

1. (1%)請比較有無 `normalize(rating)` 的差別。並說明如何 `normalize`。

(collaborator:)

無 `normalize`: $\sqrt{(private^2 + public^2)/2} = \sqrt{(0.84702^2 + 0.84827^2)/2} = 0.84765$

有 `normalize`: $\sqrt{(private^2 + public^2)/2} = \sqrt{(0.86003^2 + 0.85992^2)/2} = 0.85998$

`normalize` 方法: 做 *z-normalization*

$$mean = (\sum_{i=1}^n rating_i) / n$$

$$std = \sqrt{\left(\sum_{i=1}^n |rating_i - mean|^2\right) / n}$$

$$rating_i = \frac{(rating_i - mean)}{std}, \quad \forall i = 1, 2, \dots, n \text{ for training datas}$$

$$prediction_i = prediction_i \times std + mean, \quad \forall i = 1, 2, \dots, m \text{ for testing datas}$$

無 `normalize` 比有 `normalize` 好，但我覺得是因為有 `normalize` 會有浮點數計算上的誤差導致。在 *training* 過程中，有 `normalize` 會比無 `normalize` 讓 *loss function* 下降更快。

2. (1%)比較不同的 *latent dimension* 的結果。

(collaborator:)

$$RMSE = \sqrt{(private^2 + public^2)/2}$$

<i>latent dimension</i>	<i>private</i>	<i>public</i>	<i>RMSE</i>
4	0.86990	0.87090	0.87040
8	0.86093	0.86193	0.86143
16	0.85434	0.85257	0.85345
32	0.85217	0.85277	0.85247
64	0.84829	0.84817	0.84823
128	0.84702	0.84827	0.84765

Latent dimension 越大, *RMSE* 越小, *performance* 也越好, 且所需要的 *epochs*

也越少，但是執行一個 *epoch* 的時間也就越多。

3. (1%)比較有無 bias 的結果。

(collaborator:)

$$RMSE = \sqrt{(private^2 + public^2)/2}$$

$$\text{無 bias: } \sqrt{(private^2 + public^2)/2} = \sqrt{(0.84849^2 + 0.84681^2)/2} = 0.84765$$

$$\text{有 bias: } \sqrt{(private^2 + public^2)/2} = \sqrt{(0.84702^2 + 0.84827^2)/2} = 0.84765$$

無 *bias* 和有 *bias* 兩者的 *RMSE* 相差不多，*performance* 幾乎一樣好。

4. (1%)請試著用 DNN 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。

(collaborator:)

DNN 實做方法：

將 user embedding matrix 和 movie embedding matrix 兩個 matrix concatenate 在一起，過三層 dense layer 得到 rating。

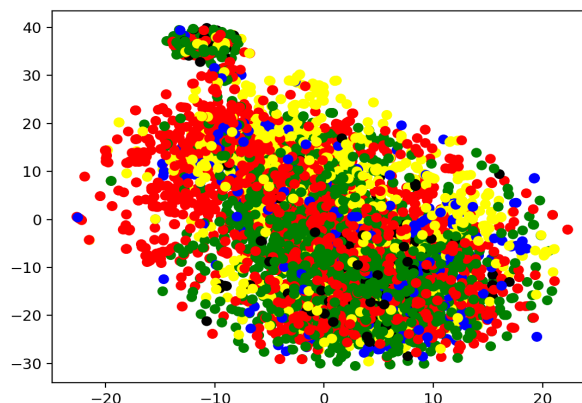
$$\text{MF: } \sqrt{(private^2 + public^2)/2} = \sqrt{(0.84702^2 + 0.84827^2)/2} = 0.84765$$

$$\text{NN: } \sqrt{(private^2 + public^2)/2} = \sqrt{(0.86382^2 + 0.86287^2)/2} = 0.86335$$

MF 的 *RMSE* 比 NN 的還低，MF 的 *performance* 比 NN 的好。

5. (1%)請試著將 movie 的 embedding 用 tsne 降維後，將 movie category 當作 label 來作圖。

(collaborator:)



Animation, Children's, Comedy, Adventure, Fantasy 分為一類，用紅色來代表

Sci-Fi, Mystery, Film-Noir 分為一類，用藍色來代表

Romance, Drama, Musical 分為一類，用綠色來代表

Action, Crime, Thriller, Horror, War 分為一類，用黃色來代表

Documentary, Western 分為一類，用黑色來代表

6. (BONUS)(1%)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。

(collaborator:)

Step1.將 age normalize

Step2.將 user 的 occupations 和 movie 的 genres 都做 one-hot encoding

Step3.將 user 的 embedding matrix、movie 的 embedding matrix、user 的 age 和 occupations 及 movie 的 genres concatenate 在一起

Step4.concatenate 完後，進入三層 dense layer

$$\text{RMSE} = \sqrt{(\text{private}^2 + \text{public}^2)/2} = \sqrt{(0.85639^2 + 0.85673^2)/2} = 0.85656$$

這個 case RMSE 比之前 NN 版本的 RMSE 低，performance 也比較好；但是，比之前 MF 版本的 RMSE 高，performance 相較之下比較差。