SDML HW1 Task3 Report

B04902016 曾奕青
B04902103 蔡昀達
B04902105 戴培倫

October 18, 2018

Contents

1 Preprocess

Title and Abstract

1.1 Tokenize

nltk 的 RegexpTokenizer

1.2 Stopwords

nltk.corpus.stopwords('english')

1.3 Stemming

- stemming
  nltk 的 PorterStemmer
- lemmatize
  nltk 的 WordNetLemmatizer

由於 title 資料較不足，使用 doc2vec 的模型訓練之後，title 的 inference vector 部分無法從訓練資料中找出自己，說明訓練結果不佳，但在使用 stemming 的情況下，訓練結果大幅改善。

2 Feature Engineering

2.1 TF-IDF

- max_feature=150

2.2 Date

- year / month / date
  ex. A: 2013/08/04, B: 2011/01/31
  6 features: 2013, 08, 04, 2011, 01, 31
- Delta
  ex. A: 2013/08/04, B: 2011/01/31
  Delta = 20130804 - 20110131 = 20673

2.3 Node Features

- degree centrality
- betweenness centrality
- load centrality
- katz centrality
- pagerank

2.4 Similarity

- Jaccard Coefficient
  兩個 node Title 文字的 Jaccard Coefficient。
  兩個 node Abstract 文字的 Jaccard Coefficient。
- Cosine Similarity
  兩個 node Title Embedding 的 similarity。
  兩個 node Abstract Embedding 的 similarity。
- Correlation
  兩個 node Title Embedding 的 correlation。
  兩個 node Abstract Embedding 的 correlation。

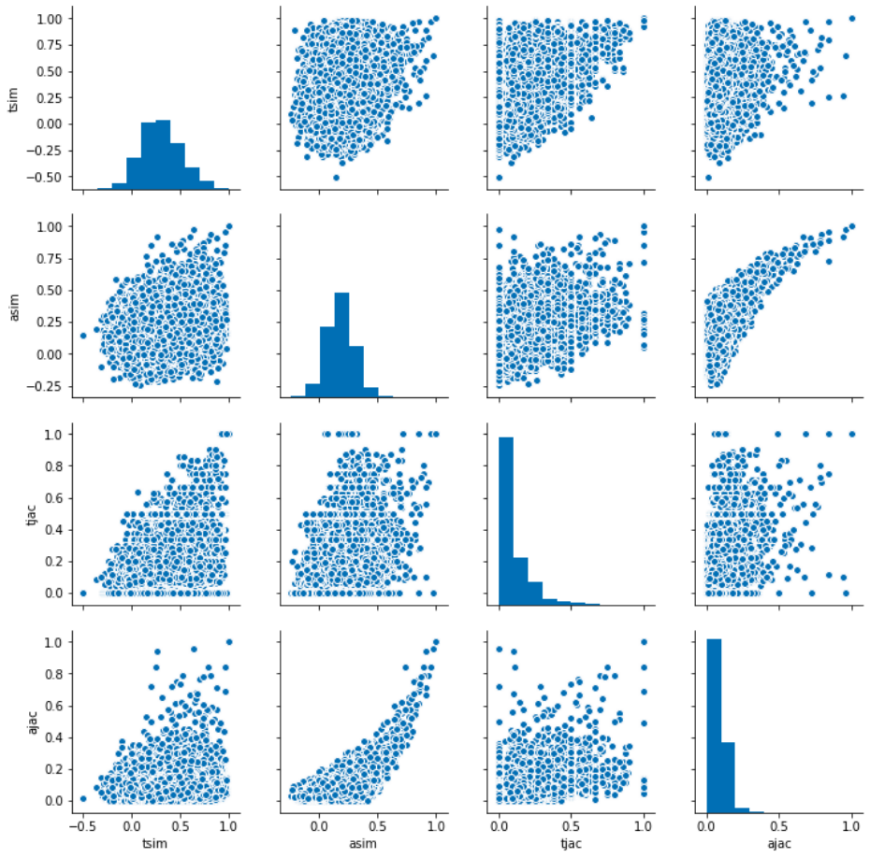


Figure 1: Test data: cosine similarity and jaccard's correlation

從圖中 (左上爲 (0,0)) 可以看出 cosine similarity 和 jaccard similarity 的關係。如 (0,2) 的 graph，可以看到當 title jaccard 趨近於 0 時，還是有非常多 title cosine similarity 很高的點，能看出 jaccard 可以補足許多 cosine similarity 沒辦法提供的訊息，在之後實驗中的 feature importance 也能看到 jaccard 的重要性非常高。

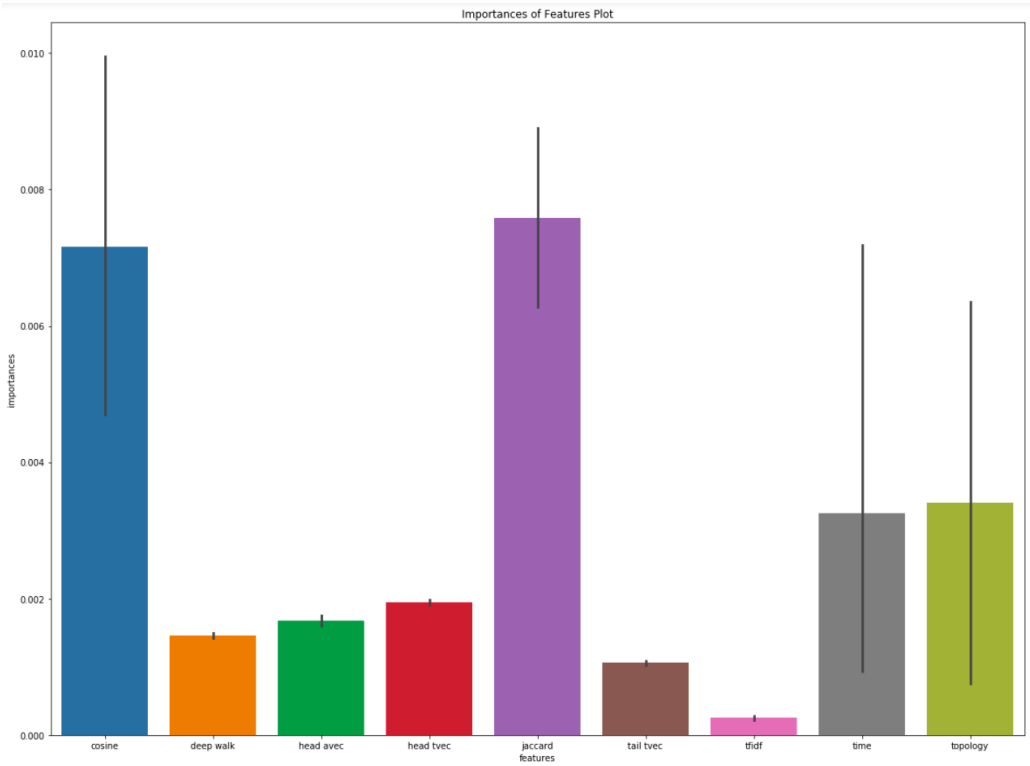


Figure 2: Feature Importance

由左至右分別爲 cosine similarity, DeepWalk Embedding, abstract embedding(from), title embedding(from), jaccard, title embedding(to), TF-IDF, time, centrality。

總結觀察到 feature 的重要順序爲

1. Time Features (delta)
2. Jaccard Similarity
3. Cosine Similarity
4. Centrality Features

TF-IDF 則沒有什影響，因此最後沒有使用。

3 Embedding

3.1 DeepWalk

- 使用 t3-train.txt 訓練 deepwalk embedding ，dimension=128。
- 沒有出現過的 node 則使用 0 向量

3.2 Doc2Vec

- Title 跟 Abstract 各做一個 dm=1 的 Model，dimension=100。
- Title 跟 Abstract 各做一個 dm=0 和 dm=1 的 Model。
  加上 dm=0 的 Model 沒有什麼差別因此之後沒有用。

3.3 LSTM AutoEncoder

從上星期至今還未收斂，還在 train，敬請期待。

4 Negative Sampling

使用拓樸結構上 path length 當作 negative samples 的依據，並檢查和 test data 文本相似度的 correlation。我們也嘗試過使用 similarity 優先調整 sampling weight，結果不理想。

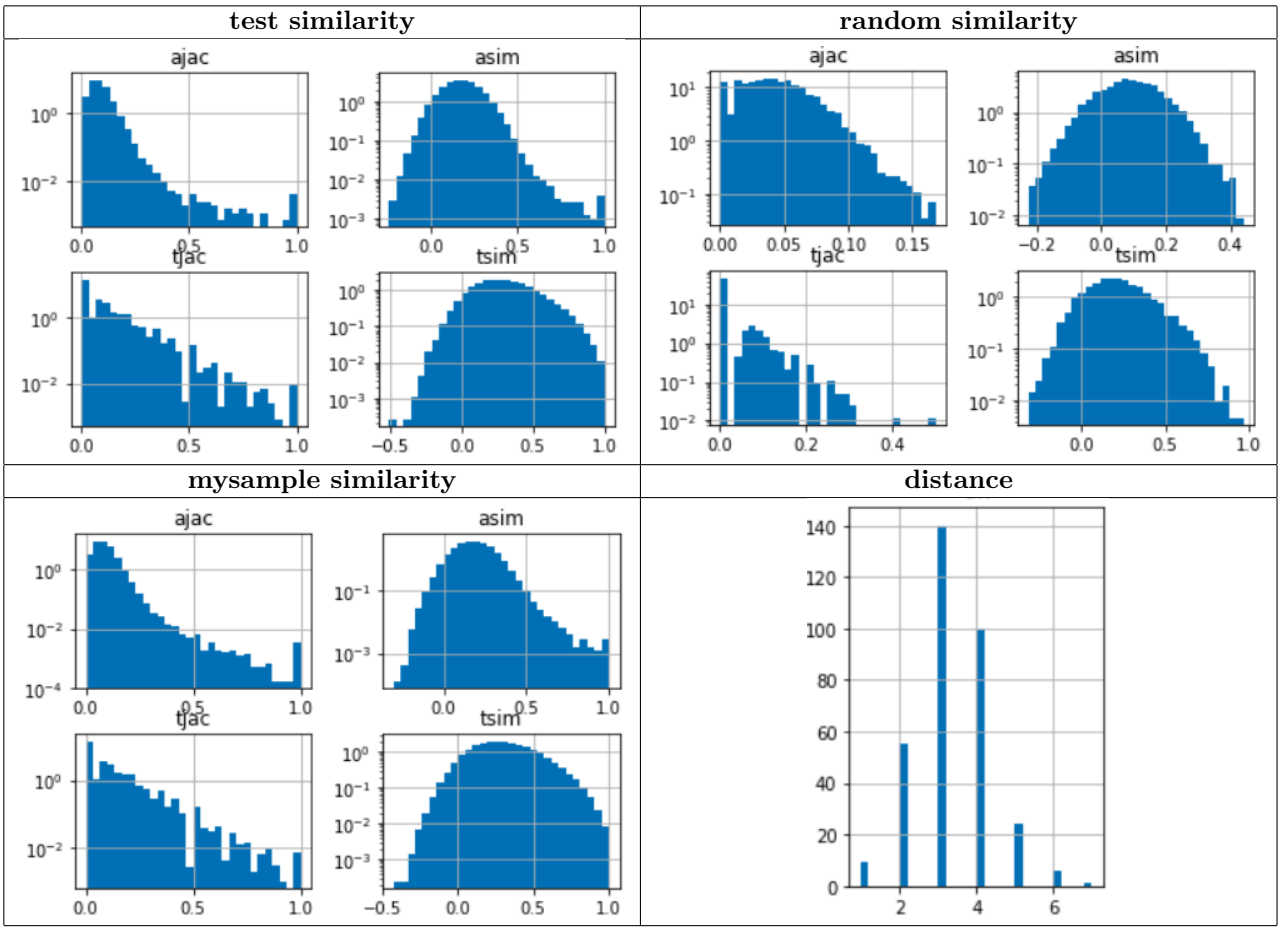
tsim: title cosine similarity
asim: abstract cosine similarity
tjac: title jaccard similarity
ajac: abstract jaccard similarity

	tsim	asim	tjac	ajac
0.15	0.099628	0.065307	0.000000	0.040000
0.25	0.161599	0.103507	0.000000	0.051724
0.5	0.290151	0.177533	0.000000	0.076923
0.75	0.435722	0.253419	0.125000	0.106667
0.95	0.663276	0.365574	0.300000	0.163043
mean	0.305581	0.180245	0.077954	0.083288
後5%	0.042368	0.027310	0.000000	0.029858
前5%	0.603031	0.340146	0.257885	0.153071

Figure 3: test similarity

	tsim	asim	tjac	ajac
0.15	0.105998	0.068927	0.000000	0.040650
0.25	0.168179	0.107561	0.000000	0.052632
0.5	0.295697	0.180759	0.058824	0.078947
0.75	0.439874	0.255580	0.133333	0.109756
0.95	0.658924	0.371054	0.307692	0.169231
mean	0.309643	0.183872	0.085585	0.085866
後5%	0.047895	0.030999	0.000000	0.030410
前5%	0.601788	0.344917	0.269684	0.159005

Figure 4: mysample similarity



從數值和圖表可以看出我們的 negative samples 在 cosine 和 jaccard similarity 上跟 test data 是相近的。Negative samples 主要是由 shortest path 2 - 4 步的 node 組成的。

5 Classifiers

5.1 Light gbm

- n_estimators = 2000
- objective = "binary"
- max_depth = 10
- early_stopping_round = 8
- learning_rate=0.1

訓練速度非常快，但在同樣參數下準確度略遜於 xgb

5.2 Random Forest

- n_estimators = 3200
- min_samples_leaf = 20
- class_weight = {0: 1, 1: 1.15}

random forest 在訓練時 positive 和 negative 準確率常會有 gap 導致最後 accuracy 不佳，因此借助了原來用於 unbalance training 的參數，以消除 positve 和 negative 的 accuracy gap，更改之後準確率也上升了不少。而這個 model 是三個中訓練最慢的，準確率也比不上 gradient boosting，做到 0.73 以後就沒再繼續使用 random forest

5.3 XGBoost

- n_estimators = 2000
- objective = "binary: logistic"
- max_depth = 10
- early_stopping_round = 8
- learning_rate=0.1

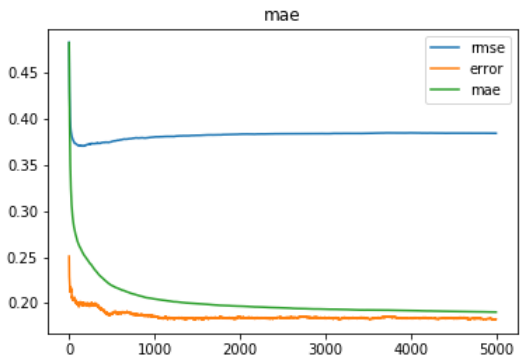
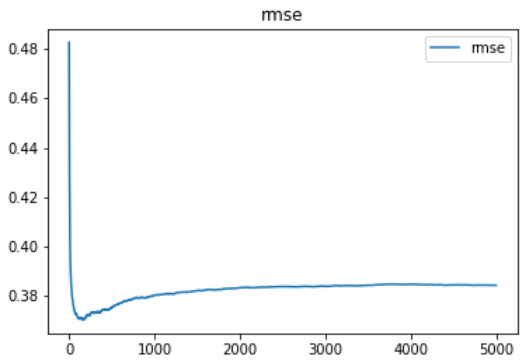
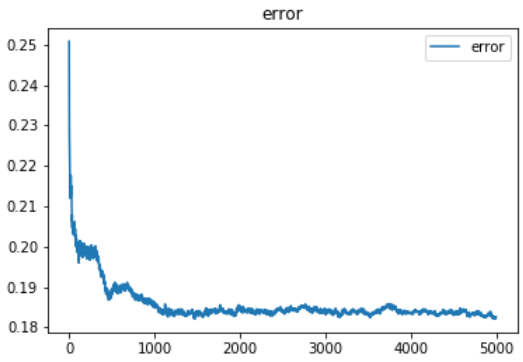
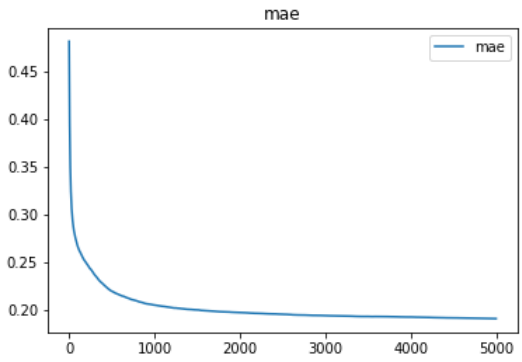
準確率最好

6 Strategy

6.1 Evaluation

在這次的作業中，我認爲最困難的問題出在 validation 和 evaluation 。validation set 會因爲 negative sampling 好壞的影響，很難具備有判斷 overfit 或是 underfit 的功能，對於調整 model 和評鑑結果有很大的障礙。Evaluation 上因爲 error surface 非常陡峭，對於使用 early stopping 有很大的選擇困難。

我所採用的策略如下，除了調整 negative sampling 的作法外，我使用多種 evaluation metric 來作爲 overfit 和 underfit 的指標，因爲單一種評量方式太不穩定。當 rmse 開始上升時，mae 和 accuracy 可能都還在變好; 當 accuracy 下降，rmse 和 mae 可能都在還在下降，因此使用多種 evaluation metrics 較能夠準確判斷，也讓我能夠有效調整模型。不過我認爲更精準的策略應該是設置足夠大的訓練次數，再經由人工檢視 error curve 來做判斷，缺點則是會花很多時間。

eval	rmse
	
error	**mae**
	

6.2 Data unbalance

in training data	not in training data
63145	11653
0 : 36970	0 : 684
1 : 26174	1 : 10967

7 Results

7.1 Single Model

XGBoost(rmse + error + mae)

- public: 0.76785
- private: 0.76849

由於比賽中有 validation 的 5000 筆資料沒有一起訓練，在比賽過後我們嘗試使用全部資料並手動設置訓練次數，可以得到更好的結果。

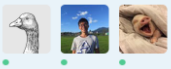
- public: 0.76713
- private: 0.77100

7.2 Ensemble

Majority out of 82 votes

- public: 0.76924
- private: 0.77250

在 ensemble 的策略上，採用 xgb 並設置不同參數做 majority vote，因爲 xgb 表現較其他模型好。然而，使用不同演算法的模型並設置權重可能會是更好的策略，但因爲時間不夠並沒有進一步實驗。

1	▲1	**b04902103_蔡昀達**		0.77250