

1 特征工程

【构建 lgb 特征与 nn 模型的 manual features。

训练数据只用了前 4 亿（尽管有的特征全部提取了，但实际只用 4 亿）；

除了特征 10，其余的训练集特征都是 4 亿存在一起的，特征 10 由于一下提取 4 亿内存会炸，所以分成前 2 亿和后 2 亿】

1.1 基础特征(长度相关按行提取特征、TFIDF 相关特征)

```
['query_length', 'title_length', 'WordMatchShare', 'WordMatchShare_query', 'WordMatchShare_title',  
'LengthDiff', 'LengthDiffRate', 'LengthRatio_qt', 'LengthRatio_tq',  
'TFIDFWordMatchShare', 'TFIDFWordMatchShare_title', 'TFIDFWordMatchShare_query']
```

1.2 NgramJaccard 特征

```
['NgramJaccardCoef_1' 'NgramJaccardCoef_2' 'NgramJaccardCoef_3' 'NgramJaccardCoef_4']
```

1.3 Levenshtein 相关

```
['Levenshtein_ratio', 'Levenshtein_distance_char', 'query_title_common_words', 'common_word_ratio']
```

1.4 sequencematch 相关

```
['lcs_substr_len', 'lcseque_len', 'longest_match_size', 'longest_match_ratio']
```

1.5 Fuzzy 相关（第 1 部分）

```
['fuzz_qratio', 'fuzz_partial_ratio']
```

1.6 Fuzzy 相关（第 2 部分）

```
['fuzz_partial_token_sort_ratio', 'fuzz_token_set_ratio', 'fuzz_token_sort_ratio']
```

1.7 熵相关

```
['query_Entropy', 'title_Entropy', 'query_title_Entropy', 'WordMatchShare_Entropy']
```

1.8 转换率特征

```
['query_convert', 'title_convert', 'query_title_convert'](强特)
```

根据 tfidf 权重对 query 提取一个关键词，title 提取两个关键词（set 排序再拼接成字符串），再进行转换率计算，并利用贝叶斯平滑缓解出现次数低带来的不准确性。分别计算 query 转换率，title 转换率和 query 与 title 的交叉转换率。

1.9 word2vec 句向量相似度

```
['w2v_avg_cosine', 'w2v_avg_cityblock', 'w2v_avg_minkowski', 'w2v_avg_braycurtis', 'w2v_avg_canberra']
```

1.10 全局出现频次相关

```
['query_title_click', 'query_nunique_title', 'query_click', 'title_nunique_query', 'title_click']
```

1.11 补充特征

```
['jaccard_similarity', 'qt_coword_query_ratio', 'qt_coword_title_ratio', 'qt_len_mean',  
'qt_common_word_acc', 'ngram_query_title_precision', 'ngram_query_title_recall', 'ngram_query_title_acc']
```

2 模型 (lightgbm + nn)

2.1 lgb 模型

(1) lgb 模型 1

使用特征 1.1 - 1.10, 4kw 数据一个 chunk(后 1kw 验证), 一共得到 10 个 lgb 模型, 将该 10 个模型对 2kw 测试集 A 和 1e 测试集 B 进行预测然后平均得到预测结果.

(2) lgb 模型 2

使用特征 1.1、1.3-1.6、1.8-1.11, 4kw 数据一个 chunk(前 1kw 验证), 一共得到 10 个 lgb 模型, 将该 10 个模型对 2kw 测试集 A 和 1e 测试集 B 进行预测然后平均得到预测结果.

2.2 nn 模型

(1) esim 模型 1 (pytorch)

使用 esim 模型, query+title+manual features 三端输入, 使用前 2 亿数据, 词向量 300d, 保存最优权重, 加载该权重分别对 2kw 和 1e 测试集进行预测。

(2) esim 模型 2 (pytorch)

同 esim 模型 1, 区别是使用后两亿的数据。

(3) esim 模型 3 (pytorch)

换成 100d 词向量, batch_size 设为 4096, learning_rate 每 4kw 数据乘以 0.8, 使用前两亿数据。

(4) lstm 模型 1 (keras)

2 层 lstm 提取特征再进行特征交互, 然后加入手工特征进入 MLP 层 (三端输入)。词向量为 300d, 使用前一亿数据, 前 8kw 训练, 后 2kw 验证。保存最优权重, 加载该权重分别对 2kw 和 1e 测试集进行预测。

(5) lstm 模型 2 (keras)

在 lstm 模型 1 最优权重的基础上, 使用第 3-4 亿的数据, 前 8kw 训练, 后 2kw 验证。保存最优权重, 加载该权重分别对 2kw 和 1e 测试集进行预测。

2.3 模型结果

表 1 7 个模型的预测结果

| 模型名称 | testA 榜结果 |
|-----------|-----------|
| lgb 模型 1 | 0.605084 |
| lgb 模型 2 | 0.604877 |
| lstm 模型 1 | 0.610921 |
| lstm 模型 2 | 0.613572 |
| esim 模型 1 | 0.615379 |
| esim 模型 2 | 0.616607 |
| esim 模型 3 | 0.626657 |

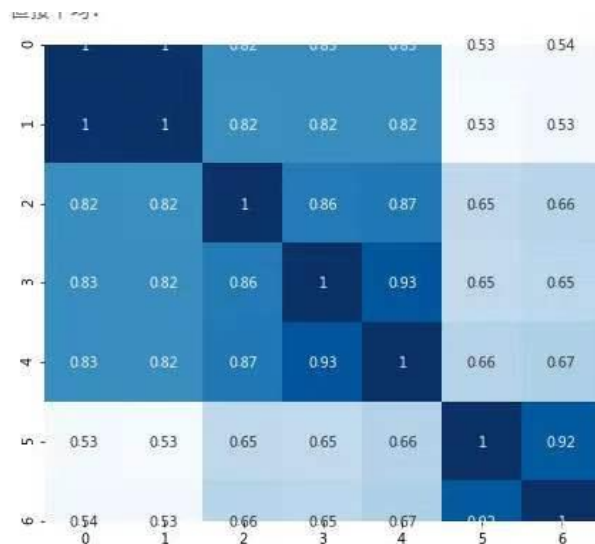


图 1 7 个模型的相关性（模型 0-7 分别为 lgb1、lgb2、esim1、esim2、esim3、lstm1、lstm2）

3 融合

利用表 1 中的 7 个模型所预测得到的结果进行加权融合，权重设置为 3:3:4:4:5:5:20。
testA 榜结果为 0.633680（第三），testB 榜结果为 0.653633（第三）。

如果任何地方存在疑虑，请随时联系我：

Email: 2016130205@email.szu.edu.cn

Tel: 13160739879