B06902125 黃柏瑋 WM HW7 (Prog 2)

Q1 :

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

首先，MF 的架構為一個 user embedding 和 item embedding，當輸入一個 user_id 和一個 item_id 時，抽出相對應的 embedding vector 內積後回傳 logits，其中 item_id 可能是 positive 也可能是 negative。

接著，說明訓練過程。這題選用的 loss 為 binary cross entropy，公式如下，其中 $y_i = 1$代表 positive 、 $y_i = 0$ 代表 negative，而 $\hat{y}_i$ 為來自 MF 的 logits：

$$BCE(\hat{y}, y) = -\sum_i y_i log(\sigma(\hat{y}_i)) - (1 - y_i) log(1 - \sigma(\hat{y}_i))$$

為了方便和 BPR 比較，訓練時使用的參數和 Q2 相同：

PARAMETER	VALUE
optimizer	Adam
MF hidden size	128
batch size	4096
learning rate	1e-3
regularization weight	1e-2
epoch	100
negative method	uniform distribution
ratio (pos : neg)	1:1

最後，為 valid mAP (valid 切法會在 Q+ 補充)和 Kaggle public mAP 的結果：

	VALID	PUBLIC
BCE	0.05107	0.05240

Q2 :

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

首先， MF 的架構和 Q1 一樣，只是這裡的輸入是一個 user_id 和兩個 item_id，其中一個為 positive，另一個為 negative，而模型在抽完 embedding 內積後會回傳兩個 logits，分別來自 positive item 和 negative item。

接著，說明訓練過程。這題選用的 loss 為 Bayesian personalized ranking，公式如下，其中$+$為 positive item 、$-$為 negative item，而 $\hat{y}$ 為來自 MF 的 logits：

$$BPR(\hat{y}_+, \hat{y}_-) = -\sum_i ln\sigma(\hat{y}_{+i} - \hat{y}_{-i})$$

為了方便和 BCE 比較，訓練時使用的參數和 Q1 相同：

PARAMETER	VALUE
optimizer	Adam
MF hidden size	128
batch size	4096
learning rate	1e-3
regularization weight	1e-2
epoch	100
negative method	uniform distribution
ratio (pos : neg)	1:1

最後，為 valid mAP (valid 切法會在 Q+ 補充)和 Kaggle public mAP 的結果：

	VALID	PUBLIC
BPR	0.05361	0.05380

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.

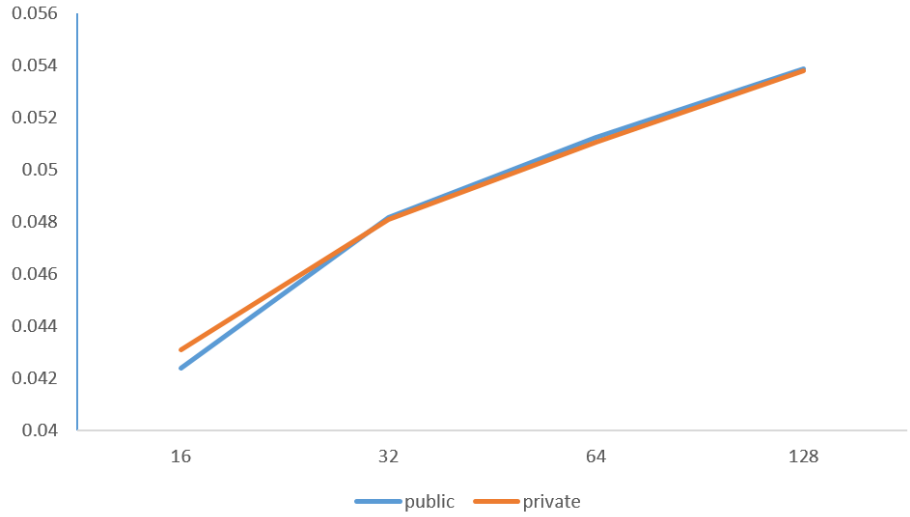
	VALID	PUBLIC
BCE	0.05107	0.05240
BPR	0.05361	0.05380

由上表可以發現，無論是 valid score 還是 Kaggle public score，BPR 的表現都優於 BCE。主要可能是因為 BPR 關注兩個 item 之間的相對關係，和「排序」的精神較有連結，而 BCE 著重於 item 與 label 之間的精確度，並假設每個 item 之間是 independent，與實際情形有些不同，導致推薦的效果稍弱一些。

Q4 :

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

本題使用的模型和 Q2 一樣，只是改動了 MF hidden size 進行比較，mAP曲線如下：



由圖中可知，當 hidden size 變大時，無論是 public 還是 private 的 mAP 都會跟著上升，代表 hidden size 越大，所能保存的訊息越完整，進而達到較好的成果。

Q5：

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

以下，將用 Q2 的模型為基準，僅改動 ratio 進行比較：

RATIO (POS:NEG)	VALID	PUBLIC	PRIVATE
1:1	0.05361	0.05380	0.05386
1:2	0.05449	0.05517	0.05351
1:3	0.05442	0.05644	0.05388
1:4	0.05372	0.05587	0.05370

由上表可知，1:3 的效果最好，1:4 時開始出現 overfitting 的跡象。

然而，由於 BPR pairwise training 的設計，將 negative sample 增加為 3 倍的效果可能和 negative sample 數目不動、epoch 增為 3 倍的效果差不多，都是將訓練過程中所見的 pair 總數增為原本的三倍，因此跑了以下實驗驗證：

	LOSS	VALID
epoch = 100, ratio = 1:3	0.04685	0.05442
epoch = 300, ratio = 1:1	0.04663	0.05443

可以見得，兩者無論 training loss 或 valid mAP 都相當接近，代表增加 negative sample 和增加 epoch 可以達到差不多的效果。而這項發現也說明 epoch 100 對原本的參數而言有點太少，從不同 ratio 的實驗表格來看，這組參數對應的最佳 epoch 數很可能位於 300 至 400 之間。

Q+：

在本次作業中，我嘗試了兩種不同的 validation 切法：1. 隨機 2. 後切

首先，先分析這兩種切法和預期的結果，其中值得注意的是，本次作業的 recommendation system 希望能根據 user 先前互動過的 item 推薦 user 可能感興趣但還沒互動過的 item，因此和時間軸高度相關：

1. 隨機：

   沒有將時間軸的觀念考慮進去，可能沒辦法符合這個 recommendation system 的需求，而隨機切法可能會產生 valid data 比 train data 還早發生的情況，也就是拿未來的資料訓練預測過去資料，因此得出的 valid score 應該會偏高。

2. 後切：

   有考慮時間軸的觀念，比較符合這個 recommendation system 的需求，得出的 valid score 應該會較貼近 Kaggle score。

接著，以 Q5 為討論對象，觀察數據表現：

	VALID 1	VALID 2	PUBLIC	PRIVATE
BPR / 1:1	0.1457	0.05361	0.05380	0.05386
BPR / 1:2	0.1509	0.05449	0.05517	0.05351
BPR / 1:3	0.1489	0.05442	0.05644	0.05388
BPR / 1:4	0.1480	0.05372	0.05587	0.05370

不難發現，隨機切的 valid score 確實比 Kaggle public score 高上許多，而後切的 valid score 則相當接近 Kaggle public score，代表上述的推測是合理的，因此 report 中所有的 valid score 都採用後切的方法。