

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

我選擇使用 hidden factors = 16,32,64,128 的 MF，loss function 選擇 BCE 不加 normalization

$$\text{BCELoss} = - \left\{ \sum_{(u,i) \in D^+} \log(\sigma(u^T i)) + \sum_{(u,j) \in D^-} \log(1 - \sigma(u^T j)) \right\}$$

negative sample method 則為隨機選擇一位使用者，然後再隨機選擇一個物品，如果(user,item)不在 training positive 的 set 中的話，就將他放進去計算 loss。在實作上也試著調整 negative sampling rate(#negative / #positive)，發現調整至約 1×10^{-4} 會收斂至較好的分數。Learning rate 的調整則為一開始先用 0.1 訓練至收斂後，讀取最好的 model 再用更低的 learning rate 進行訓練，最後用到 $1e-4$ 的 learning rate 訓練到最佳的結果，在訓練的過程中，因為其他 hidden factor 的 validation score 都沒有 64 好，所以沒有上傳。此方法最佳結果為: hidden factor = 64, public MAP=0.04520。

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

我使用的 BPRLoss 如下，一樣使用 16、32、64、128 的 hidden factor 進行訓練，並調控適當的 weight penalization term 避免過擬合的現象。Learning rate 的調控方式如同 Q1。

$$\text{BPRLoss} = - \sum_{(u,i,j) \in D} \log(\sigma(u^T i - u^T j)) + \lambda ||\theta||^2$$

Negative sample 的方法為從 positive training 挑選一對(u,i)，然後再隨機選取一個 item 的數字 j，如果該對(u,j)不在 positive training 的話，就把它加進去當成 (u,i,j)，調整 negative sampling rate 的方法就是在調控畫底線的區域成功進行完全幾次，而發現用 1:1 的結果就可以過 strong baseline。最後得到 public 最好的模型為，hidden factor=128、 $\lambda = 10^{-6}$ ，learning rate 從 0.1 調整至 0.01，在 kaggle 的 public 分數為 0.05163。

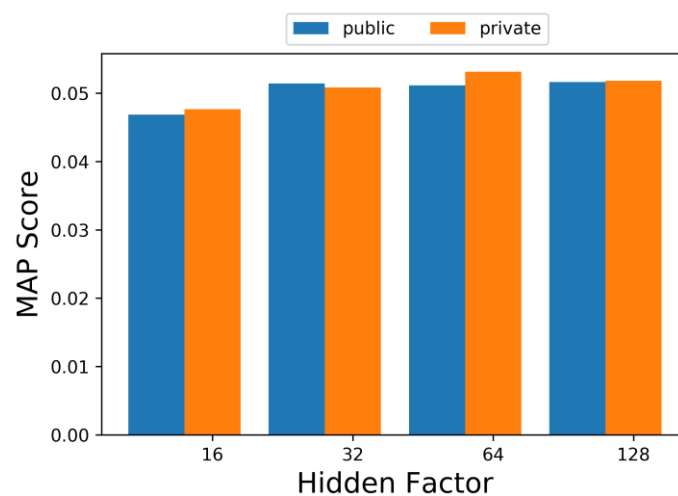
Q3 : Compare your results of Q1 and Q2. Do you think the BPR loss benefits the performance? If do, write some reasons of why BPR works well; If not, write some reasons of why BPR fails.

跟據 Q1 跟 Q2 的結果，可以推論 BPR Loss 對於訓練的效果較佳，原因在於：使用 BCE 的 Loss Function，會讓每個 negative sample 和 positive sample 具有相同的 effect，對 weight 具有同性質的影響程度，而且因為 positive sample 和 negative sample 是分開取的，所有訓練過程並沒有考慮到同個 user 對不同 item

的影響。然而，當使用 BPR Loss 時，它隱含的意義為令同個使用者對於 positive item 和 negative item 偏好程度的差距越大越好，這樣就有考慮到同個使用者對於不同 item 偏好程度的狀況。而且，negative sample 並不一定代表它就是使用者不喜好的 item，可能只是還沒看過而已，所以用差距的方式來計算 loss 會顯得比較合乎邏輯。

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

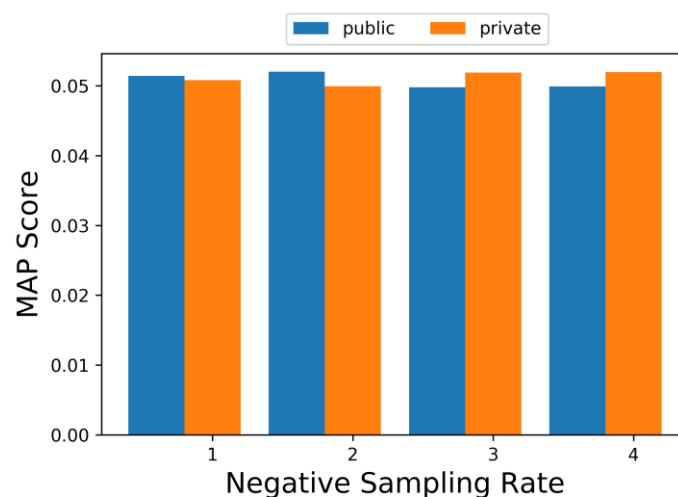
不同 hidden factor 所得到 kaggle public 和 private score 結果如下：



從圖中可以觀察到 hidden factors=16 的分數明顯低於其他三者，而其他三者的分數大約一致，可以判斷在 hidden factors ≥ 32 的時候已經達到收斂的狀態。

(Bonus 10%) Q5 : Change the ratio between positive and negative pairs, compare the results and discuss your finding.

因為在 Q4 觀察到 hidden factor=32 就已經收斂，所以以這個參數進一步實驗不同的 negative sampling rate 能否提升分數。結果如下：



在 public 的話，由 negative sampling rate=1 的分數最佳，在 private 的話，由 negative sampling rate=4 的分數最佳，但是並沒有發現顯著的差異，推測 hidden factors 跟 weight penalize 對於結果的影響比較大。