Training and Predicting Criteo's Terascale Data Set on a Machine with 128GB RAM

Li Chen

October 19, 2017

1 Introduction

In 2015, Criteo released a 3TB data set [1] for click-through-rate (CTR) prediction. Logistic regression (LR) is a common technique for the task. With the file being 3TB in size, which is usually larger than the RAM of most machines, typical solvers such as `LIBLINEAR` by Fan et al. [2] cannot be used. The reason is that these algorithms load the entire data into the memory.

To train this data set, we consider the solver in `LIBLINEAR-CDBLOCK` by Yu et al. [5]. The solver splits the data into blocks, each of which can fit in the memory. Throughout the training process only one block of data is kept in the meomory at a time.

The raw data published by Criteo includes click logs of 24 days. As what Google [4] did, we made the first 23 days of data as training set and the 24th as the testing set. For feature engineering, we use the method from Juan et al. [3] which is a simplified version of the winning solution from a competition[1] hosted jointly by Kaggle and Criteo [3].

2 Obtaining the Data

The following command downloads and decompresses the data file.

```
$ wget https://s3-us-west-2.amazonaws.com/criteo-public-svm-data/criteo_tb.svm.tar.gz
$ tar -zxvf criteo_tb.svm.tar.gz
```

After decompression, there are two files: `criteo_tb` is the training set and `cirteo_tb.t` is the test set. To check the data integrity, you can check the md5sum by the following command.

```
$ md5sum criteo_tb
$ md5sum criteo_tb.t
```

The correct output values of training and testing set should be `67acf52010a3e412b1ec646a8c4978f7` and `f05978ecb7c8d954706bfe6e11777994` respectively.

3 Data Training with `LIBLINEAR-CDBLOCK`

We do the training on a machine with 5TB disk and 128GB RAM.

[1] `https://www.kaggle.com/c/criteo-display-ad-challenge`

3.1 Data Split

Because the solver consumes 66GB of RAM before loading any data due to the large number of training instances, training data must be split into 80 blocks to fit in our 128GB RAM. Run the following command to do the splitting job.

```
$ ./blocksplit -m 80 criteo_tb
```

The command above takes about 42 hours on our machine.

3.2 Training

For each block loaded into the memory, an L2-regularized LR sub-problem is solved by a dual coordinate descent method [5, 6]. Run the following command to do the training and obtain the model file `criteo_tb.s7.model` for future prediction.

```
$ ./blocktrain -s 7 criteo_tb.80 criteo_tb.s7.model
```

The procedure above takes about 55 hours for 5 outer iterations on our machine.

3.3 Prediction

Having the model obtained by training, we can make the prediction on the test data (data from the 24th day). The following command makes the prediction for every instance in the test file `criteo_tb.t` and saves results (probability outputs) in the file `s7_out`.

```
$ ./predict -b 1 criteo_tb.t criteo_tb.s7.model s7_out
```

The procedure above takes about 30 minutes on our machine.

3.4 Result

The following command evaluates the logarithmic loss of our prediction.

```
$ python3 logloss.py criteo_tb.t s7_out
```

The script `logloss.py` can be downloaded in the link[2].

The loss value should be about 0.1286 which is less than the value 0.1293 achieved by Google [4] for the same training/testing setting. The procedure above takes about 18 minutes on our machine.

References

[1] Criteo. Criteo releases industry's largest-ever dataset for machine learning to academic community. `http://www.criteo.com/news/press-releases/2015/06/criteo-releases-industrys-largest-ever-dataset/`, 2015.

[2] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: a library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[3] Y. Juan, Y. Zhuang, W.-S. Chin, and C.-J. Lin. Field-aware factorization machines for CTR prediction. In *Proceedings of the ACM Recommender Systems Conference (RecSys)*, 2016.

[2] `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary/logloss.py`

[4] A. Sterbenz. Using Google cloud machine learning to predict clicks at scale. https://cloud.google.com/blog/big-data/2017/02/using-google-cloud-machine-learning-to-predict-clicks 2017.

[5] H.-F. Yu, C.-J. Hsieh, K.-W. Chang, and C.-J. Lin. Large linear classification when data cannot fit in memory. In *Proceedings of the Sixteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 833–842, 2010.

[6] H.-F. Yu, F.-L. Huang, and C.-J. Lin. Dual coordinate descent methods for logistic regression and maximum entropy models. *Machine Learning*, 85(1-2):41–75, October 2011.

A Number of Instances per Day

The training file is generated simply by concatenating data from day 1 to day 23 orderly. If users want to access data from a particular day, they can grab an interval of lines in the training set file. Such an interval can be easily obtained using the information below about the number of instances per day.

Number of instances per day	
Day	# of instances
1	195,841,983
2	199,563,535
3	196,792,019
4	181,115,208
5	152,115,810
6	172,548,507
7	204,846,845
8	200,801,003
9	193,772,492
10	198,424,372
11	185,778,055
12	153,588,700
13	169,003,364
14	194,216,520
15	194,081,279
16	187,154,596
17	177,984,934
18	163,382,602
19	142,061,091
20	156,534,237
21	193,627,464
22	192,215,183
23	189,747,893
24	178,274,637
Total	4,373,472,329