Report

R08922160
山下夏輝

Q1 : Describe your MF with BCE (e.g. parameters, loss function, negative sample method and MAP score on Kaggle public scoreboard)

My BCE loss function

- get the probability of the item i and item j. i has interacted by users already and j has not done.
- get the binary cross entropy loss of each items.
- add them as a loss score

My Kaggle scores

score 1

0.00148

- this model
  - loss: BCE
  - optimizer: SGD
  - learning rate: 0.0001
  - weight decay: 0.0001
  - momentum: 0.9
  - number of factor: 124
  - number of negative sample: 32
  - number of epoch: 19
  - train data: validation data = 9：0

score 2

0.00162

- this model
  - loss: BCE
  - optimizer: SGD

    - learning rate: 0.0001
    - weight decay: 0.0001
    - momentum: 0.9
    - number of factor: 124
    - number of negative sample: 64
    - number of epoch: 19
    - train data: validation data = 9：0

My experiment

My models which experimented for getting higher scores are below but the experiment results show that it is difficult to get higher scores.

- model 1
    - loss: BCE
    - optimizer: SGD
    - learning rate: 0.0001
    - weight decay: 0.0001
    - momentum: 0.9
    - number of factor: 32
    - number of negative sample: 32
    - number of epoch: 130
    - train data: validation data = 8 : 1
- experiment result
  (the more blue the color is, the more BETTER the value is in metrics)

0	1.3859941	0.001572909
1	1.3859935	0.001572909
2	1.3859935	0.001572909
3	1.3859934	0.001572909
4	1.3859937	0.001572909
5	1.3859937	0.001572909
6	1.3859937	0.001572909
7	1.3859942	0.001572909
8	1.3859938	0.001572909
9	1.3859935	0.001572909
10	1.3859937	0.001572909
11	1.3859932	0.001572909
12	1.385994	0.001572909
13	1.385994	0.001572909
14	1.385994	0.001572909
15	1.3859934	0.001572909
16	1.3859941	0.001572909
17	1.3859938	0.001572909
18	1.3859935	0.001572909
19	1.385994	0.001572909
20	1.3859937	0.001572909
21	1.3859931	0.001572909
22	1.3859937	0.001572909
23	1.385994	0.001572909
24	1.3859942	0.001572909
25	1.3859936	0.001572909
26	1.3859935	0.001572909
27	1.3859937	0.001572909
28	1.3859937	0.001572909
29	1.3859938	0.001572909

The MAP does not change among each epoch. So I adjusted the learning rate from0.0001 to 0.05

- model 2
  - loss: BCE
  - optimizer: SGD
  - learning rate: 0.05
  - weight decay: 0.0001
  - momentum: 0.9
  - number of factor: 32
  - number of negative sample: 32
  - number of epoch: 30
  - train data: validation data = 8 : 1
- experiment result
  (the more blue the color is, the more BETTER the value is in metrics)

epc	loss	MAP
0	1.3859928	0.001131002
1	1.3859924	0.001130983
2	1.3859925	0.001130983
3	1.3859925	0.001130983
4	1.3859931	0.001130975
5	1.3859925	0.001130975
6	1.3859925	0.001130975
7	1.3859919	0.001130969
8	1.3859922	0.001130982
9	1.3859923	0.001130982
10	1.3859918	0.001130907
11	1.3859923	0.001130895
12	1.3859923	0.001130895
13	1.385992	0.001130899
14	1.3859922	0.001130913
15	1.385992	0.001130893
16	1.3859923	0.001130893
17	1.3859925	0.001130893
18	1.3859928	0.001130893
19	1.3859925	0.001130899
20	1.3859918	0.001130899
21	1.385992	0.001130899
22	1.3859923	0.001130891
23	1.3859923	0.001130936
24	1.3859928	0.00113099
25	1.385992	0.00113099
26	1.3859916	0.00113099
27	1.3859916	0.00113099
28	1.3859915	0.00113099
29	1.3859917	0.001131

The MAP changes a little among each epoch but does not increase. So I adjusted the learning rate again from 0.05 to 1.

- model 3
  - loss: BCE
  - optimizer: SGD
  - learning rate: 1
  - weight decay: 0.0001
  - momentum: 0.9
  - number of factor: 32
  - number of negative sample: 32
  - number of epoch: 30
  - train data: validation data = 8 : 1
- experiment result
  (the more blue the color is, the more BETTER the value is in metrics)

epo	loss	MAP
0	1.3859932	0.001390487
1	1.385994	0.001390511
2	1.3859935	0.001390506
3	1.385993	0.001390095
4	1.3859924	0.001390486
5	1.3859923	0.001390606
6	1.3859922	0.001390546
7	1.3859917	0.001390644
8	1.3859913	0.001390651
9	1.3859906	0.001390804
10	1.3859904	0.001391068
11	1.3859899	0.00139107
12	1.3859885	0.001391102
13	1.385988	0.001391175
14	1.385988	0.001390172
15	1.3859873	0.00138965
16	1.3859862	0.001389644
17	1.3859859	0.001389581
18	1.3859837	0.001389169
19	1.385984	0.001389597
20	1.3859833	0.001388837
21	1.3859825	0.001388613
22	1.3859817	0.001388097
23	1.3859813	0.00138828
24	1.3859804	0.001388262
25	1.3859794	0.001383933
26	1.3859785	0.001383754
27	1.3859774	0.001384075
28	1.3859766	0.001384166
29	1.3859761	0.001384308

The better the loss score is, the worse MAP score goes.

Q2: Describe your MF with BPR (e.g.parameters, lossfunction, negative sample method and MAP score on Kaggle public scoreboard)

My BPR loss function

- according to the slide of assignment specification, it is implemented
- assume the probability of item i is higher than item j.
- the probability of item i is minused from item j
- sigmoid it
- log it
- sum every of them

My Kaggle scores

How to select the result file uploading onto Kaggle

- train model with train data set : validation set = 8 : 1 using the same parameter as the model trained with all data
- get MAP with validation set
- find what the parameter get the best MAP
- get the result file with the model trained with all data using the best parameter found the last step
- upload it on kaggle

score 1

0.04842

- model
  - loss: BPR
  - optimizer: SGD
  - learning rate: 0.0001
  - weight decay: 0.0001
  - momentum: 0.9
  - number of factor: 128
  - number of negative sample: 64
  - number of epoch: 42
  - train data: validation data = 9 : 0

score 2

0.04826

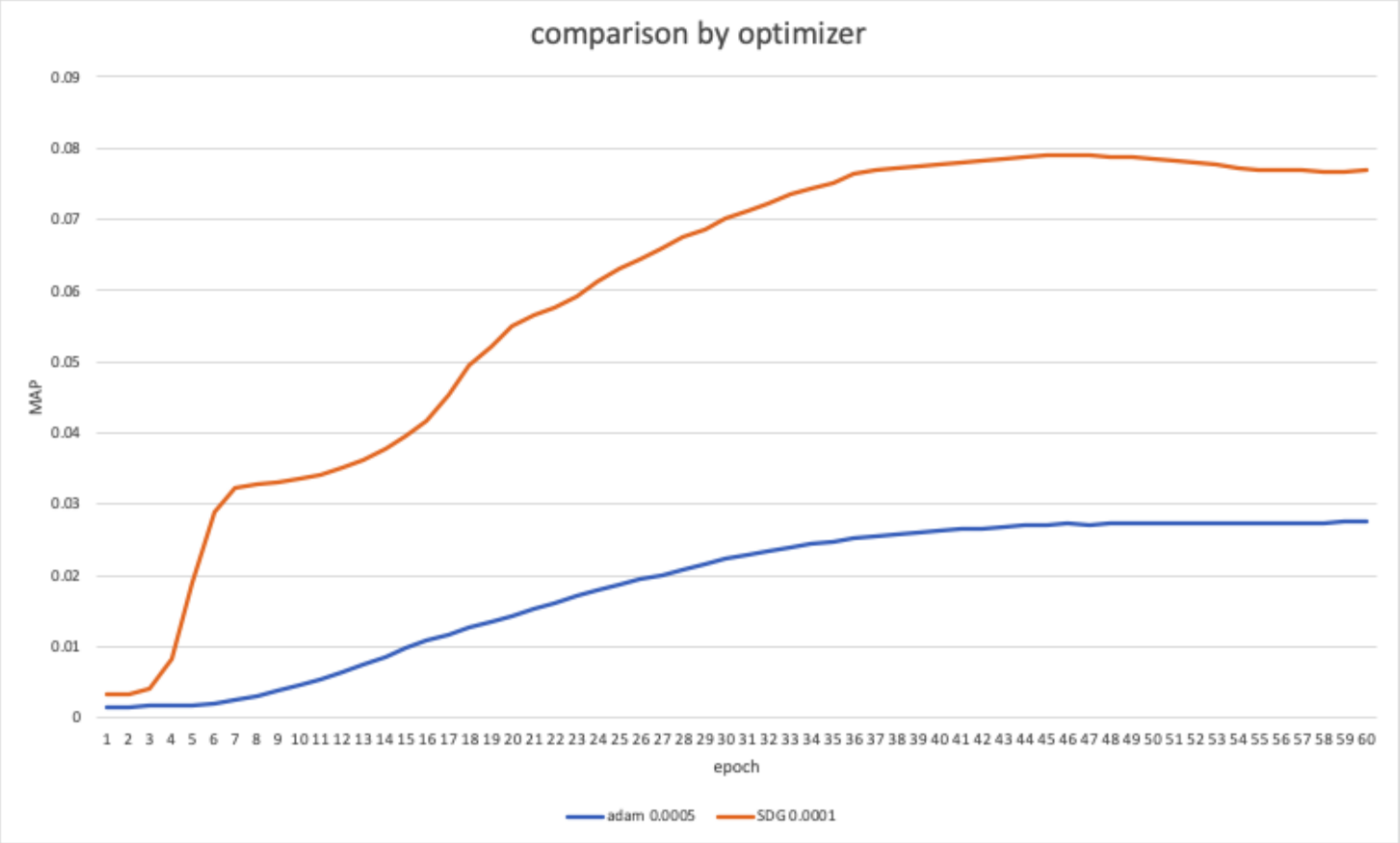
- model
  - loss: BPR
  - optimizer: SGD
  - learning rate: 0.0001
  - weight decay: 0.0001
  - momentum: 0.9
  - number of factor: 128
  - number of negative sample: 64
  - number of epoch: 41
  - train data: validation data = 9 : 0

My experiment

changed optimizer and trained model. Optimizers used SDG and Adam and changed the number of negative sample. The model with Adam did not increase much more than SDG one. The detail of the

parameters is as below.

- model
  - loss: BPR
  - optimizer: SGD
  - learning rate: 0.0001
  - weight decay: 0.0001
  - momentum: 0.9
  - number of factor: 128
  - number of negative sample: 64
  - number of epoch: 60
  - train data: validation data = 8 : 1
  - model
    - loss: BPR
    - optimizer: Adam
    - learning rate: 0.0005
    - weight decay: 0.0001
    - number of factor: 128
    - number of negative sample: positive : negative = 1 : 1
    - number of epoch: 60
    - train data: validation data = 8 : 1



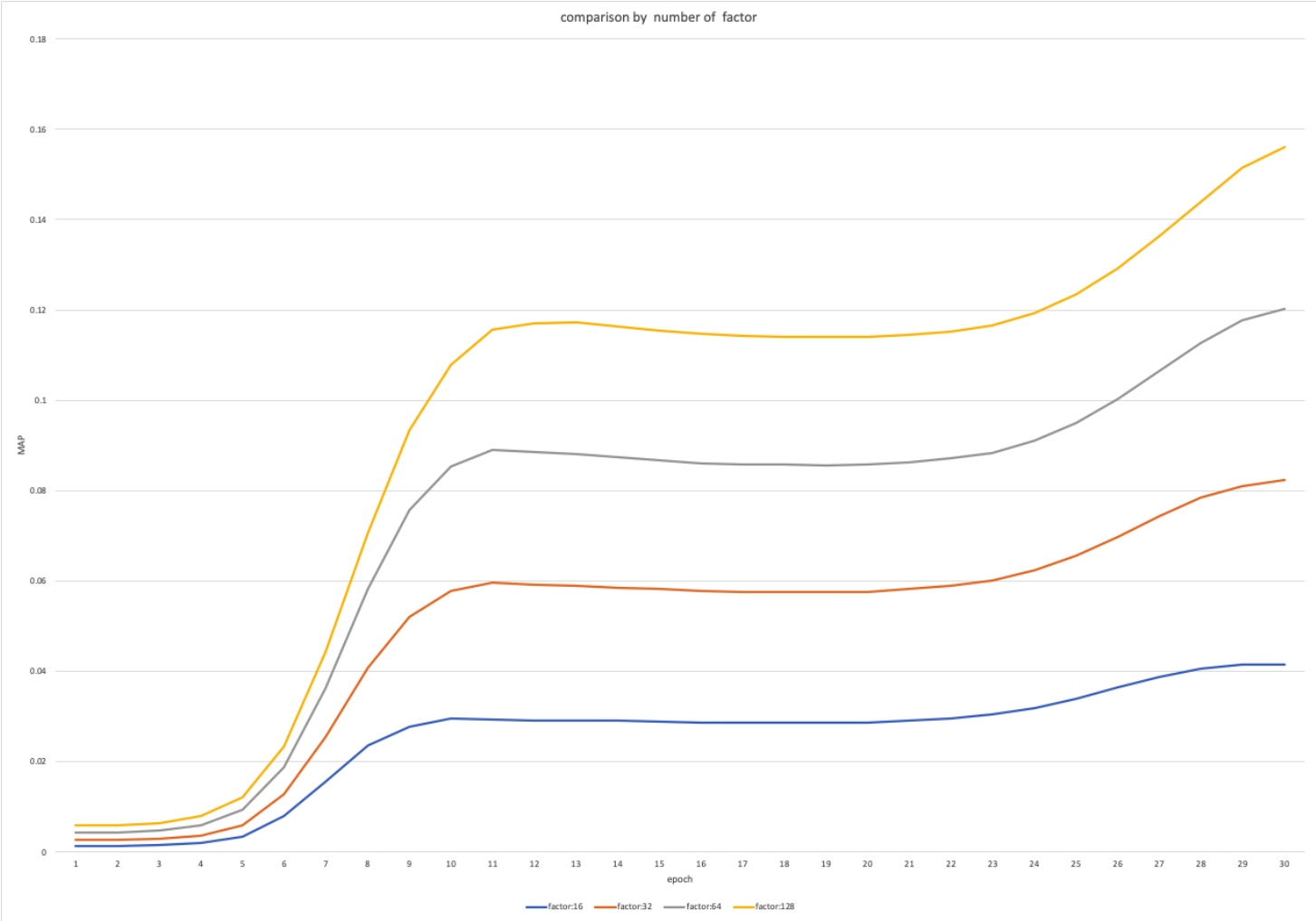
Q3: Compare your results of Q1 and Q2. Do you think the BPR loss benefits the performance? If do, write some reasons of why BPR works well; If not, write some reasons of why BPR fails.

I think BPR loss is much better for performance than BCE loss. This reason is that the premise of assumption of BPR benefits performance.

- BPR: optimizing parameters based on a pair of instances.

Because of it, the parameters are optimized for the relationship of each items and between a user and the items and it is better to represent for the implicit preference of the users for the items which users has not interacted.

Q4: Plot the MAP curve on testing data(Kaggle) for hidden factors d = 16, 32, 64, 128 and describe your finding.



As the the graph shows, the more the number of factor is, the better the performance would be. As you can see, MAP increases by a certain degree when the number of the factor is doubled.

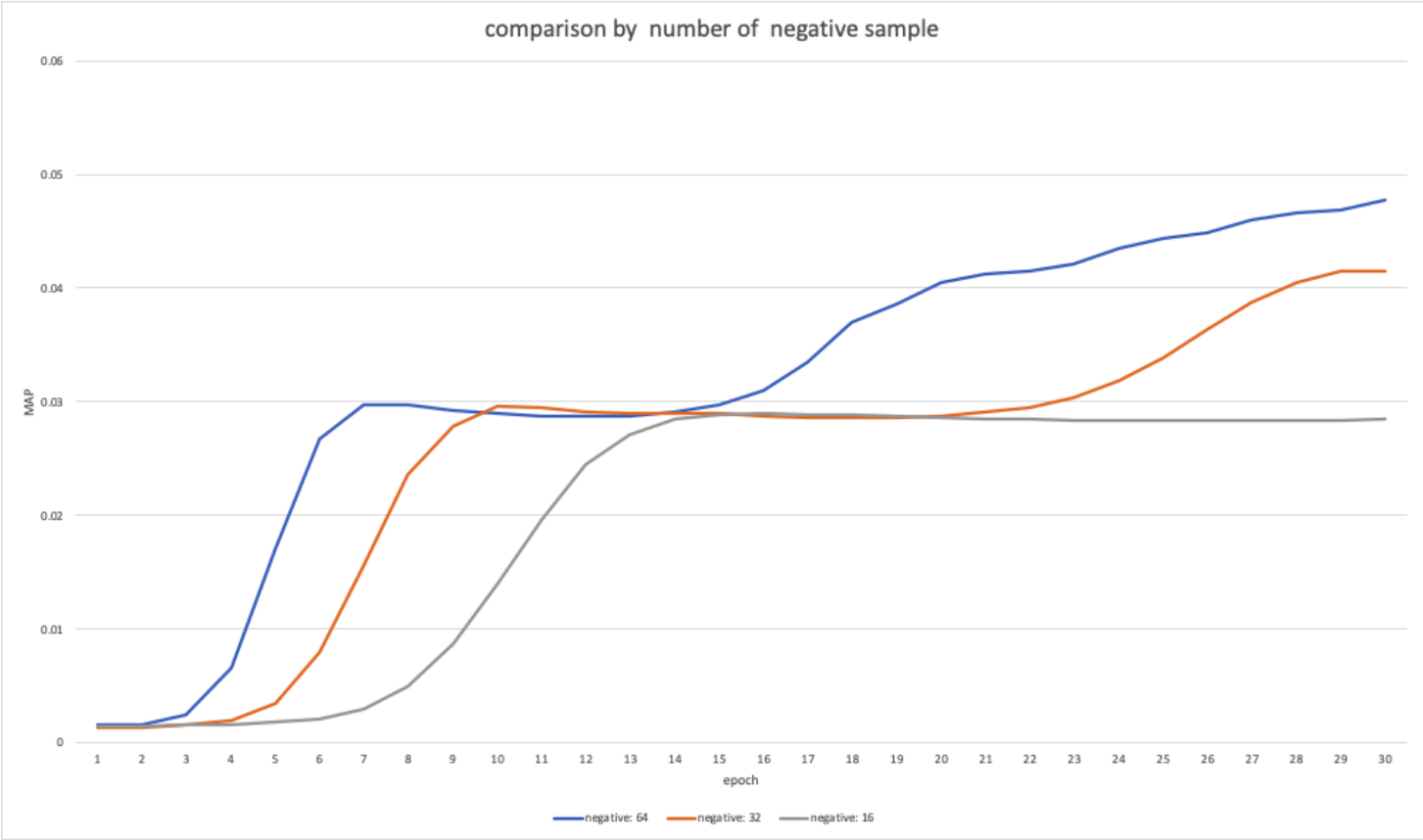
This also could mean that the number of factor (16-128) is still not enough to get the maximized performance with this dataset. So we can try some experiments with more larger number of factors and make sure how many factor can maximize the performance with this dataset.

- model
  - loss: BPR
  - optimizer: SGD
  - learning rate: 0.0001
  - weight decay: 0.0001
  - momentum: 0.9
  - number of factor: 16, 32, 64, 128
  - number of negative sample: 32
  - number of epoch: 30
  - train data: validation data = 8 : 1

comment

題目裡有些到 “testing data（kaggle）” 但，一直沒有過 base line 所以沒有充分的 result data（MAP on kaggle），所以使用我自己切的 validation set 來做實驗畫圖了。
能力不足而沒有超過 base line，只能如此了。助教，十分不好意思。

Q5: Change the ratio between positive and negative pairs, compare the results and discuss your finding.



As the the graph shows, the more the number of factor is, the faster the performance would grow up. And also the more the number of factor is, the earlier the second curve of performance growth would come. In addition to it, the more the number of factor is, the better the performance would be in the result.
This graph imply not only positive samples but also the number of negative samples influence the performance. In the other word, the number of the combination of positive sample and negative sample influence the performance.

- model
  - loss: BPR
  - optimizer: SGD
  - learning rate: 0.0001
  - weight decay: 0.0001

- momentum: 0.9
- number of factor: 128
- number of negative sample: 16, 32, 64
- number of epoch: 30
- train data: validation data = 8 : 1