HW3

Huajun Chai

998584845

Predict Closed Questions on Stack Overflow

Introduction

In this homework assignment, we are given some csv files about the posts on stack overflow. The posts have attributes of postid, userid, title, body, time, openstatus and so on. We are asked to use machine learning technique to train a model using part of the data, and then use that model to predict the status of a certain question.

Tools used

Since we are asked to predict the status of the post, and the post can have 5 potential status:

- Not a real question
- Too localized
- Not constructive
- Off topic
- Open

The problem is a multi-class classification problem. We can use algorithms like SVM, random forest, k-nearest and so on to categorize them and make predictions.

In my work, I turn to an existing tool developed by a group of genius from Yahoo and Microsoft. This tool is known as Vowpal Wabbit. VW is the essence of speed in machine learning, able to learn from terafeature datasets with ease. Via parallel learning, it can exceed the throughput of any single machine network interface when doing linear learning, a first amongst learning algorithms[1].

Step through

First:

You need to make sure your machine has the Boost library installed since VW relies on that library.

You can download and install boost 1.55 from this website: http://www.boost.org/.

Download-> extract from compressed file->go to the main folder, run "./bootstrap.sh".->Then run "./b2 –prefix==PREFIX"(here, PREFIX is the location that you want you boost library to install)

Second:

You need to install VW. There are multiple versions of VW available. But the most recommended one is that from github. It is the most recent version, and fewer bugs. You can clone the whole repository from here: https://github.com/JohnLangford/vowpal_wabbit.

Go to vowpal_wabbit folder, run "make"

After all these are done, we are good to run our code.

How to do

The VW tool takes a specific format of input. So before we can actually use the tool, we need to do some data pre-processing.

First,

We need to extract the desired columns to feed to the training model. Here we simply choose

- Post_id
- Reputation
- Title
- Body
- Tags

as the feature columns.

We use the following command to convert the original csv file to the modified version of csv:

```
python convert_csv.py ${WD}/data/train-sample.csv ${WD}/tmp/train-sample-processed.csv
```

After this, we will feed the new csv file into another convertor function to form a vw format file:

```
python convert_vw.py ${WD}/tmp/train-sample-processed.csv ${WD}/tmp/train-sample.vw
```

Now we have the vw format input, we are ready to do the training and predictiong process.

We use the "train-sample.csv" to train the model:

```
${VW} --loss_function logistic --oaa 5 -d ${WD}/tmp/train-sample.vw -f train_model
```

Here, --oaa 5 means "One against all, and use 5 classes classification". Loss function is chosen as logistic. After this, we will get a train_model file, which will be used in the prediction step.

```
${VW} --loss_function logistic --oaa 5 -i train_model -t -d ${WD}/tmp/public_leaderboard.vw -r predictions.txt
```

We use the command line above to do the prediction. –t tells the tool not to train, just predict. And we will get a file named predictions.txt. This file is not the final result, we need to further process it.

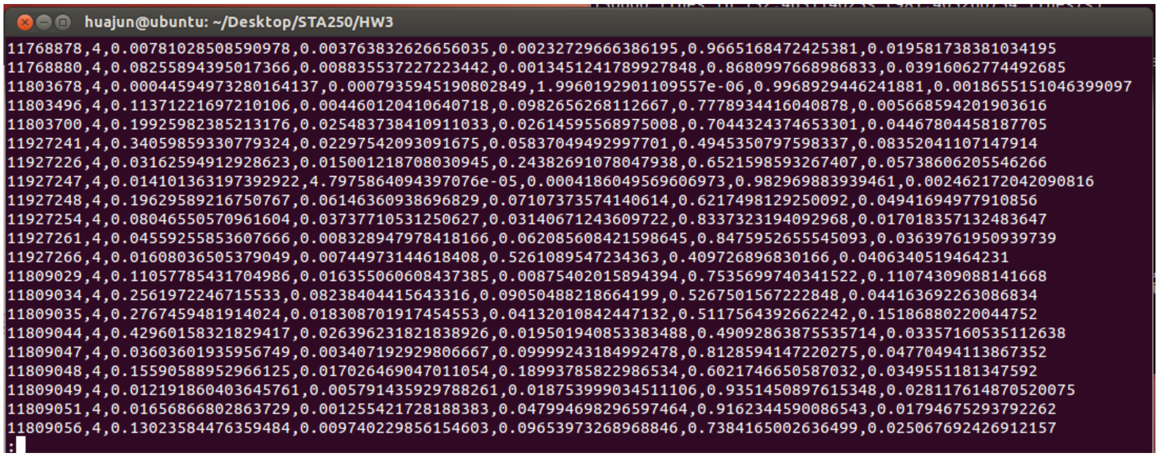
```
python sigmoid.py predictions.txt prediction_result.txt
```

Now the entries are normalized, the value represents the probability that a post is in that status.

Result and validation

Training set: train-sample.csv

Prediction set: public_leaderboard.txt



```
11768878,4,0.00781028508590978,0.003763832626656035,0.00232729666386195,0.9665168472425381,0.019581738381034195
11768880,4,0.08255894395017366,0.008835537227223442,0.001345124178992784 8,0.8680997668986833,0.03916062774492685
11803678,4,0.00044594973280164137,0.0007935945190802849,1.9960192901109557e-06,0.9968929446241881,0.0018655151046399097
11803496,4,0.11371221697210106,0.004460120410640718,0.0982656268112667,0.7778934416040878,0.005668594201903616
11803700,4,0.19925982385213176,0.02548373841091103 3,0.02614595568975008,0.7044324374653301,0.04467804458187705
11927241,4,0.34059859330779324,0.02297542093091675,0.05837049492997701,0.4945350797598337,0.08352041107147914
11927226,4,0.03162594912928623,0.015001218708030945,0.2438269107804793 8,0.6521598593267407,0.05738606205546266
11927247,4,0.01410136319739292 2,4.7975864094397076e-05,0.0004186049569606973,0.982969883939461,0.002462172042090816
11927248,4,0.19629589216750767,0.06146360938696829,0.07107373574140614,0.6217498129250092,0.04941694977910856
11927254,4,0.08046550570961604,0.03737710531250627,0.03140671243609722,0.8337323194092968,0.01701835713248364 7
11927261,4,0.04559255853607666,0.008328947978418166,0.062085608421598645,0.8475952655545093,0.03639761950939739
11927266,4,0.01608036505379049,0.007449731446184 08,0.5261089547234363,0.40972689683016 6,0.040634051946423 1
11809029,4,0.11057785431704986,0.01635506060843738 5,0.00875402015894394,0.7535699740341522,0.11074309088141668
11809034,4,0.256197224671553 3,0.0823840441564331 6,0.09050488218664199,0.526750156722284 8,0.04416369226308683 4
11809035,4,0.2767459481914024,0.01830870191745455 3,0.0413201084244713 2,0.5117564392662242,0.1518688022004475 2
11809044,4,0.4296015832182941 7,0.026396231821838926,0.019501940853383488,0.4909286387553571 4,0.033571605351126 38
11809047,4,0.03603601935956749,0.003407192929806667,0.09999243184992478,0.8128594147220275,0.04770494113867352
11809048,4,0.1559058895296612 5,0.01702646904701105 4,0.1899378582298653 4,0.6021746650587032,0.0349551181347592
11809049,4,0.012191860403645761,0.005791435929788261,0.018753999034511106,0.9351450897615348,0.028117614870520075
11809051,4,0.01656866802863729,0.00125542172818838 3,0.04799469829659746 4,0.916234459008654 3,0.01794675293792262
11809056,4,0.1302358447635948 4,0.009740229856154603,0.09653973268968846,0.7384165002636499,0.02506769242691215 7
:
```

The first column is the post_id, and the second column is the prediction result which has the following table:

1: Not a real question

2: Not constructive

3. Off topic

4. Open

5 Too localized

The rest columns are the probability that this post is in that status. Which choose the max probability to assign the status for that post.

There are benchmark's on the kaggle.com. We can use that to evaluate the result of ours.

Note

I write a Makefile for this program in order to make life easier:

If you want to make prediction, use "make" in terminal.

If you want to clean files, use "make clean" in terminal.

PS: I'm really sorry Professor that I do a shitty job on this assignment. I want to do my best. But time is just not enough for me. This is definitely my fault. I don't have a good time schedule this quarter. Some of the structure and architecture of the code for this assignment was borrowed from other online resources. You can refer to here if you like: http://fastml.com/predicting-closed-questions-on-stack-overflow/. The online resource has some bugs and defects. I have understood it, and fixed them. Again I apology for the delay of submission, and not that good job for this assignment. I know you must be disappointed, but I hope you won't be angry.

Best,

Huajun

[1] http://hunch.net/~vw/