社群網路與推薦系統

HW3 Report

賴廷瑋 F44054045

目錄

Introduction

本次作業有涉及 NN 的部分皆**利用 pytorch 實踐**，並**參考原論文**後自己實踐，無利用現成的 model 套件，因參考論文與自己實踐花費耗時，實踐共 10 個模型。

此外，因自己使用本地環境執行，對 Douban Bank 大資料集在某些模型下有 RAM 不夠的問題，且若利用 Recall 或是 NDCG metric 進行比較，會需要非常 sparse 的資料作爲 input(沒有互動的部分也要考慮)，故考量到運算資源問題，僅使用 RMSE 做模型比較的標準。而 GBDT-LR 模型因訓練時間過久，無即時產生結果，但有以 sklearn 實踐模型架構。

故本次作業重心在對於模型的了解和實踐外，也針對 movielens 資料集的結果作總結分析，以提升自己對於模型的理解和參數對模型的影響，以及模型如何對於資料作處理等概念。

Table 本次作業實踐方法整理

類別	方法
Typical	User-based CF [UCF-s] (cosine as similarity)
Typical	User-based CF [UCF-p] (Pearson correlation as similarity)
Typical	Matrix Factorization [MF]
Typical	Factorization Machine [FM]
Typical	Pre-training via GBDT for LR
Typical	Pre-training via XGBoost for LR
NN-based	FM-supported Neural Networks [FNN]
NN-based	Deep Factorization Machine [DeepFM]
Recent NN-based	xDeepFM
My own method	XGBReg + NN

Methodology

- User-based CF

  利用使用者對於 item 評分的資訊，計算使用者之間的相似度，若兩使用者相似度高，則將一方對某 item 的評分納入計算另一方對某 item 的評分考量。

- MF

  將 user-item matrix 分解成兩個較低維度的矩陣相乘

- FM

  將資料組成 sparse feature matrix 後，將 rating 以回歸形式表示，並將二次項權重 matrix 進一步進行 vector 拆解，改善 sparse 問題。

- GBDT-LR

  利用 GBDT 自動學習資料的特徵組合，解決 FM 僅針對兩兩之間變量關係的問題，並以線性模型 LR 輸出 0-1 之間的數值。

- XGB-LR

  利用 XGBoost 自動學習資料的特徵組合，解決 FM 僅針對兩兩之間變量關係的問題，並以 LR 輸出 0-1 之間的數值。

- FNN

  將 sparse feature matrix 的每個 field 利用 fully connected layer 轉成 dense embedding (k+1)，每組分別代表 w1, v1, v2,…vk，再將每個 field 所得之 dense embedding 做 concatenation 後即得後續 NN 架構的 input，讓 NN 自動學習各 feature vector 之間的交互關係。

- DeepFM

  將每個 field 的 sparse feature 轉成 dense embedding 後同時分別進行 FM layer 和 NN layer 的計算，學習出二階和高階交互作用的關係，最後將 FM layer 和 NN layer 的結果相加，進行 activation function 轉置後即得預測。

- XDeepFM

  將 sparse feature 轉換成 dense feature 後，經過三個不同架構分別學習 explicit 和 Implicit 特徵，三個架構分別爲 Linear 層, CIN 層, 及 DNN 層。其中 CIN 層結合 CNN 的概念主要學習 explicit 特徵。最後將此三層 output 相加後經過 activation 得出最終預測。

- My own Method – XGBReg_NN

  參考 XGB 作爲 feature embedding 的想法，把 XGBoostRregressor 得到的 leaf node embedding 作爲 NN 層(兩層)的 INPUT，並結合 Linear 部分和 FM 部分相加，得出預測結果。

Experimental analysis

Experiment Settings

1. 依照規定，所有模型皆進行 5 次 cross validation，kfold 的部分使用 shuffle 且設定 random state 爲 42，確保每個模型對於每個資料的分割都一致。
2. Optimizer 的部分固定使用 adam 並且 learning rate 爲 1e-2。無使用任何 weight decay
3. Criterion(LOSS function)固定使用 nn.MSE 且 sqaured = False 作爲 RMSE 的計算，僅 MF 的部分爲依照課程講解自行設計 Loss Function。
4. 第一次實驗盡量皆保持所有模型參數一致，後續進行調整後觀察對於準確度的變化。
5. 每次實驗僅有一次 cross_validation，沒有 epoch 的介入。

模型	參數
UCF	取最接近的 20 鄰居作爲 Rating 的計算
MF	Latent factor k = 10, lambda = 1
FM	Latent factor k = 10
GBDT-LR	N_estimator= 10, max_depth= 3
XGB-LR	N_estimator= 10, max_depth= 3
FNN	K= 10, 兩層全連接層 dimensions: (32, 32)
DeepFM	K= 10, 兩層全連接層 dimensions: (32, 32)
xDeepFM	K= 10, 兩層全連接層 dimensions: (32, 32)
XGBReg_NN	K= 10, 兩層全連接層 dimensions: (32, 32) N_estimator= 10, max_depth= 3

各模型對於各 dataset 的表現 (Metric: RMSE)

模型	MovieLens
UCF(cosine)	2.04
UCF(pearson)	2.10
MF	3.18
FM	2.18
GBDT-LR	X
XGB-LR	1.2
FNN	1.54
DeepFM	1.47
xDeepFM	1.75
XGBReg_NN	1.67

Q1: Compared with the typical methods, can our NN-based approaches achieve comparable accuracy? Why?

從實驗結果來說，在未調整參數前，若僅經過一次的 cross validation，NN-based model 表現即整體優於 typical methods，從未調參結果來看，NN 模型平均的 RMSE 落在 1-2 之間，而 typical model 大多爲 2 以上(除使用 XGB-LR)。

原因

可以發現從 XGB-LR 開始之後的模型表現(RMSE)皆有明顯的提升，可推斷原因是這些模型有考慮 Feature 之間更高階的交互作用關係，進而提升對於 Rating 的準確度，例如在 xgb-lr 中透過 xgb 學習分類特徵的 leaf embedding 或是在 NN model 把 sparse matrix 轉成 dense feature 後透過深度網路學習高階關係，這個特質讓 NN based model 有更好的表現。

Q2: Are there any hyperparameters in each model that significantly affect the performance?

- 以下 metric 皆爲 RMSE

不同鄰居數影響

	Topk= 10	Topk= 20	Topk= 30
CF(cosine)	1.9	2.04	2.21

At hidden_dims= (32, 32), 不同 latent factor 影響

	K=15	K= 10	K= 5
MF	2.99	3.18	3.46
FM	2.19	2.18	2.19
FNN	1.49	1.54	1.59
DeepFM	1.50	1.47	1.57
XDeepFM	1.28	1.75	1.75

At K = 10, 不同全連接層 hidden_dim 影響

	32	64	128
FNN	1.54	1.48	1.4
DeepFM	1.47	1.49	1.42
xDeepFM	1.75	3.78	3.99

At max_depth= 3

	N_estimator= 10	N_estimator= 30
XGB_LR	1.2	1.2
XGBReg_NN	1.67	1.8

At n_estimator= 10

	Max_depth= 3	Max_depth = 5
XGB_LR	1.2	1.2
XGBReg_NN	1.67	1.55

Insights

- 對於 CF，從實驗來說，計算該 User 時所考慮的鄰居數愈小，在 test data 上，平均而言有更好的效果。
- 論 latent factor 對於不同 model 的影響，整體來看 K 愈大對於 RMSE 有些許的減少，但對於某些模型，K 愈大並不代表一定愈好，像是 FM 與 deepFM 在 k=10 時有相較佳的表現，而值得注意的是 xDeepFM 在 K=15 時有很大的表現提升，雖然如此，可以預想每個模型都有適合的 K 值，並不是愈大愈好。
- 論全連接層的 hidden dimension 對不同 model 的影響，FNN 和 DeepFM 在 hidden_dim 增加時有模型表現提升的現象，而 xDeepFM 在 hidden_dim 爲最小時反而是有更好的表現，且該提升顯著，可推論 xDeepFM 本身模型結構較爲複雜，若用較高的 hidden_dim 數將造成模型不易收斂，可能有 gradient diminishing 的問題。
- 論 n_estimator 和 max_depth 對 XGB 模型的影響，以 XGB_LR 模型而言，改變參數對於預測結果較無影響，而較低的 n_estimator

對於自行設計的 XGBReg_NN 有些許的效能提升，猜測可能因複雜度降低而進而得到泛化能力較好的模型，而 max_depth 提高時對於 XGBReg_NN 也有些許的提升，可能解釋爲更深的樹產生更好的 feature selection。

- 對於 NN-Based model，可發現在 k=15，hidden_dims= (32, 32) 時，xDeepFM 有最好的表現，且對於 NN-Based model, 同樣和 typical model 經過一樣的訓練次數，若繼續進行訓練，能有更好的表現。
- Typical model 整體而言可能會遇到訓練或者計算較耗時的問題，然而以 nn-based model 若有 GPU 等加速運算可以節省運算時間，並在應用上達到 online training 的目標。
- 因資料都爲 categorical feature，相較於連續變數，XGBoost 較不擅長處理 categorical feature，雖然實驗中 XGB_LR 有很好的表現，但模型是已經透過 Sklearn 內部訓練完畢後才做預測，有在 output 中發現 XGB 幾乎都是預測平均值，故在此 XGB 應有失準的現象，若將其他 NN-based model 繼續進行多次訓練，相信表現一定會比 XGB-LR 更好(XGB 模型無法與 NN-based model 公平比較)

Q3: Can you create a new end-to-end NN that combine the advantages of nicely-performed methods to beat all methods?

集結了 DeepFM 和 xDeepFM 的想法，也類似於 DeepGBM 的概念，利用 FM 模型對於二項交互作用關係良好的表達以及 nn 層對於高階交互作用關係的學習，也利用 XGBOOST 對於特徵分類的能力，將 FM 部分與 XGB, NN 連接層組合起來，去學習更好的 feature embedding 後也讓 NN 層學習高度的交互關係，是我設計的 XGBReg_NN 的主要想法。

以實驗結果來看，表現似乎與其他 NN-based model 差不多，而 XGBReg_NN 的實踐方法即利用 XGBRegression 對 sparse feature matrix 對不同的樹做 embedding，產生(資料數, n_estimators)的 embedding 後，再做爲兩層全連接層的 input，而效果沒有明顯提升的原因，我想是這邊用到 regression tree 的方式並沒有達到理想中 feature selection 的效果，若能利用 XGBClassifier 得到 leaf node index 並做 embedding(如 DeepGBM 想法)，應會有更好的結果。

Conclusions

novelty of my method

XGBReg_NN 結合了傳統 FM 與 XGB 特徵選取的概念，並結合 NN 的結構學習更好的高階作用關係，利用 XGB 對於連續特徵的特徵選擇有很強的能力，以及 NN model 對於離散變量能有更好的學習能力，我們能把連續型特徵透過 xgb+nn 的方式學習得更好，但因爲本資料集皆爲離散變量，xgb 在此可能無法發揮它最大的效益，但效果也與大部分的 nn 模型匹配。

summarization of the findings

總體來說

- NN 模型平均有比傳統模型有更好的預測能力
- Latent factor：K 值並不是愈大愈好，每個模型都有不同適合的 k 值
- 對於 NN model 來說，並非 hidden_dim 愈大或愈小愈好，例如 dimension 在 32-128 範圍增大時對於 fnn 和 deepfm 有較好的表現，而 xdeepfm 則在 hidden_dim = 32 有最好的表現。

- 對於 xgb 模型來說，n_estimator 和 max_depth 對於模型的影響並不是非常大，但適度減少 n_estimator 可以減少過擬合的問題，而適度的 max_depth 可以增加模型對於特徵選擇的能力。
- 最重要的點是，能讓預測最準最 robust 並不在模型本身的深度或是參數的調整，而在於如何有效地從資料取得好的 feature，且能針對不同類型的資料分別作優化，例如連續和離散分開，以及對 sparse feature matrix 做 dense embedding，這些都是近期的新模型提出的想法，而在本次實驗中，也驗證了該方法確實比傳統 model 更有效。

How to improve in the future

依照 DeepGBM 的想法，若能利用更好的特徵擷取方式，再進行 Dense feature 的轉換，搭配不同 NN 模型的算法(如增加 CNN 等特徵萃取的想法)，我想對於模型的預測會更加的穩定，且很重要的是要對模型運算時間有更好的解決方法，因推薦系統會有不斷新的資料產生，需要隨時隨刻進行模型的調整與預測，若能延續 DeepGBM 的精神，加上更全面的特徵選取方法，我想在應用上會有更好的表現。

Citations

- DeepFM: A Factorization-Machine based Neural Network for CTR Prediction, 2017
- Deep Learning over Multi-field Categorical Data– A Case Study on User Response Prediction, 2016
- xDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems, 2018
- Factorization Machines, 2010
- Matrix Factorization Techniques for Recommender Systems, 2009