

IR FinalProject : Learning To Rank

实验报告

10152130122 钱庭涵

10152130122@stu.ecnu.edu.cn

一、 Introduction

1. 背景介绍：

LTR (Learning To Rank) 学习排序是一种监督学习的排序方法。LTR 已经被广泛应用到文本挖掘的很多领域，比如IR中排序返回的文档，推荐系统中的候选产品、用户排序，机器翻译中排序候选翻译结果等等。

IR领域传统的排序方法一般通过构造相关度函数，然后按照相关度进行排序。但传统的排序方法很难融合多种信息，且如果模型中参数比较多，也会使得调参非常困难，并很可能会出现过拟合现象。于是人们很自然的想到了用机器学习来解决这一问题，就有了Learning To Rank。机器学习方法很容易融合多种特征，并且有成熟深厚的理论基础，其参数是通过迭代优化出来的，有一套成熟理论解决稀疏、过拟合等问题。

2. 实验目的：

对于给定topic及某个引擎检索的结果，使用LTR的方法训练模型，进行结果的重新排序。

二、 Related Work

1. 实验要求：

LTR主要有三种排序学习方法：Point-wise、Pair-wise、List-wise。

实现简单的Point-wise方法，计算得到一些特征并进行分类或回归，可以调用相关工具包。

动手编写一些评价指标，如MAP、NDCG、P@K。

在上述要求的LTR算法基础上，进行拓展、比较或改进。

2. 使用数据：

① 2017TAR.zip

包括训练集和测试集的每个topic的title字符串和待排序文档列表pids，对于训练集，还包括qrel_abs_train标准相关性答案。

② docs.training.tar

包括训练集每个topic对应的文档列表及内容，可能存在不包括查询中所有pid的情况。

③ docs.tesing.tar

包括测试集每个topic对应的文档列表及内容，可能存在不包括查询中所有pid的情况。

④ 2017_test_qrels

包括测试集的标准相关性答案qrel_abs_test.txt。

3. 输出结果格式：

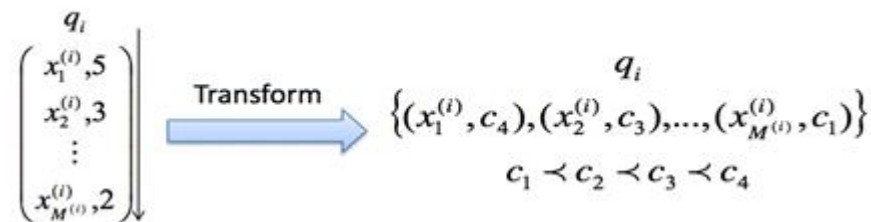
按照 TOPIC_ID Q0 DOC_PID RANK SCORE RUN_ID 输出。

三、 Methods

1. 前导知识

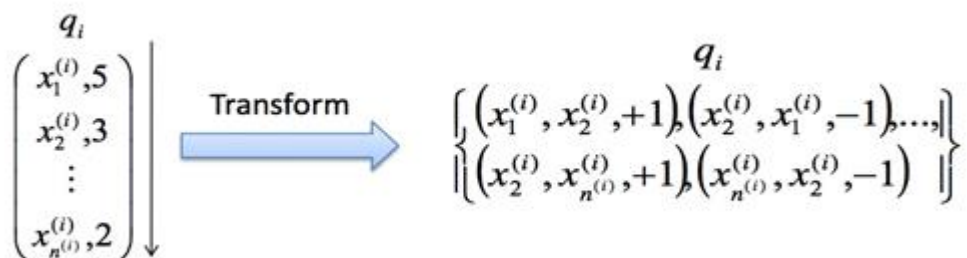
① Point-wise

Point-wise方法的处理对象是单独的一篇文档，将文档转换为特征向量后，机器学习系统根据从训练数据中学习到的分类或者回归函数对文档打分，打分结果即是搜索结果。



② Pair-wise

对于搜索系统来说，系统接收到用户查询后，返回相关文档列表，所以问题的关键是确定文档之间的先后顺序关系。Point-wise方法完全从单个文档的分类得分角度计算，没有考虑文档之间的顺序关系。Pair-wise方法将排序问题转化为多个pair的排序问题，比较不同文章的先后顺序。



③ List-wise

Point-wise方法将训练集里每一个文档当作一个训练实例，Pair-wise方法将同一个查询的搜索结果里任意两个文档对作为一个训练实例，List-wise方法与上述两种方法都不同，它直接考虑整体序列，针对Ranking评价指标进行优化。比如常用的MAP, NDCG。

④ TF-IDF

TF-IDF的主要思想是：如果某个词或短语在一篇文章中出现的频率TF高，并且在其他文章中很少出现，则认为此词或者短语具有很好的类别区分能力，适合用来分类。

TF-IDF实际上是TF * IDF，TF为词频，IDF为反文档频率，具体计算公式为：

$$score_{t,d} = tf_{t,d} * idf(t) = tf_{t,d} * \log \frac{N}{n_t}$$

⑤ BM25

BM25与TF-IDF类似，但计算方式不同。BM25方法中，将IDF值全部调整为正值，并增加了k和b两个参数，用来进行归一化约束，防止某个词的词频过大，以及调整文档长度对相关性影响的大小。具体计算公式为：

$$score_{t,d} = idf(t) * \frac{tf_{t,d} * (k + 1)}{tf_{t,d} + k * \left(1 - b + b * \frac{|d|}{avgdl}\right)}$$

$$idf(t) = \log \frac{N - n_t + 0.5}{n_t + 0.5}$$

⑥ VSM

VSM表示通过向量的方式来表征文本，一个文档被描述为一系列关键词的向量，因为每个关键词对总体的贡献度不同，所以对这些关键词赋予不同的权重词频TF。对于query查询来说，一个关键词的TF即该关键词在查询的所有词中的词频。

计算查询向量和每个文档向量的余弦相似度，即为文档的得分。

$$sim(d_1, d_2) = \frac{\vec{V}(d_1) \cdot \vec{V}(d_2)}{|\vec{V}(d_1)| |\vec{V}(d_2)|}$$

⑦ 逻辑回归

逻辑回归主要用于二分类问题，利用了Logistic函数（或称为Sigmoid函数），构造预测函数 $h(x)$ 和损失函数 $J(x)$ ，迭代计算使得损失函数最小，求得此时的回归参数 θ 。

$$h_{\theta}(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

$$J(\theta) = \frac{1}{m} \sum_{i=1}^n \text{Cost}(h_{\theta}(x_i), y_i) = -\frac{1}{m} \left[\sum_{i=1}^n y_i \log h_{\theta}(x_i) + (1 - y_i) \log(1 - h_{\theta}(x_i)) \right]$$

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x_i) - y_i) x_i^j$$

⑧ Sklearn SVM（调用sklearn库）

SVM是在分类与回归分析中分析数据的监督式学习模型与相关的学习算法。给定一组训练实例，每个训练实例被标记为属于两个类别中的一个或另一个，SVM训练算法创建一个将新的实例分配给两个类别之一的模型，使其成为非概率二元线性分类器。

SVM模型是将实例表示为空间中的点，这样映射就使得单独类别的实例被尽可能宽的明显的间隔分开。然后，将新的实例映射到同一空间，并基于它们落在间隔的哪一侧来预测所属类别。

⑨ MAP

单个主题的平均准确率是每篇相关文档检索出后的准确率的平均值。主集合的平均准确率(MAP)是每个主题的平均准确率的平均值。

MAP 是反映系统在全部相关文档上性能的单值指标。系统检索出来的相关文档越靠前(rank 越高), MAP就可能越高。如果系统没有返回相关文档，则准确率默认为0。

⑩ NDCG

对于排在结位置 n 处的NDCG的计算公式为：

$$N(n) = \underbrace{Z_n}_{\text{Normalization}} \underbrace{\sum_{j=1}^n}_{\text{Cumulating}} \underbrace{(2^{r^{(j)}} - 1)}_{\text{Gain}} \underbrace{/\log(1+j)}_{\text{Position discount}}$$

在NDCG中，文档的相关度可以分为多个等级进行打分。

DCG表示前p个位置累计得到的效益：

$$DCG_p = \sum_{i=1}^p \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

IDCG是指理想情况下的DCG，即DCG取得最大值的情况。其中|REL|表示文档按照相关性从大到小的顺序排序，取前p个文档组成的集合。

$$IDCG_p = \sum_{i=1}^{|REL|} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

由于每个查询语句所能检索到的结果文档集合长度不一，p值的不同会对DCG的计算有较大的影响，所以需要进行归一化处理。

$$NDCG_p = \frac{DCG_p}{IDCG_p}$$

多个查询语句的平均NDCG即为所求评测指标。

2. point-wise.py

① 训练集预处理：

对query进行预处理，后缀为'.pids'的文件存储着该查询对应的候选文档id，读取并以新的格式存储到新目录下；后缀为'.title'的文件存储着该查询对应的term，读取并进行一系列处理（全部转换成小写字母，词条化，去除停用词，词性归并，词干还原，去除大部分无用符号和首尾空格），将处理后的新字符串存储到对应新目录下。

对document进行预处理，ArticleTitle标签的文本和AbstractText标签的文本是所需的term，使用xml的方式获取这些文本，有些文档没有AbstractText标签，那么只提取ArticleTitle标签的文本。当遇到编码问题或非法字符串时，忽略该文档。

② 训练集建立索引：

对于每一个topic，建立其各自的索引文件，并同时记录存储每个文档的长度、文档总长度、文档总数量。为提高运行速度、减少占用内存空间，采用分块再合并的方式建立索引，并将每个topic的索引文件分成27个（分别为26个英文字母开头的term和其他term），且索引中按照key进行排序，便于之后的查找。

索引中记录当前topic对应的所有文档的term，记录term在每个包含它的文档中的词频。

建立过程中，当遇到编码问题或非法字符串时，删除该文档。

③ 训练集计算特征

这里采用了三种方法（TF-IDF、BM25、VSM），来计算每个topic对应的查询和文档的相关性特征，BM25的k取1.5、b取0.75。计算所需词频从上一步建立的索引中查找。

按照得分降序的顺序，对当前topic下三种方法各自的结果进行排序，按照 DOC_PID RANK SCORE 的格式，将结果存储到相应目录下。

④ 训练集合并特征

这里LTR采用的特征有6个：TF-IDF的排名及得分、BM25的排名及得分、VSM的排名及得分。因此需要将这6个特征进行合并。

对于每一个topic、每一个文档，按照 DOC_PID TF-IDF_RANK TF-IDF_SCORE BM25_RANK BM25_SCORE VSM_RANK VSM_SCORE 的格式，存储到相应目录下文件中。

⑤ 训练集获得01标签

读取训练集标准答案qrels_abs_train，按照合并后特征文件中，文档pid的顺序，将相应的相关度（0或者1）写入label文件，即每个topic中，每个文档的特征和标签在行号顺序上是一一对应的。

⑥ 训练集逻辑回归

根据这6个特征，使用逻辑回归进行阈值概率的计算，迭代步长alpha取0.0001，迭代次数取2000次。使用所有topic的所有的文档特征和标签来进行逻辑回归，并预先进行归一化处理，再进行多次迭代计算权重函数。

⑦ 测试集预处理、建立索引、计算特征、合并特征

与训练集类似，对测试集也进行预处理、建立索引、计算特征、合并特征。

⑧ 测试集预测

读取测试集合并后的特征文件，对于每一个topic，对相应特征文件进行归一化处理之后，根据上述训练集训练得到的theta权重函数，计算每个文档对应的阈值，并按照阈值（相关性概率）降序对每个topic中的文档pid进行排序，按照 TOPIC_ID Q0 DOC_PID RANK SCORE RUN_ID的格式，将结果写入对应的结果文件。

3. LTR.py

① 单用TF-IDF、BM25、VSM的相关度得到排序结果

根据point-wise方法计算得到的特征文件，对于每个topic下的TF-IDF.res、BM25.res、VSM.res, 按照要求的结果文件格式输出存储。

对于每种方法，将所有topic的结果合并到一起，得到三个结果文件。

② 使用TF-IDF_RANK BM25_RANK VSM_RANK 3个特征进行训练

根据point-wise方法计算且合并得到的总特征文件，抽取三种方法的3个排名特征，合并到一起，生成新的合并特征文件。

对于训练集和测试集都进行上述操作。

③ 训练集逻辑回归

根据这3个排名特征，使用逻辑回归进行阈值概率的计算，迭代步长alpha仍取0.0001，迭代次数仍取2000次。使用所有topic的所有的文档特征和标签来进行逻辑回归，并预先进行归一化处理，再进行多次迭代计算权重函数。

④ 测试集预测

读取测试集合并后的特征文件，对于每一个topic，对相应特征文件进行归一化处理之后，根据上述训练集训练得到的theta权重函数，计算每个文档对应的阈值，并按照阈值（相关性概率）降序对每个topic中的文档pid进行排序，按照 TOPIC_ID Q0 DOC_PID RANK SCORE RUN_ID的格式，将结果写入对应的结果文件。

⑤ pair-wise, 使用sklearn中的svm算法

调用sklearn库中的svm算法进行二分类预测，读取使用3个排名特征的合并特征文件，使用训练集所有topic的合并特征文件进行归一化处理并构建分类模型。

SVM的核函数使用'linear'，probability设为True以便能够得到相关性概率。

对于每一个topic，对相应特征文件进行归一化处理之后，使用上述构建的分类训练模型进行相关性概率的预测，按照 TOPIC_ID Q0 DOC_PID RANK 1_RATE RUN_ID的格式，将结果写入对应的结果文件。

4. evaluation.py

① 标准答案和结果文件读入

读入测试集标准答案qrel_abs_text.txt文件，将每个topic对应的相关文档总数和相关文档列表存入字典，同时按照旧的评测程序所需的标准答案格式，生成旧的格式的标准答案qrels_testing.res，以便于使用旧的评测程序进行评测。

读入测试集重排后的结果，将每个topic对应的文档pid、文档排名、文档相关性得分存入字典。

② 计算MAP评测指标

根据上述读入的标准答案字典rele_dict和结果文件字典res_dict，计算每个topic的MAP指标，所有MAP的平均值即为该结果的MAP评测指标。

③ 计算NDCG评测指标








等级设置为5级，按照相关性概率从小到大，将其均分为1~5五个等级。

取不同前p (p=10,20,50,100,200) 个结果进行计算，计算每个topic的DCG、IDCG，由此得到每个topic的NDCG指标，所有NDCG的平均值即为该p值下该结果的NDCG评测指标。

四、 Results

1. 运行结果

所有运行结果都放在res文件夹中

 res	2018/7/2 1:14	文件夹	
 10152130122_钱庭涵_BM25.res	2018/7/1 20:58	Compiled Resou...	6,784 KB
 10152130122_钱庭涵_point-wise.res	2018/7/2 0:44	Compiled Resou...	8,126 KB
 10152130122_钱庭涵_rank3.res	2018/7/2 1:08	Compiled Resou...	7,554 KB
 10152130122_钱庭涵_sklearn-SVM.res	2018/7/2 0:33	Compiled Resou...	7,504 KB
 10152130122_钱庭涵_TF-IDF.res	2018/7/1 20:58	Compiled Resou...	6,783 KB
 10152130122_钱庭涵_VSM.res	2018/7/1 20:58	Compiled Resou...	6,363 KB

① 10152130122_钱庭涵_TF-IDF.res

使用查询与文档的TF-IDF相关度进行排序的结果。

10152130122_钱庭涵_TF-IDF.res						
1	CD007431	0	4278185	1	5.7910793023957945	10152130122_TF-IDF
2	CD007431	0	6227357	2	5.755336856868198	10152130122_TF-IDF
3	CD007431	0	9516703	3	5.63852803318103	10152130122_TF-IDF
4	CD007431	0	2218716	4	4.6770652302197835	10152130122_TF-IDF
5	CD007431	0	6229214	5	4.343309476796847	10152130122_TF-IDF
6	CD007431	0	6233510	6	4.343309476796847	10152130122_TF-IDF
7	CD007431	0	8137841	7	4.187481438636085	10152130122_TF-IDF
8	CD007431	0	2657484	8	4.134405396810636	10152130122_TF-IDF
9	CD007431	0	4416290	9	3.808940511496016	10152130122_TF-IDF
10	CD007431	0	776316	10	3.799412138537351	10152130122_TF-IDF

② 10152130122_钱庭涵_BM25.res

使用查询与文档的BM25相关度进行排序的结果。

10152130122_钱庭涵_BM25.res						
1	CD007431	0	4278185	1	0.03197822621031521	10152130122_BM25
2	CD007431	0	17355018	2	0.023676892434602827	10152130122_BM25
3	CD007431	0	9516703	3	0.023361637184043667	10152130122_BM25
4	CD007431	0	2657484	4	0.022830665031264248	10152130122_BM25
5	CD007431	0	4416290	5	0.021031942293996977	10152130122_BM25
6	CD007431	0	2218716	6	0.01938256389406981	10152130122_BM25
7	CD007431	0	6227357	7	0.019082933527116985	10152130122_BM25
8	CD007431	0	6229214	8	0.01799728262150002	10152130122_BM25
9	CD007431	0	6233510	9	0.01799728262150002	10152130122_BM25
10	CD007431	0	123953	10	0.012409262924219684	10152130122_BM25

③ 10152130122_钱庭涵_VSM.res

使用查询与文档的VSM相似度进行排序的结果。

10152130122_钱庭涵_VSM.res						
1	CD007431	0	18394495	1	0.834847109936722	10152130122_VSM
2	CD007431	0	1533326	2	0.827837354384716	10152130122_VSM
3	CD007431	0	9148533	3	0.8173029975005109	10152130122_VSM
4	CD007431	0	10973402	4	0.8021806287494233	10152130122_VSM
5	CD007431	0	16538210	5	0.8001976040538968	10152130122_VSM
6	CD007431	0	15871484	6	0.7890763647670372	10152130122_VSM
7	CD007431	0	3237291	7	0.7872958216222171	10152130122_VSM
8	CD007431	0	8233471	8	0.787295821622217	10152130122_VSM
9	CD007431	0	10556409	9	0.7868995982344532	10152130122_VSM
10	CD007431	0	9825386	10	0.7839294959021856	10152130122_VSM

④ 10152130122_钱庭涵_point-wise.res

使用TF-IDF的得分及排名、BM25的得分及排名、VSM的得分及排名共6个特征，进行逻辑回归预测相关性概率并排序的结果。

10152130122_钱庭涵_point-wise.res						
1	CD007431	0	17762802	1	0.020771010899851547	10152130122_point-wise
2	CD007431	0	15830972	2	0.01815159474328929	10152130122_point-wise
3	CD007431	0	1826546	3	0.016721274639285723	10152130122_point-wise
4	CD007431	0	10537383	4	0.014347720855668355	10152130122_point-wise
5	CD007431	0	2975062	5	0.011554092198472966	10152130122_point-wise
6	CD007431	0	6617177	6	0.011520915954277849	10152130122_point-wise
7	CD007431	0	8187559	7	0.010626509661652642	10152130122_point-wise
8	CD007431	0	8137841	8	0.009567932093570385	10152130122_point-wise
9	CD007431	0	15118028	9	0.009323536218834692	10152130122_point-wise
10	CD007431	0	2940266	10	0.008597760334391842	10152130122_point-wise

⑤ 10152130122_钱庭涵_rank3.res

使用TF-IDF的排名、BM25的排名、VSM的排名共3个特征，进行逻辑回归预测相关性概率并排序的结果。

10152130122_钱庭涵_rank3.res						
1	CD007431	0	17762802	1	0.19477735339083757	10152130122_rank3
2	CD007431	0	15830972	2	0.11883317798727415	10152130122_rank3
3	CD007431	0	6227357	3	0.11501335119720213	10152130122_rank3
4	CD007431	0	1826546	4	0.09652626226453045	10152130122_rank3
5	CD007431	0	8137841	5	0.09324230864004802	10152130122_rank3
6	CD007431	0	10537383	6	0.06982790818190472	10152130122_rank3
7	CD007431	0	6617177	7	0.044945586009440465	10152130122_rank3
8	CD007431	0	2975062	8	0.043700233429342074	10152130122_rank3
9	CD007431	0	8187559	9	0.04240590907033131	10152130122_rank3
10	CD007431	0	15118028	10	0.032436418640063046	10152130122_rank3

⑥ 10152130122_钱庭涵_sklearn-SVM.res

使用sklearn库的SVM算法，TF-IDF的排名、BM25的排名、VSM的排名共3个特征，构建预测模型，预测相关度为1的概率并排序的结果。

10152130122_钱庭涵_sklearn-SVM.res						
1	CD007431	0	12486352	1	0.0234103834449	10152130122_sklearn-SVM
2	CD007431	0	16875182	2	0.0229159702121	10152130122_sklearn-SVM
3	CD007431	0	18211590	3	0.0227164710826	10152130122_sklearn-SVM
4	CD007431	0	15827541	4	0.0225746817358	10152130122_sklearn-SVM
5	CD007431	0	1533326	5	0.0224868293654	10152130122_sklearn-SVM
6	CD007431	0	1388253	6	0.0224367912143	10152130122_sklearn-SVM
7	CD007431	0	11759490	7	0.0224351683482	10152130122_sklearn-SVM
8	CD007431	0	7976944	8	0.0224078736259	10152130122_sklearn-SVM
9	CD007431	0	9322321	9	0.0224050052062	10152130122_sklearn-SVM
10	CD007431	0	15375820	10	0.0223864284155	10152130122_sklearn-SVM

2. 使用自己编写的MAP和NDCG进行评测的结果

所有评测结果都放在own_MAP_NDCG文件夹中

own_MAP_NDCG	2018/7/2 1:14	文件夹	
MAP_NDCG_BM25.res	2018/7/1 22:01	Compiled Resou...	1 KB
MAP_NDCG_point-wise.res	2018/7/2 0:45	Compiled Resou...	1 KB
MAP_NDCG_rank3.res	2018/7/2 1:09	Compiled Resou...	1 KB
MAP_NDCG_sklearn-SVM.res	2018/7/2 0:34	Compiled Resou...	1 KB
MAP_NDCG_TF-IDF.res	2018/7/1 22:01	Compiled Resou...	1 KB
MAP_NDCG_VSM.res	2018/7/1 22:01	Compiled Resou...	1 KB

① MAP_NDCG_TF-IDF.res

```

MAP_NDCG_TF-IDF.res x
1 result = 10152130122_TF-IDF
2 MAP = 0.23497491473721308
3 NDCG@10 = 0.05840017689990691
4 NDCG@20 = 0.08628233363629771
5 NDCG@50 = 0.14677338749554528
6 NDCG@100 = 0.23168916124378167
7 NDCG@200 = 0.3166395662946584

```

② MAP_NDCG_BM25.res

```

MAP_NDCG_BM25.res x
1 result = 10152130122_BM25
2 MAP = 0.16257132262992063
3 NDCG@10 = 0.05208305603006198
4 NDCG@20 = 0.08070296111873214
5 NDCG@50 = 0.147847378010641
6 NDCG@100 = 0.22618294423969112
7 NDCG@200 = 0.3103480891016465

```

③ MAP_NDCG_VSM.res

```

MAP_NDCG_VSM.res x
1 result = 10152130122_VSM
2 MAP = 0.4241306311331432
3 NDCG@10 = 0.053953878603953516
4 NDCG@20 = 0.08497180609932414
5 NDCG@50 = 0.15225615132157175
6 NDCG@100 = 0.2292472729820795
7 NDCG@200 = 0.3113641658194547

```

④ MAP_NDCG_point-wise.res

```

MAP_NDCG_point-wise.res x
1 result = 10152130122_point-wise
2 MAP = 0.3920862235371445
3 NDCG@10 = 0.05208310056568292
4 NDCG@20 = 0.08070303012690405
5 NDCG@50 = 0.14784750443322986
6 NDCG@100 = 0.2261831494789407
7 NDCG@200 = 0.3103484303856518

```

⑤ MAP_NDCG_rank3.res

```

MAP_NDCG_rank3.res x
1 result = 10152130122_rank3
2 MAP = 0.39238866073305395
3 NDCG@10 = 0.05220180343602994
4 NDCG@20 = 0.08080727192055794
5 NDCG@50 = 0.1479178194699927
6 NDCG@100 = 0.22625939350799432
7 NDCG@200 = 0.31052379962273574

```

⑥ MAP_NDCG_sklearn-SVM.res

```

MAP_NDCG_sklearn-SVM.res x
1 result = 10152130122_sklearn-SVM
2 MAP = 0.19997341311247555
3 NDCG@10 = 0.05208310056568291
4 NDCG@20 = 0.08070303012690405
5 NDCG@50 = 0.1478475044332299
6 NDCG@100 = 0.2261831494789407
7 NDCG@200 = 0.31034843038565185

```

3. 使用旧的评测程序进行评测的结果

所有评测结果都放在trec_res文件夹中

trec_res		2018/7/2 1:14	文件夹
trec_BM25.res	2018/7/1 21:09	Compiled Resou...	1 KB
trec_point-wise.res	2018/7/2 1:10	Compiled Resou...	2 KB
trec_rank3.res	2018/7/2 1:10	Compiled Resou...	1 KB
trec_sklearn-SVM.res	2018/7/2 1:10	Compiled Resou...	2 KB
trec_TF-IDF.res	2018/7/1 21:09	Compiled Resou...	1 KB
trec_VSM.res	2018/7/1 21:09	Compiled Resou...	1 KB

① trec_TF-IDF.res

trec_TF-IDF.res		
1	runid	all 10152130122_TF-IDF
2	num_q	all 30
3	num_ret	all 117561
4	num_rel	all 1857
5	num_rel_ret	all 1857
6	map	all 0.0860
7	gm_map	all 0.0395
8	Rprec	all 0.0823
9	bpref	all 0.0565
10	recip_rank	all 0.1917
11	iprec_at_recall_0.00	all 0.2234
12	iprec_at_recall_0.10	all 0.1531
13	iprec_at_recall_0.20	all 0.1194
14	iprec_at_recall_0.30	all 0.1052
15	iprec_at_recall_0.40	all 0.0949
16	iprec_at_recall_0.50	all 0.0868
17	iprec_at_recall_0.60	all 0.0780
18	iprec_at_recall_0.70	all 0.0706
19	iprec_at_recall_0.80	all 0.0575
20	iprec_at_recall_0.90	all 0.0498
21	iprec_at_recall_1.00	all 0.0434
22	P_5	all 0.0733
23	P_10	all 0.0933
24	P_15	all 0.0800
25	P_20	all 0.0800
26	P_30	all 0.0722
27	P_100	all 0.0670
28	P_200	all 0.0585
29	P_500	all 0.0410
30	P_1000	all 0.0300

② trec_BM25.res

trec_BM25.res		
1	runid	all 10152130122_BM25
2	num_q	all 30
3	num_ret	all 117561
4	num_rel	all 1857
5	num_rel_ret	all 1857
6	map	all 0.0661
7	gm_map	all 0.0290
8	Rprec	all 0.0582
9	bpref	all 0.0360
10	recip_rank	all 0.0988
11	iprec_at_recall_0.00	all 0.1352
12	iprec_at_recall_0.10	all 0.1148
13	iprec_at_recall_0.20	all 0.0839
14	iprec_at_recall_0.30	all 0.0791
15	iprec_at_recall_0.40	all 0.0777
16	iprec_at_recall_0.50	all 0.0746
17	iprec_at_recall_0.60	all 0.0641
18	iprec_at_recall_0.70	all 0.0616
19	iprec_at_recall_0.80	all 0.0531
20	iprec_at_recall_0.90	all 0.0478
21	iprec_at_recall_1.00	all 0.0434
22	P_5	all 0.0267
23	P_10	all 0.0367
24	P_15	all 0.0511
25	P_20	all 0.0583
26	P_30	all 0.0622
27	P_100	all 0.0520
28	P_200	all 0.0430
29	P_500	all 0.0333
30	P_1000	all 0.0247

③ trec_VSM.res

trec_VSM.res		
1	runid	all 10152130122_VSM
2	num_q	all 30
3	num_ret	all 117561
4	num_rel	all 1857
5	num_rel_ret	all 1857
6	map	all 0.1025
7	gm_map	all 0.0545
8	Rprec	all 0.1226
9	bpref	all 0.0822
10	recip_rank	all 0.3484
11	iprec_at_recall_0.00	all 0.3691
12	iprec_at_recall_0.10	all 0.1990
13	iprec_at_recall_0.20	all 0.1637
14	iprec_at_recall_0.30	all 0.1425
15	iprec_at_recall_0.40	all 0.1030
16	iprec_at_recall_0.50	all 0.0880
17	iprec_at_recall_0.60	all 0.0720
18	iprec_at_recall_0.70	all 0.0656
19	iprec_at_recall_0.80	all 0.0582
20	iprec_at_recall_0.90	all 0.0467
21	iprec_at_recall_1.00	all 0.0434
22	P_5	all 0.1667
23	P_10	all 0.1467
24	P_15	all 0.1244
25	P_20	all 0.1333
26	P_30	all 0.1322
27	P_100	all 0.1027
28	P_200	all 0.0748
29	P_500	all 0.0464
30	P_1000	all 0.0329

④ trec_point-wise.res

trec_point-wise.res		
1	runid	all 10152130122_point-wise
2	num_q	all 30
3	num_ret	all 117561
4	num_rel	all 1857
5	num_rel_ret	all 1857
6	map	all 0.1102
7	gm_map	all 0.0552
8	Rprec	all 0.1254
9	bpref	all 0.0896
10	recip_rank	all 0.3145
11	iprec_at_recall_0.00	all 0.3372
12	iprec_at_recall_0.10	all 0.2195
13	iprec_at_recall_0.20	all 0.1581
14	iprec_at_recall_0.30	all 0.1391
15	iprec_at_recall_0.40	all 0.1145
16	iprec_at_recall_0.50	all 0.1010
17	iprec_at_recall_0.60	all 0.0849
18	iprec_at_recall_0.70	all 0.0748
19	iprec_at_recall_0.80	all 0.0618
20	iprec_at_recall_0.90	all 0.0492
21	iprec_at_recall_1.00	all 0.0433
22	P_5	all 0.1667
23	P_10	all 0.1633
24	P_15	all 0.1644
25	P_20	all 0.1517
26	P_30	all 0.1411
27	P_100	all 0.1033
28	P_200	all 0.0760
29	P_500	all 0.0473
30	P_1000	all 0.0335

⑤ trec_rank3.res

	trec_rank3.res	
1	runid	all 10152130122_rank3
2	num_q	all 30
3	num_ret	all 117561
4	num_rel	all 1857
5	num_rel_ret	all 1857
6	map	all 0.1092
7	gm_map	all 0.0544
8	Rprec	all 0.1247
9	bpref	all 0.0900
10	recip_rank	all 0.2928
11	iprec_at_recall_0.00	all 0.3159
12	iprec_at_recall_0.10	all 0.2208
13	iprec_at_recall_0.20	all 0.1594
14	iprec_at_recall_0.30	all 0.1385
15	iprec_at_recall_0.40	all 0.1124
16	iprec_at_recall_0.50	all 0.0992
17	iprec_at_recall_0.60	all 0.0839
18	iprec_at_recall_0.70	all 0.0745
19	iprec_at_recall_0.80	all 0.0616
20	iprec_at_recall_0.90	all 0.0491
21	iprec_at_recall_1.00	all 0.0433
22	P_5	all 0.1800
23	P_10	all 0.1533
24	P_15	all 0.1511
25	P_20	all 0.1567
26	P_30	all 0.1411
27	P_100	all 0.1023
28	P_200	all 0.0738
29	P_500	all 0.0468
30	P_1000	all 0.0334

⑥ trec_sklern-SVM.res

trec_sklern-SVM.res x		
1	runid	all 10152130122_sklern-SVM
2	num_q	all 30
3	num_ret	all 117561
4	num_rel	all 1857
5	num_rel_ret	all 1857
6	map	all 0.0814
7	gm_map	all 0.0384
8	Rprec	all 0.0916
9	bpref	all 0.0555
10	recip_rank	all 0.1054
11	iprec_at_recall_0.00	all 0.1567
12	iprec_at_recall_0.10	all 0.1432
13	iprec_at_recall_0.20	all 0.1358
14	iprec_at_recall_0.30	all 0.1131
15	iprec_at_recall_0.40	all 0.0891
16	iprec_at_recall_0.50	all 0.0793
17	iprec_at_recall_0.60	all 0.0695
18	iprec_at_recall_0.70	all 0.0618
19	iprec_at_recall_0.80	all 0.0570
20	iprec_at_recall_0.90	all 0.0478
21	iprec_at_recall_1.00	all 0.0433
22	P_5	all 0.0400
23	P_10	all 0.0667
24	P_15	all 0.0756
25	P_20	all 0.0783
26	P_30	all 0.0733
27	P_100	all 0.0760
28	P_200	all 0.0603
29	P_500	all 0.0430
30	P_1000	all 0.0315

4. 评测结果对比表格

评测方法	传统方法/LTR方法（均保留四位小数）					
	TF-IDF	BM25	VSM	point-wise	rank3	sklearn-SVM
	以下为使用自己编写的MAP和NDCG进行评测的结果					
MAP	0.2350	0.1626	0.4241	0.3921	0.3924	0.2000
NDCG@10	0.0584	0.0521	0.0540	0.0521	0.0522	0.0521
NDCG@20	0.0863	0.0807	0.0850	0.0807	0.0808	0.0807
NDCG@50	0.01468	0.1478	0.1523	0.1478	0.1479	0.1485
NDCG@100	0.2317	0.2262	0.2292	0.2262	0.2263	0.2262
NDCG@200	0.3166	0.3103	0.3114	0.3103	0.3105	0.3103
	以下为使用旧的评测程序进行评测的结果					
map	0.0860	0.0661	0.1025	0.1102	0.1092	0.0814
P_5	0.0733	0.0267	0.1667	0.1667	0.1800	0.0400
P_10	0.0933	0.0367	0.1467	0.1633	0.1533	0.0667

五、 Discussion

1. 在使用自己编写的MAP和NDCG进行评测的结果中，NDCG值都随着p的值增大而增大，横向比较不同排序方法，MAP和所有NDCG的结果在方法效果好坏上得到相同的结果，即：

6种方法中，效果最好的为VSM，MAP值为0.4241，其次是rank3和point-wise，MAP值分别为0.3924和0.3921，rank3的结果比point-wise略好一点，第四是TF-IDF方法，MAP值为0.2350，第五是sklearn-SVM方法，MAP值为0.2000，最差的是BM25方法，MAP值仅有0.1626。

2. 在使用旧的评测程序进行评测的结果中，MAP和P_10的结果在方法效果好坏上得到相同的结果，与P_5有略微不同（在MAP和P_10的结果中，point-wise的结果好于rank3，在P_5的结果中则相反），即：

6种方法中，效果最好的为point-wise，map值为0.1102，其次是rank3，map值为0.1092，rank3的结果比point-wise略差一点，第三是VSM方法，map值为0.1025，它的效果与前两名相距不多，第四是TF-IDF方法，map值为0.0860，第五是sklearn-SVM方法，map值为0.0814，只比TF-IDF方法差0.0046，最差的是BM25方法，map值仅有0.0661。

3. 对比自己实现的MAP指标和评测程序评测出来的map指标，在数值上有较大出入，在方法的好坏比较上，前三名顺序正好相反，后三名顺序相同，可能是由于内部实现方法不同造成的差异。
4. point-wise和rank3方法的效果相差不大，因为一个特征的排名是按照得分降序排序的，即排名的顺序和得分大小正好相反，因此本质上达到的效果是一样的，特别是在归一化处理之后。
5. 按理说BM25方法是在TF-IDF方法上的优化，效果应该会更好，在本实验中，BM25的效果最差，可能是数据集本身造成的。
6. VSM方法的效果比TF-IDF和BM25好，是由于VSM考虑了查询本身的特征加上查询与文档之间的相似度，而TF-IDF和BM25只考虑了查询与文档的相关度，没有考虑查询本身的特征。
7. 由于VSM效果较好，TF-IDF效果中等，BM25效果较差，因此将三者结合起来，进行逻辑回归预测，效果的好坏就不好说。即point-wise和rank3方法可能由于LTR方法的实现，而变得排序准确率更高于VSM，也可能由于TF-IDF和BM25的较差效果，拖累了VSM的较好效果，而造成最后的排序准确率低于VSM。
8. sklearn-SVM方法属于pair-wise方法，考虑了文档与文档之间的联系和相关度，再去构建预测模型，但是最后的效果并不好，仅高与BM25方法，低于其他四种方法，可能是由于本实验的数据集中，文档与文档之间的联系并不大，sklearn-SVM将它们联系到一起反而使得效果变差。

六、 Conclusion





1. 传统的机器学习方法，如TF-IDF、BM25、VSM，只能考虑一种信息，难以融合多种参数。其中TF-IDF和BM25只考虑了查询与文档的相关度，不考虑查询本身的特征，BM25则是在TF-IDF的基础上，将IDF值都调整为正值并进行归一化约束。VSM则考虑了查询本身的特征和查询与文档之间的相似度，效果好于TF-IDF和BM25。
2. LTR的Point-wise方法，可以结合多种传统机器学习方法的参数进行模型构建，融合多种特征得到更全面的特征预测结果。
但有时排序的先后顺序较重要，排在前面的结果更优，而Point-wise方法则考虑全局的相关性，并不对排序的先后顺序优劣做惩罚。
3. LTR的Pair-wise方法，考虑了文档与文档之间的联系、相关度和先后顺序，将原来的排序问题转换成多个文档对的排序问题，保证了文档对内文档的先后顺序不变。
但它也存在问题，Pair-wise方法考虑了文档对内文档的相对先后顺序，却没有考虑文档出现在搜索列表中的位置，而排在搜索结果前面的文档更优，如果靠前的文档出现判断错误，代价明显高于排在后面的文档。
而且对于不同的查询，其相关文档数量差异很大，转换为文档对之后，文档对的数量差异也很大。
4. LTR的List-wise方法，直接考虑给定查询下的文档集合的整体序列，直接优化模型输出的文档序列，使得其尽可能接近真实文档序列，即在待排序文档列表的基础上进行优化。
5. 本实验使用了6种方法，其中有3种传统排序方法(TF-IDF、BM25、VSM)，2种Point-wise方法(point-wise、rank3)，1种Pair-wise方法(sklearn-SVM)。在自己实现的评测指标和旧评测程序下，方法好坏的顺序存在差异，前三名顺序正好相反，后三名顺序相同。
使用自己编写的MAP和NDCG进行评测：(效果从好到坏)
VSM > rank3 > point-wise > TF-IDF > sklearn-SVM > BM25
使用旧的评测程序进行评测：(效果从好到坏)
point-wise > rank3 > VSM > TF-IDF > sklearn-SVM > BM25
(其中point-wise和rank3的效果只有略微差距)
6. 以上方法除了算法本身考虑的特征和相关性，在不同数据集上的效果也会不同，数据集本身的差异也会影响方法的效果。

七、References


1. a_achengsong, “逻辑回归原理”
<https://blog.csdn.net/csgazwsxedc/article/details/69690655>
2. pakko, “逻辑回归”
<https://blog.csdn.net/pakko/article/details/37878837>
3. 一寒惊鸿, “【算法】 Learning to rank 小结”
<https://blog.csdn.net/dengxing1234/article/details/72953952>
4. lightty, “IR 的评价指标-MAP,NDCG 和 MRR”
<https://blog.csdn.net/lightty/article/details/47079017>
5. jiangyongyu, “信息检索评价指标 NDCG、a-NDCG”
<https://blog.csdn.net/jiangyongy/article/details/78698096>
6. Zhang_P_Y, “【机器学习】 Learning to Rank 简介”
<https://blog.csdn.net/LG1259156776/article/details/52749709>
7. Nanjunxiao, “学习排序 Learning to Rank 小结”
<https://blog.csdn.net/nanjunxiao/article/details/8976195>
8. Starzhou, “机器学习排序之 Learning to Rank 简单介绍”
<https://blog.csdn.net/starzhou/article/details/51438140>
9. 笨兔勿应, “Learning to Rank 简介”
<https://www.cnblogs.com/bentuwuying/p/6681943.html>
10. Wentingtutu, “Learning to Rank 入门小结 + 漫谈”
<https://www.cnblogs.com/wentingtutu/archive/2012/03/13/2393993.html>

八、 附件

1. 以上报告中的代码放在 code 文件夹中

 code	2018/7/2 1:14	文件夹	
 evaluation	2018/7/2 1:09	JetBrains PyChar...	4 KB
 LTR	2018/7/2 1:09	JetBrains PyChar...	10 KB
 point-wise	2018/7/2 0:47	JetBrains PyChar...	23 KB

2. Lab6 的实验报告中，代码分析的部分，放在 Lab6 实验报告文件夹中

 Lab6 实验报告 (在Lab6中已提交过)	2018/7/2 10:50	文件夹	
 10152130122_钱庭涵_Lab6	2018/7/2 10:49	PDF 文件	971 KB