
Machine Learning HW6

TA Hour

TAs
ntu.mlta@gmail.com

Outline

- Issues
- Report problem
- Sample code
 - Rating normalization
 - Model

Issues - 1 - Movie, User data

- movies.csv, users.csv: 原來是以 , 做分隔, 5/27號新上傳的data是以 :: 做為分隔
- movies.csv: traing data中有些movie是沒有在movie.csv中的, 原來的dataset就沒有這些值, 這純屬正常現象, 請同學不用擔心

report problem

Report problem

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

說明：

取training data rating的mean以及standard deviation, 再把training data的rating做normalization再用normalized過後的rating做為training的target。

在testing的時候則 $pred_{test} = pred_{test}^* \times train_{std} + train_{mean}$

Report problem

(1%)比較不同的latent dimension的結果

說明:

放上實驗數據及簡單說明即可

Report problem

(1%)比較有無bias的結果

說明:

$$r_{i,j} = U_i \cdot V_j + b_i^{user} + b_j^{movie}$$

每個user可能都會有自己評分的傾向, 像是傾向於把每部電影都評得很高分或者很低分; 同樣的電影也會有這樣的趨勢。

Report problem

1. (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。

說明：

這問題可以將user embedding以及movie embedding concatenate在一起再過DNN得出rating, 或者將user embedding以及item embedding分別通過兩個DNN得出movie vector以及item vector再去做內積得出rating。

output可以把這個問題視為regression問題或者1-5的分類問題。

DNN如果參數調的夠好是可以比單純MF有更好的表現。

Report problem

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

說明：

這題希望同學可以取MF的movie embedding，作圖再簡單分析為何會有這樣子的結果。由於這個dataset的movie類別分得很細，如果用原始的類別是畫不出有區分效果的圖的。同學可以自由的把像是'Thriller','Horror','Crime'這幾個相似的類別視為相同的大類別，'Drama','Musical'視為另外一個大類別，再用這些作為tag去作圖。

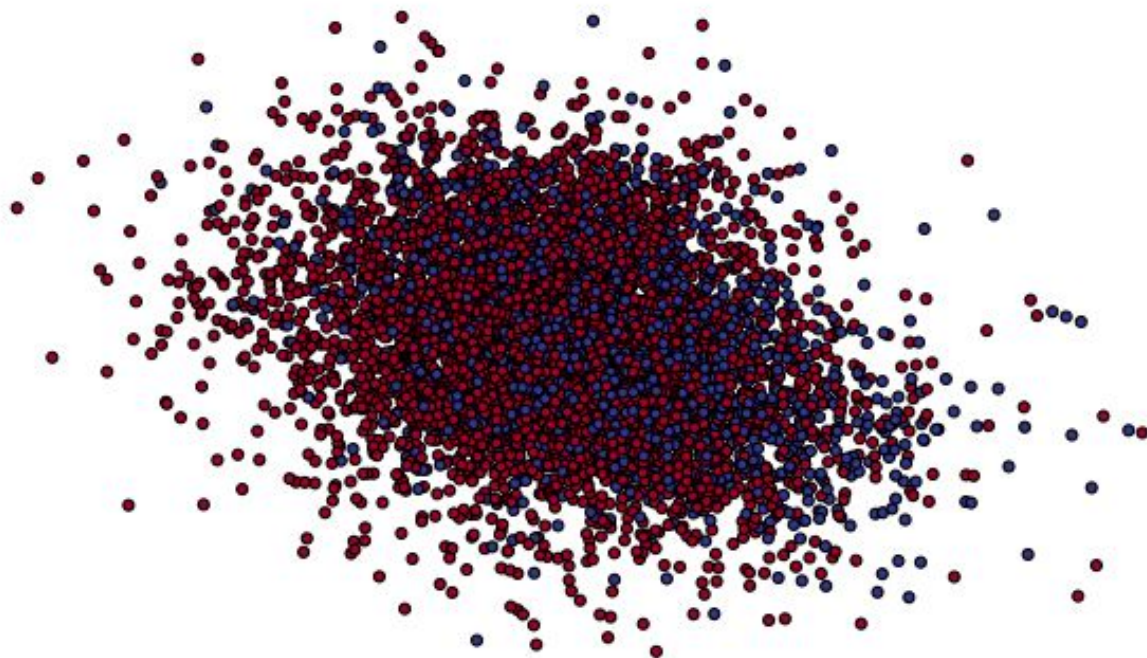
Report problem

(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。

說明:

不用使用這些額外的feature其實就可以輕鬆的過strong baseline了, 這題同學可以只用user data或者只用movie data來作為額外的feature, 可能有些feature是有用的有些是沒用的, 同學可以試著去分析看看, 像是用上題的方法對每個feature都當作tag去作圖, 再看看哪些feature會有比較明顯的區分效果

以性別來做tag作圖



sample code

Model - MF

```
def get_model(n_users, n_items, latent_dim=6666):
    user_input = Input(shape=[1])
    item_input = Input(shape=[1])
    user_vec = Embedding(n_users, latent_dim, embeddings_initializer='random_normal')(user_input)
    user_vec = Flatten()(user_vec)
    item_vec = Embedding(n_items, latent_dim, embeddings_initializer='random_normal')(item_input)
    item_vec = Flatten()(item_vec)
    user_bias = Embedding(n_users, 1, embeddings_initializer='zeros')(user_input)
    user_bias = Flatten()(user_bias)
    item_bias = Embedding(n_items, 1, embeddings_initializer='zeros')(item_input)
    item_bias = Flatten()(item_bias)
    r_hat = Dot(axes=1)([user_vec, item_vec])
    r_hat = Add()([r_hat, user_bias, item_bias])
    model = keras.models.Model([user_input, item_input], r_hat)
    model.compile(loss='mse', optimizer='sgd')
    return model
```

Model - DNN

```
def nn_model(n_users, n_items, latent_dim=7777):  
    user_input = Input(shape=[1])  
    item_input = Input(shape=[1])  
    user_vec = Embedding(n_users, latent_dim, embeddings_initializer='random_normal')(user_input)  
    user_vec = Flatten()(user_vec)  
    item_vec = Embedding(n_items, latent_dim, embeddings_initializer='random_normal')(item_input)  
    item_vec = Flatten()(item_vec)  
    merge_vec = Concatenate()([user_vec, item_vec])  
    hidden = Dense(150, activation='relu')(merge_vec)  
    hidden = Dense(50, activation='relu')(hidden)  
    output = Dense(1)(hidden)  
    model = keras.models.Model([user_input, item_input], output)  
    model.compile(loss='mse', optimizer='sgd')  
    model.summary()  
    return model
```

Get movie embedding

```
#get embedding
user_emb = np.array(model.layers[2].get_weights()).squeeze()
print('user embedding shape:', user_emb.shape)
movie_emb = np.array(model.layers[3].get_weights()).squeeze()
print('movie embedding shape:', movie_emb.shape)
np.save('user_emb.npy', user_emb)
np.save('movie_emb.npy', movie_emb)
```


Draw

```
def draw(x,y):  
    from matplotlib import pyplot as plt  
    from tsne import bh_sne  
    y = np.array(y)  
    x = np.array(x,dtype=np.float64)  
    # perform t-SNE embedding  
    vis_data = bh_sne(x)  
    # plot the result  
    vis_x = vis_data[:, 0]  
    vis_y = vis_data[:, 1]  
  
    cm = plt.cm.get_cmap('RdYlBu')  
    sc = plt.scatter(vis_x, vis_y, c=y, cmap=cm)  
    plt.colorbar(sc)  
    plt.show()
```


Q&A