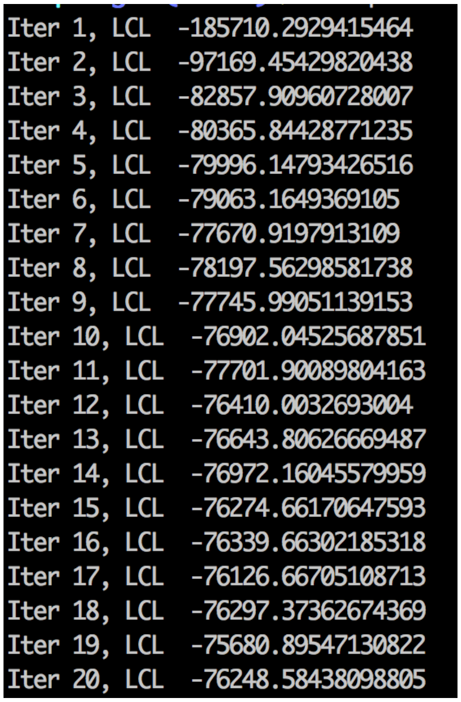
SGD for 10605 15 Fall

Jingyuan Liu
AndrewId: jingyual
jingyual@andrew.cmu.edu

1 Question 1

As we can see from the figure, the algorithm would converge at last. The LCL was close in the last several iterations.



```
Iter 1, LCL  -185710.2929415464
Iter 2, LCL  -97169.45429820438
Iter 3, LCL  -82857.90960728007
Iter 4, LCL  -80365.84428771235
Iter 5, LCL  -79996.14793426516
Iter 6, LCL  -79063.1649369105
Iter 7, LCL  -77670.9197913109
Iter 8, LCL  -78197.56298581738
Iter 9, LCL  -77745.99051139153
Iter 10, LCL  -76902.04525687851
Iter 11, LCL  -77701.90089804163
Iter 12, LCL  -76410.0032693004
Iter 13, LCL  -76643.80626669487
Iter 14, LCL  -76972.16045579959
Iter 15, LCL  -76274.66170647593
Iter 16, LCL  -76339.66302185318
Iter 17, LCL  -76126.66705108713
Iter 18, LCL  -76297.37362674369
Iter 19, LCL  -75680.89547130822
Iter 20, LCL  -76248.58438098805
```

Figure 1: LCL over iterations

2 Question 2

The accuracy curve of the 11 different parameters is:

We could see the general trend of the performances is first arising with the x, then decreasing. To see it in details, we can draw a figure in small range:

We could find that the best point is when mu is 0.001. I think this result is very reasonable. As we all know, the mu is used to do regularization. If the mu is too small, then the model would be overfitting, which would lower the performances. On the other hand, if mu is too big, then many parameters would vanish towards zero, which would make the model too "simple", and also lower the accuracy.

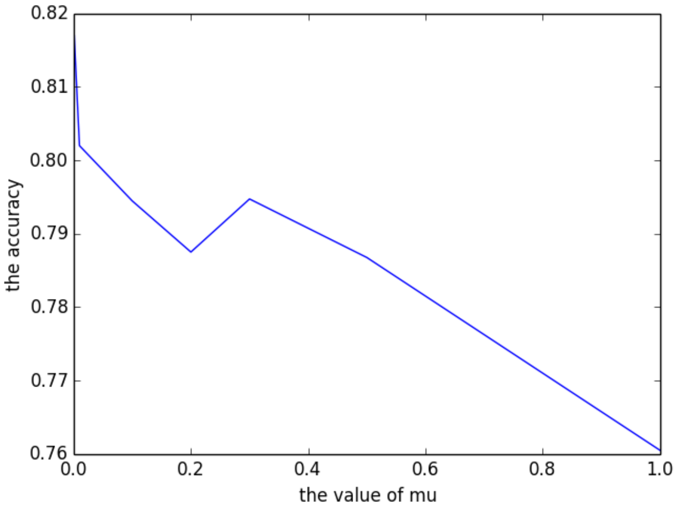


Figure 2: Accruracy over mu, whole range

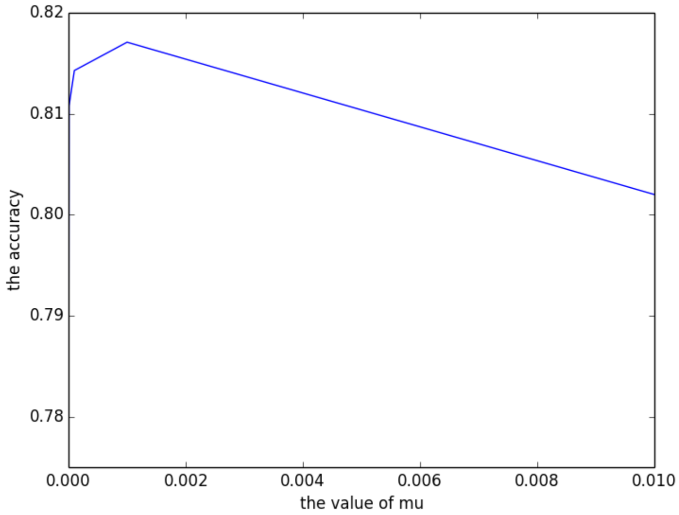


Figure 3: Accuracy over mu, small range

3 Question 3

The result is as follows, we could see that the best D is the 10^5.

Besides, we could see that the trend of the accuracy curve is very reasonable. We could see that the accuracy is improving with the increasement of x. However, the trend is becomming more and more slow. When x is small, the increasing of feature space would bring much performances improvment. When x is big, increasing the feature space would not be very effective. This is reasonable, and proves that hash tricks is an reasonable way for large datasets.

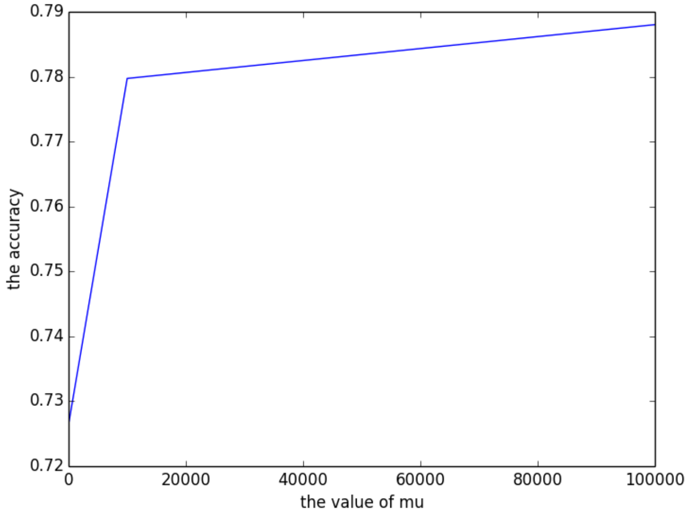


Figure 4: Accuracy over D

4 Question 4

We can use different threshold and got different iterations. The specific result is here:

First, threshold: 100, running time: 26.0865, accuracy: 0.670113

Second, threshold: 50, running time: 39.692, accuracy: 0.813421

Third, using SGD, running for 20 iters, time: 29.548, accuracy 0.812102

As we can see here, if we set the threshold to a very small value, then the program would take more time to converge, and got a better performances.

5 Question 5

No, we can not guarantee that arriving to the same W and H everytime. This is because for the matrix factorization task, we just random initilization and then doing the updates in every iteration. What's more, the iteration number was not sure to guarantee that the algorithm would converge to the same point.

6 Question 6

The (b) is not valid for DSGD, because it does not satisfy the requirements of the strata for the DSGD algorithms. In the DSGD algorithms, the set of blocks, the stratum, should not overlap. But in (b), the blocks are overlapped.

7 Question 7

The (d) strata should only achieve a suboptimal from the point of view of parallelism. This is because in DSGD, we need to group the data. We should repeat that untill all blocks are covered. Otherwise, we are not taking the whole dataset. Therefore, losing a block of data would cause a suboptimal of the dataset from the point of view of parallelism.

8 Question 8

I did not receive helps from others, and did not give direct helps to others.