Dr. Jia Wu from Macquarie University in Australia held two lectures for us. The first one is trust prediction in social networks, and the other is to guide cross-language entity alignment through adversarial knowledge embedding. Although our current level of knowledge does not fully understand the content of the lecture, we have also encountered some new knowledge and benefited greatly. In the class, the teacher's explanation is very careful, and sometimes guides us in the actual operation in Weka. Questions in the classroom can be asked at any time, and the teacher will answer patiently. After class, students will also have heated discussions in the QQ group of the course to help each other and solve the problems which we encountered. After the class, I also wrote the course report carefully, reviewing the knowledge I learned. I am very lucky to have the opportunity to participate in this course.