

P1: 請比較有無 normalize(在 rating 上)的差別。並說明如何 normalize. (1%)

Collaborator: 無

training 過程 20 個 epoch, batch size 為 128

normalize: 把整個 rating 做 normalization(個別使用者的 bias 已經被考慮, 假設各自的 std 差別不大)

無 normalize 的結果為 public 0.89052, private 0.89207

有 normalize 的結果為 public 0.86614, private 0.86611

normalize 結果明顯較好, rate 的大小關係有意義, 符合理論

P2: 比較不同的 latent dimension 的結果(1%)

Collaborator: b04901018 游昇融

latent dimension	public	private
10	0.86618	0.86600
15	0.86317	0.86448
20	0.86318	0.86246
24	0.86614	0.86611
30	0.87003	0.87126
35	0.87249	0.87243

在此數量附近, 20 左右表現最好, 並且在 latent dimension 較大的時候表現相對較差

P3: 比較有無 bias 的結果。(1%)

Collaborator: 無

training 過程 20 個 epoch, batch size 為 128

無 bias 的結果為 public 0.87406, private 0.87527

有 bias 的結果為 public 0.86614, private 0.86611

可以看出 bias 確實是有改善的, 符合理論

P4: 請試著用 DNN(投影片 p.28)來解決這個問題, 並且說明實做的方法(方法不限)。並比較 MF 和 NN 的結果, 討論結果的差異。(1%)

Collaborator: 無

參考 sample code 的方法, 把 user 跟 movie 的 input 接在一起, 然後過兩層的 Dense

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 1)	0	
input_2 (InputLayer)	(None, 1)	0	
embedding_1 (Embedding)	(None, 1, 24)	144984	input_1[0][0]
embedding_2 (Embedding)	(None, 1, 24)	93216	input_2[0][0]
flatten_1 (Flatten)	(None, 24)	0	embedding_1[0][0]
flatten_2 (Flatten)	(None, 24)	0	embedding_2[0][0]
dot_1 (Dot)	(None, 1)	0	flatten_1[0][0] flatten_2[0][0]
Total params: 238,200			
Trainable params: 238,200			
Non-trainable params: 0			

DNN: public 0.88290, private 0.89027

MF: public 0.86614, private 0.86611

結果而言，MF 效果明顯較好，除了 bias 的差別之外，這個 case 考慮對應關係，MF 理論上也比 DNN 更適合

P5: 請試著將 movie 的 embedding 用 tsne 降維後，將 movie category 當作 label 來作圖(如投影片 p.29)。(1%)

Collaborator: b04901018 游昇融

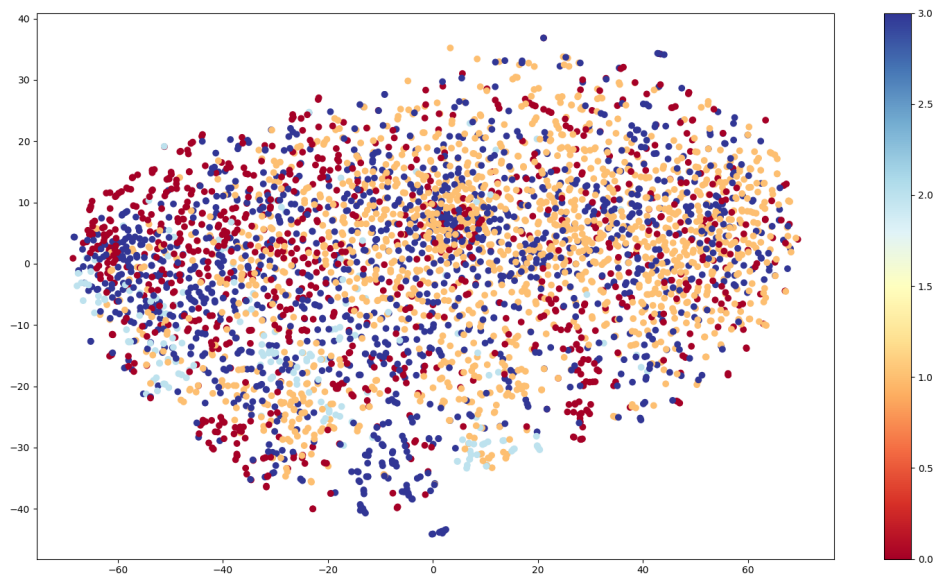
label 方法:

'Thriller' 'Horror' 'Crime' 第 1 類

'Drama' 'Musical' 第 2 類

'Animation' 'Children's' 第 3 類

其他 第 4 類



圖中雖然蠻亂的，有些地方一塊一塊還是看的出來

BONUS: 試著使用除了 rating 以外的 feature, 並說明你的作法和結果，結果好壞不會影響評分。(1%)

Collaborator: b04901015 傅子興

這裡延續第五題，把電影分成四大類下去做 embedding

'Thriller' 'Horror' 'Crime' 第 1 類

'Drama' 'Musical' 第 2 類

'Animation' 'Children's' 第 3 類

其他 第 4 類

而 user 維持用 id

結果

public: 1.02671

private: 1.02629

離 simple baseline 有點距離，大致分類效果很差，推估人類的分類方式可能和實際評分在考慮的 dimension 有差距