Solution Abstract Problem 2: Almost Famous

Cindy Lamm

Monday 19th January, 2015

1 Approach

I identified question 1 as classification problem (supervised or unsupervised) which I wanted to solve using Spark (after the performance problems I encountered for the classification in R with 7.3 million observations).

I opted for the following approach to solve the questions:

- analyse file structure
- explore data
- prepare data for modeling
- estimate and validate models for spam classification
- prepare the web data for prediction
- classify entries in web data as spam
- remove spam entries from web data
- calculate required metrics

1.1 Analyse file structure

I had a short look into the log data in json format via jq. Once I saw that the data is very regular (same number of keys per row, same names for keys) I converted the data to csv using json2csv (which is pre-installed in the data science toolbox) to load it into R for exploratory analysis.

1.2 Explore data

To get an impression of the web and spam data I had a look at overall summary statistics but also at summary statistics aggregated over visits and visitors. Since I am not that familiar with web analytics I assume that a `uid` uniquely identifies a visitor which is a synonym for user.

From manual inspection it seems that visits that are generated via the bot network only contain the action `adclick` after they landed on the page. However if I would classify the web data according to this simple rule, then the click through rate for ads on the page would be zero - which seems too extreme.

1.3 Prepare data for modeling

In order to run a classification/clustering algorithm in Spark I need to code the categorical variables (all but `tstamp`) with integer levels (which means that I implicitely assume that it is okay if an algorithm treats these variables as continuous features, because the resulting model is still good enough). I converted the `tstamp` to epoch values (i.e. to seconds since 1970-01-01 00:00:00 GMT) to have a true continuous variable without loosing the time information.

For the supervised classification I added a variable `is_spam` with value 1 to the spam data and with value 0 to the web data (knowing that with this assumption some web data is falsely labeled). I pooled the spam data and different amounts of the web data into a training data sets - which I used as well for testing (see section 1.5).

1.4 Estimate and validate models for spam classification

I tried the different classification and cluster algorithms: I started with unsupervised clustering using k-means (with $k = 2$) on a training sample with all spam log data and (the top) 200.000 observations of the web log data[1] I had the hope that by using $k = 2$ all spam data would end up in one cluster and all web data in the other, so that when I predict the complete web data on this clustering all so far unknow spam entries go into the spam-cluster. However when I used the estimated cluster center to classify the training data, the result was not as good as I would have expected:

TODO: add venn diagram here or error/accuracy values

1.5 Current Limitations

Currently I used the same data for training, validation and prediction, which decreases the generalization performance of the estiamted model. It would be better to split the available spam and web log data into independent sets for training and validation. For classification/clustering one could use the time difference between the actions and/or the total time spent per visit as input. I also deem worth trying to cluster/classify on aggregated visit and/or user data. Given that I relabeled the categorical variables with (rather small) integer labels it might be necessary to scale the epoch values, because they otherwise bias the differences. It might also be better to use 1-of-n encoding for the categorical variables instead of just giving them numerical labels.

2 Used Software

In general I developed on a Macbook Pro with a 2.3 GHz Intel Core i7 Processor and 16GB Memory.

- git
- jq
- json2csv (pre-installed in the data science toolbox)
- LateX
- R in RStudio with packages knitr, plyr
- shell command line
- Spark with Python API and MLlib package

3 Time Spent

-

Summing up I spend xx hours on this problem.

[1] Using random 200.000 observations from the web log data didn't seem to be a good idea given the timely ordered structure.