

Web信息处理与应用

第七节 结果评价

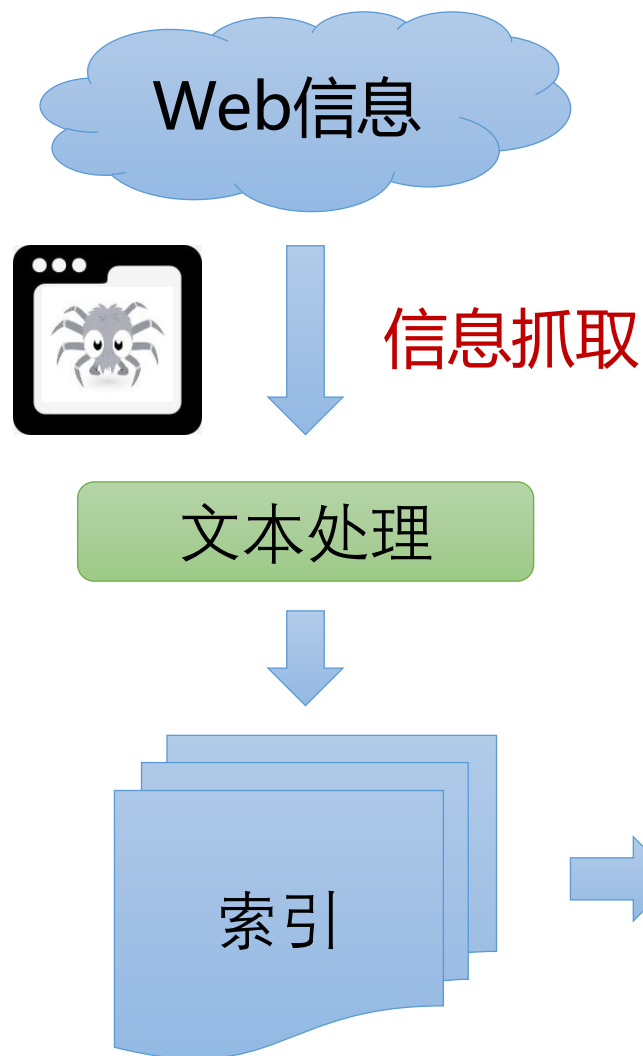
徐童 2021.10.25

- 一个好的检索应该是怎样?
- 如何做出选择?
 - B医生, 既治牙病, 又治眼病, 从医二十年
 - C医生, 专治牙病, 只有五年从医经验
- 从择医的角度考虑, 需要同时考虑专长与医术
- 对于网页排序而言, 也是如此
 - 网页内容匹配程度 → 医生的专长
 - 网页内容的质量 → 医生的经验与水平

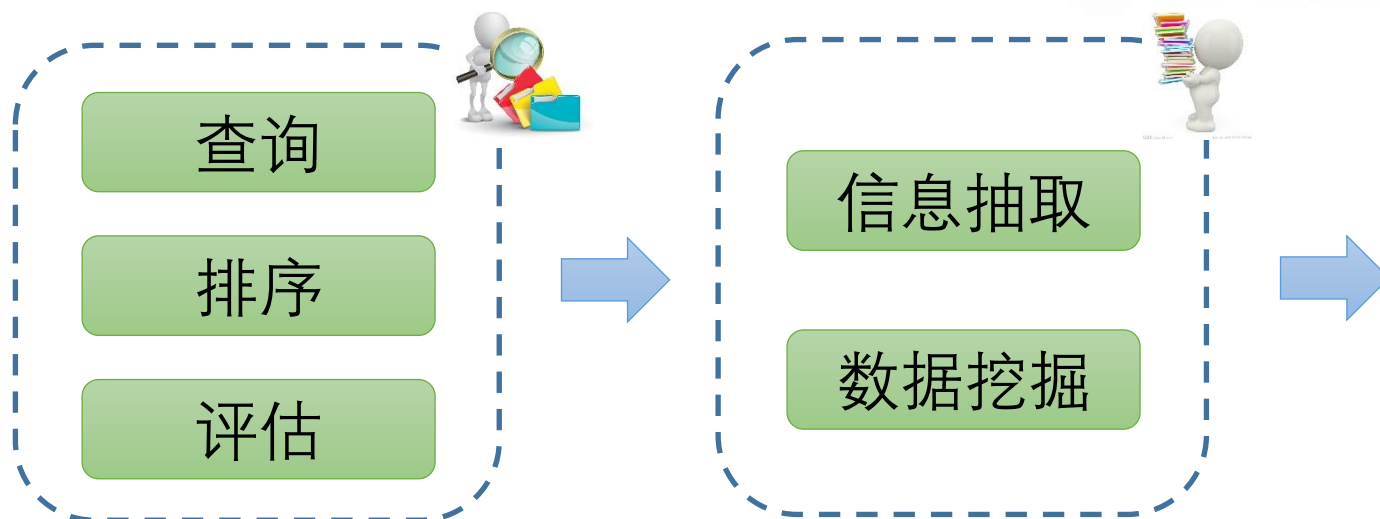
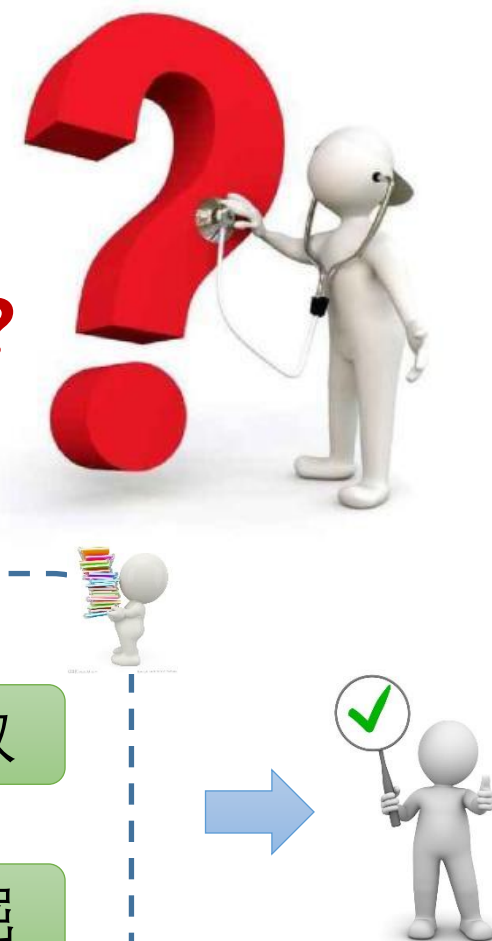


- **信息检索模型中的相关性**
- 相关性可通过以下两个维度进行衡量
 - 主题相关：文档与查询在主题上的一致性
 - 某种程度可体现在字面意义上的匹配性
 - 用户相关：文档在多大程度上满足用户需求
 - 可通过用户反馈判定，也可结合用户意图进行判定
- 相应的，可以通过二元（相关/不相关）或排序的方式进行评判
- 不同的维度有着不同的[评价体系](#)！

- 本课程所要解决的问题



第六个问题：
如何评估所得网页排序的质量？



- 结果评价的前提

- 相同的文档集合，相同的查询主题集合，相同的评价指标
- 比较内容：对不同的检索系统（排序方法）进行比较
 - 不同文档在规模、结构、主题上存在很大差异
 - 控制干扰因素，才能客观比较不同排序方法的优劣

Collection	Number of documents	Size	Average number of words/doc.
CACM	3,204	2.2 Mb	64
AP	242,918	0.7 Gb	474
GOV2	25,205,179	426 Gb	1073

- **结果评价的常见内容**

- 最主要（用户最关注）的两方面内容
 - 性能（Effectiveness）
 - 返回了多少相关文档，是否有遗漏，排序是否靠前
 - 效率（Efficiency）
 - 响应速度如何，时间和空间开销有多高
- 除此之外，其他指标也可衡量查询的结果
 - 结果的多样性、权威性、时新性与更新频率



- 结果评价的常见内容

- 评价效率（Efficiency）的常见指标

指标	英文名	解释
索引时间	Elapsed Indexing Time	特定系统中构建文档索引所需的总时间
索引处理器时间	Indexing Processor Time	构建文档索引所需要的CPU秒数， 即不考虑文件读写时间与并行加速影响
查询吞吐量	Query Throughput	每秒能够处理的查询数量
查询延迟	Query Latency	用户在输入查询后得到查询结果的等待时间
索引临时空间	Indexing Temporary Space	为构建索引所临时占据的硬盘空间
索引空间	Indexing Size	存储索引所需的硬盘空间

- 结果评价的常见内容
- 评价性能 (Effectiveness) 的常见指标
 - 面向单个查询的评价指标
 - 无序/二元结果: Precision、Recall、F-value...
 - 有序/多元结果: P@N、R@N、AP、NDCG...
 - 面向多个查询的评价指标
 - MAP、MRR及其各种拓展指标

- 单查询评价

- 无序结果评价

- 有序结果评价

- 相关度分级

- 多查询评价

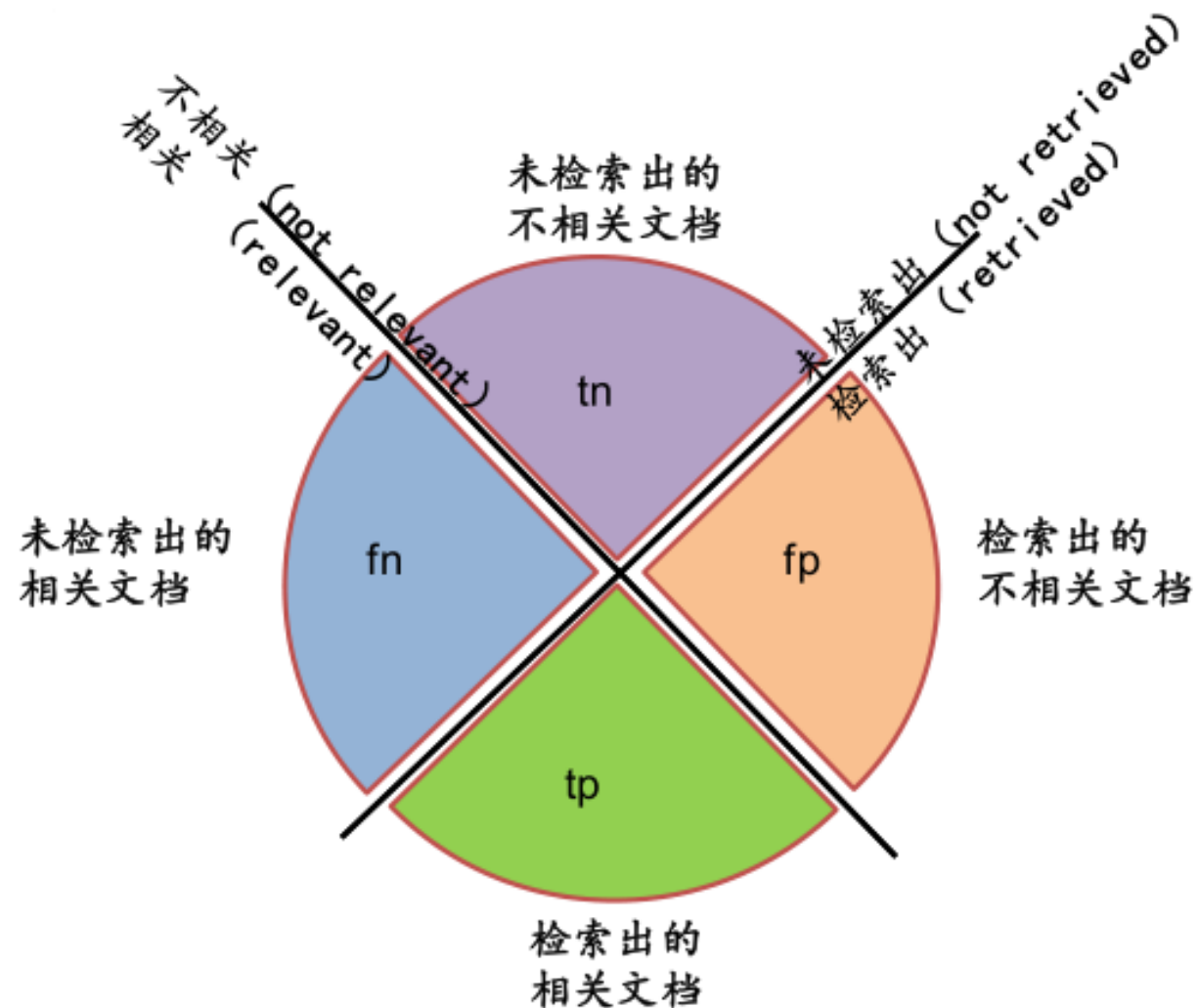
- 结果多样化评价

- 基本评价指标的矩阵化表示

- 两个视角下对于数据集的切分
 - P/N: Positive or Negative, 表示算法对样本的判断
 - T/F: True or False, 表示算法判断的正确与否
 - 四种简写的含义:
 - TP: True Positive, 样本为正例, 且被判定为正, 即真正
 - FN: False Negative, 样本为正例, 但错误地被判定为负, 即假负
 - FP: False Positive, 样本为负例, 但错误地被判定为正, 即假正
 - TN: True Negative, 样本为负例, 且被判定为负, 即真负

	被检索文档	未检索文档
相关文档	TP	FN
不相关文档	FP	TN

- 文档集合的基本划分



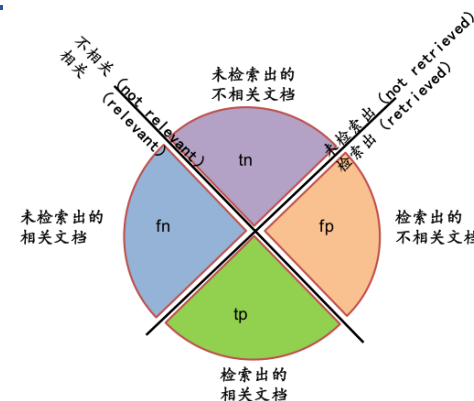
- 面向单查询的基本评价指标

- 准确率 (Precision)

- 指检索出的文档中，相关文档所占的比例，也称查准率
- 计算公式为 $TP/(TP+FP)$

- 召回率 (Recall)

- 指所有相关文档中，被检索出来的部分的比例，也称查全率
- 计算公式为 $TP/(TP+FN)$



- 一个P-R指标计算的实例

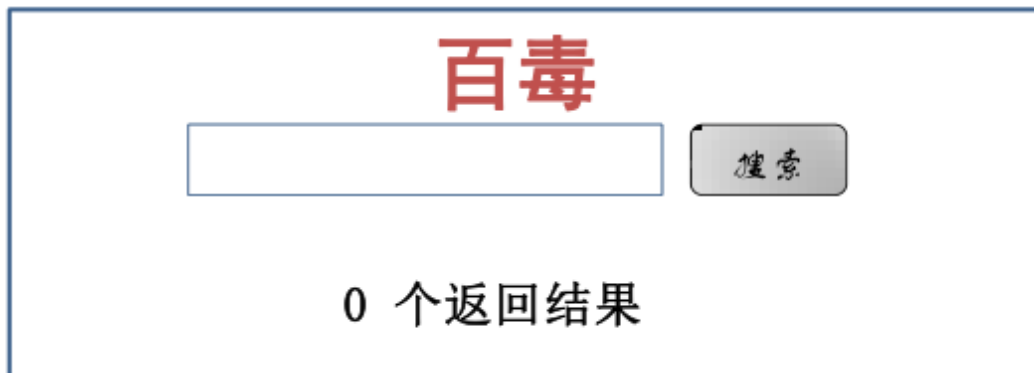
- 当查询1 的标准答案集合为 {d3,d4,d6,d9}时，可知：
 - 对于系统1， 查询1： 正确率 $2/5$ ， 召回率 $2/4$
 - 对于系统2， 查询1： 正确率 $2/4$ ， 召回率 $2/4$

系统&查询	1	2	3	4	5
系统1， 查询1	d3✓	d6✓	d8	d10	d11
系统1， 查询2	d1	d4	d7	d11	d13
系统2， 查询1	d6✓	d7	d2	d9✓	/
系统2， 查询2	d1	d2	d4	d13	d14

- 为什么某种方案被抛弃?

- 既然TP与TN都是正确结果，为什么不直接计算(TP+TN)的全局比例?
 - $(TP+TN)/(TP+TN+FP+FN)$ ，即Accuracy，在模式分类中经常被使用
 - 然而，它在信息检索的相关任务中并不常见，为什么?

如何以最低的代价做一个Accuracy接近 100% 的搜索引擎?



百毒

0 个返回结果

- 准确率与召回率的应用场景

- 作为最为基础的评价指标，准确率与召回率在众多领域有着广泛的应用
 - 凡是分类问题，基本都采用准确率与召回率衡量其性能

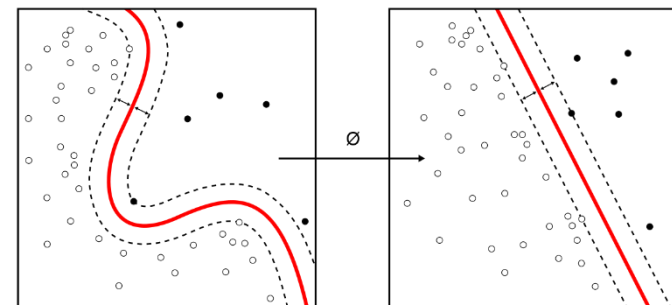
中文分词

我→只→是→做→了→一→些→微→小→的→工→作→○

文本判别



样本分类



- **准确率与召回率的平衡**

- 不同应用场景中，对于准确率和召回率有着不同的侧重
 - 邮件分类：宁愿放过一些垃圾邮件，也不能错杀正常邮件
 - 牺牲（对垃圾邮件的）召回率，保证较高准确率
 - 智慧医疗：宁愿多判断一些疑似患者，不能漏掉一个真实病人
 - 牺牲（对确诊病人的）准确率，保证较高召回率



- 召回率的近似计算

- 对于大规模文档集合，列举每个查询的所有相关文档是不可能的事情
 - 因此，不可能准确地计算召回率



- **召回率的近似计算**

- 解决方法：缓冲池（Pooling）方法
 - 针对某一检索问题，各个算法分别给出检索结果中的Top N个文档
 - 将这些结果汇集起来并进行人工标注，从而得到一个相关的文档池
 - **潜在假设**：大多数相关文档都在这个文档池（Doc Pooling）中
- 这一方法的可行性在于，虽然它实际上仍然无法得到全部相关文档，因此并不能得到召回率的绝对值。但是，它可以比较各个算法的**相对优劣**
 - 因此，这一算法在各个测评中被广泛采用，N通常取50-200

- 从P-R的平衡到F值

- 如前所述，准确率与召回率之间存在权衡
 - 如何综合评价一个算法在这两项指标上的性能？
- F值（F-measure），即准确率与召回率的加权调和平均数

$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha) \frac{1}{R}} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

- 通常情况下，我们取 $\alpha=0.5$ 或 $\beta=1$ （即两者同等重要）
 - 此时，可得基本的F1值，即 $F=2PR/(P+R)$

- 从P-R的平衡到F值

- 为何不使用算数平均，而使用调和平均综合这两个指标？如何综合评价一个算法在这两项指标上的性能？
 - 调和平均较为“保守”，在结果上小于等于算术平均或几何平均
 - 较小数提升的拉动作用应比较大数提升的拉动作用更显著，而算术平均是简单地将相同提升/降低幅度的影响视作是等价的
 - 算数平均和几何平均在处理极端情况下的效果并不够合理
 - 一个不好的例子：如果采用算术平均计算F值，那么一个返回全部文档的搜索引擎的其F 值就不低于50%，显然这不是一个好结果
- 准确率、召回率与F值是信息检索任务中最为基础和常用的三个指标

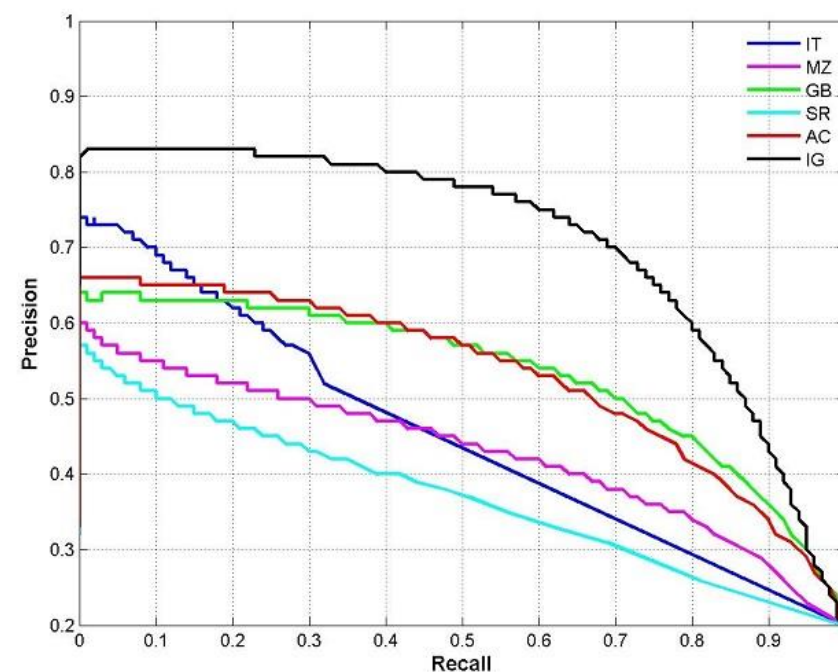
- **P-R曲线与ROC曲线**

- 在分类问题中，准确率与召回率的平衡是通过选定不同阈值实现的
 - 例如，通过调控相关性阈值 θ ，可以控制检索所得的文档数量
 - 较低的阈值可以使得返回更多文档，但也混入大量不相关的文档
 - 较高的阈值可以保障文档的相关性，但也会遗漏许多相关的文档
 - 如何选择合适的阈值？
 - 通过绘制不同阈值下的指标变化曲线，可以帮助我们做出选择

Tips: 对每个文档，计算其与查询的相关性系数，若大于 θ 即认定为相关文档

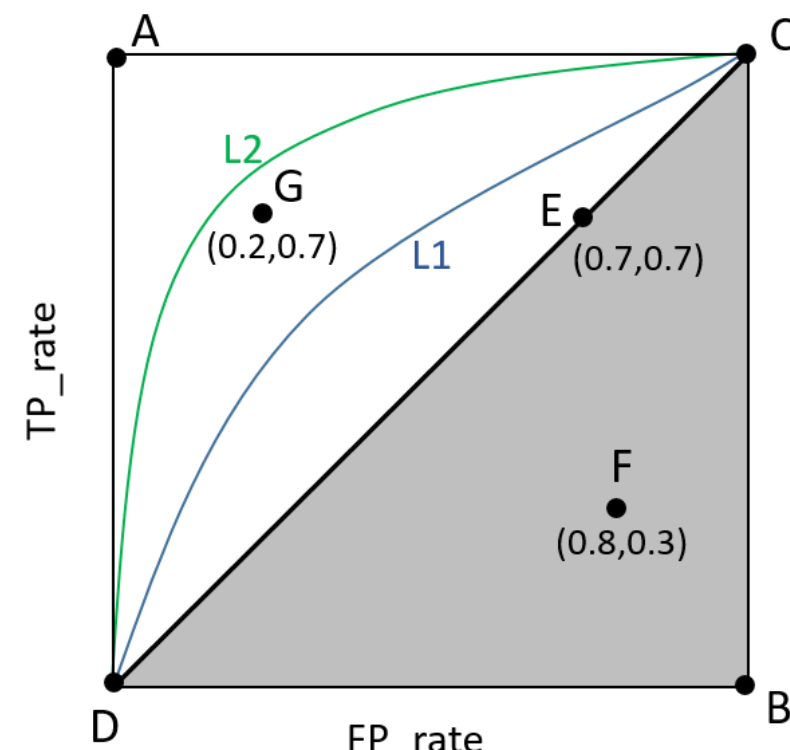
- **P-R曲线与ROC曲线**

- P-R曲线 (Precision-Recall Curve)
 - 以准确率和召回率分别作为两条轴线
 - 通过选定不同的阈值得到不同的P-R点并连接成线
 - 通过P-R曲线, 可以直观地看出准确率与召回率之间的平衡关系



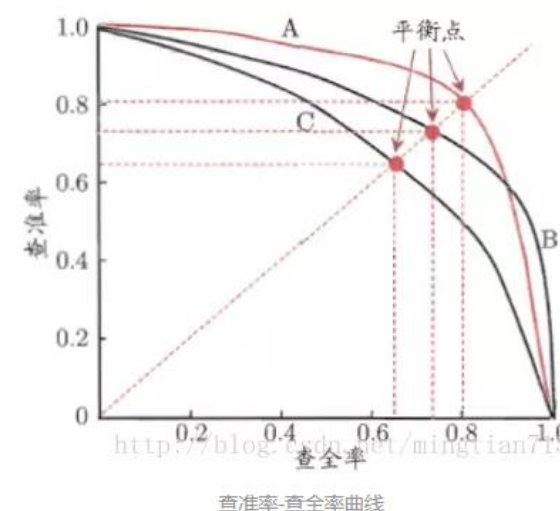
- **P-R曲线与ROC曲线**

- ROC曲线 (Receiver Operating Characteristic Curve, 接受者操作特征曲线)
 - 以真正率【 $TP/(TP+FN)$ 】和假正率【 $FP/(FP+TN)$ 】作为两条轴线
 - 真正率/命中率, 假正率/误报率
 - 通过选定不同的阈值得到不同的真正率-假正率点并连接成线
 - 对角线表示区分能力为0, 即随机猜测
 - 在对角线上端越远, 效果越好
 - 低于对角线的结果无意义 (无区分度)



- **P-R曲线与ROC曲线**

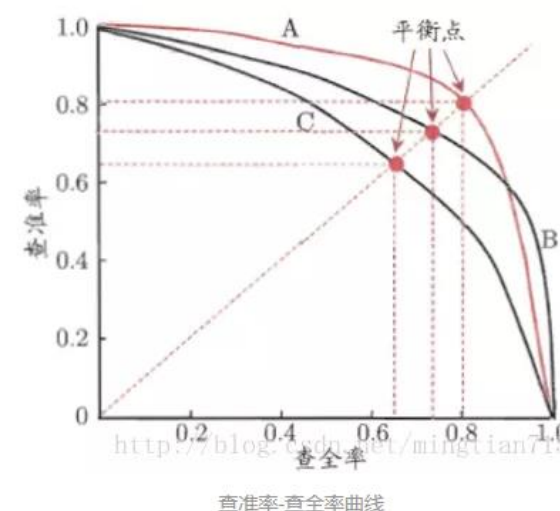
- 如何基于P-R曲线或ROC曲线判别算法好坏
 - 如果线A将线B完全包住，显然线A对应的算法效果更好
 - 如果两条线发生重合，则可依据以下规则判别：
 - 计算AUC，AUC更高者效果更好
 - Area under curve，即曲线下面积
 - 可通过积分近似计算
 - 另外，当使用P-R曲线时，可使用平衡点计算
 - 平衡点即Precision = Recall的点，值越高越好



- **P-R曲线与ROC曲线**

- P-R曲线与ROC曲线的选择

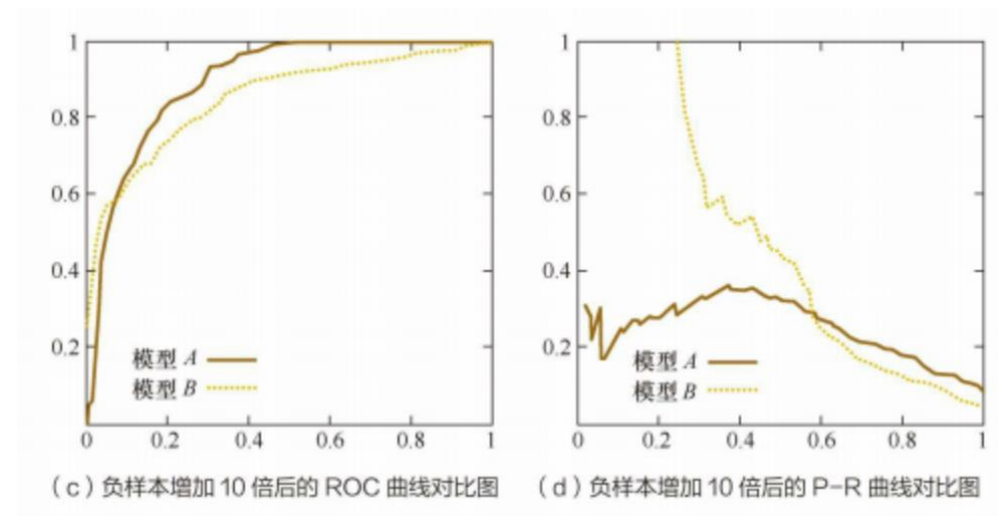
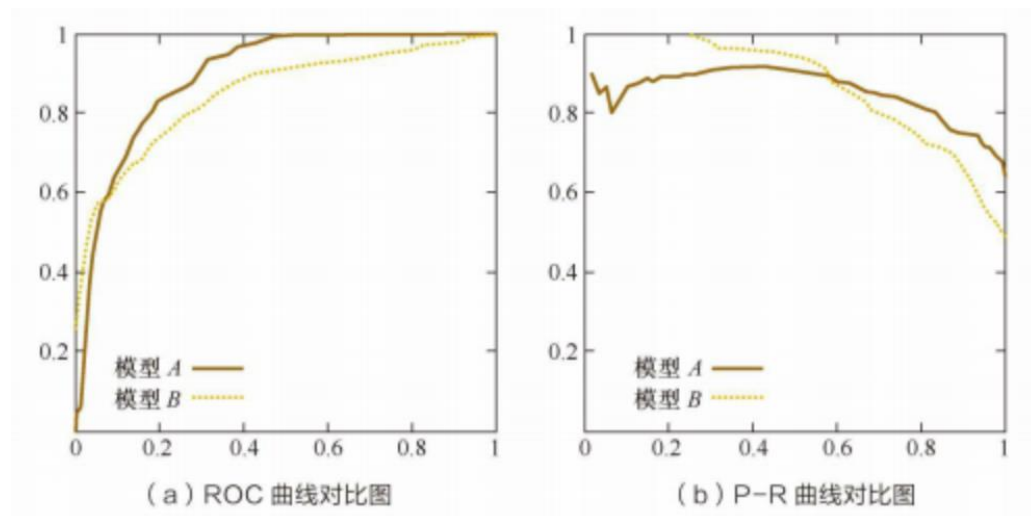
- ROC曲线兼顾正负样例，更为全面，而P-R曲线则只考虑正例
 - 用户往往更关心正样本，如果面向特定应用场景（如检索），P-R曲线是个好选择
- 正负样本比例失调时，P-R曲线更合适
 - 当负样本比重过高时，负例的数目众多致使FPR的增长不明显，导致ROC曲线呈现一个过分乐观的效果估计，从而难以体现出性能的差异性



- **P-R曲线与ROC曲线**

- P-R曲线与ROC曲线的选择

- P-R曲线受分布影响大，多份数据且正负比例不一时ROC曲线更合适
 - ROC曲线两个指标各自针对正负样本，而Precision只针对正样本，受影响较大



- 单查询评价

- 无序结果评价

- 有序结果评价

- 相关度分级

- 多查询评价

- 结果多样化评价

- 用户更关心有序结果
- 一个好的搜索引擎，应该尽可能将用户需要的文档排在较为靠前的位置



Baidu



Google

- **从无序的P、R到有序的P@N、R@N**
- P、R、F都是基于集合计算的指标，面向无序文档集合进行计算
 - 因此，它们无法直接应用于有序文档集合，需要进行拓展
 - 而这一点显然局限了它们的功能，并且可能导致误导
 - 例如，两个系统对某查询都返回20个文档，其中相关文档数都是10，但第一个系统是前10条结果，后一个系统是后10条结果。
 - 显然，在这一情况下，前者更为优越，但单纯P、R难以区分。
- 解决方案：引入序的作用和区分
 - P@N、R@N、AP、NDCG...

- **P、R@N的概念与意义**

- $P@N$, 即Precision@N, 指前N个检索结果文档的准确率
 - 由于大多数用户只关注第一页或前几页, 因此 $P@10$ 、 $P@20$ 等对于大规模搜索引擎来说是很合适的评价指标。
 - 如果相关文档数小于N, $P@N$ 的理论上限必定小于 1
- 同理, 可得 $R@N$, 即Recall@N, 指前N个检索结果找回的相关文档比例
 - 由于返回结果有限, $Recall@N$ 值, 甚至其理论上限往往都远小于 1
 - 理论上限为 $N/\text{相关文档数}$, 即使通过Pooling加以控制仍然较小

- **P、R@N的实例**

- 如果查询1 的标准答案集合为 {d3,d4,d6,d9}, 那么有:
 - 系统1 查询1: $P@2=1$, $P@5=2/5$;
 - 系统2 查询2: $P@2=1$, $P@5=3/5$

系统&查询	1	2	3	4	5
系统1, 查询1	d3 ✓	d6 ✓	d8	d10	d11
系统1, 查询2	d1 ✓	d4	d7	d11	d13 ✓
系统2, 查询1	d6 ✓	d7	d2	d9 ✓	/
系统2, 查询2	d1 ✓	d2 ✓	d4	d13 ✓	d14

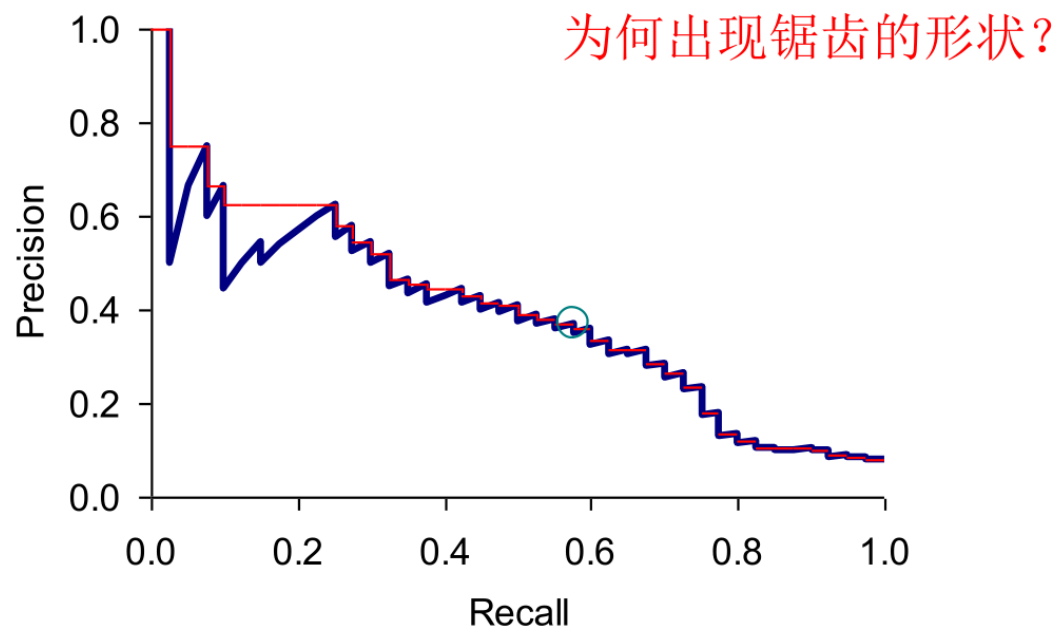
- 一种特例：R-Precision

- 检索结果中，在所有相关文档总数位置上的正确率
 - 由于N往往小于相关文档总数，因此设计了这一特殊指标
 - 例如，如果相关文档总数为3，那么考察P@3作为R-Precision
 - 实例：如果查询2 的标准答案集合为 {d1,d2,d13}，那么：
 - 系统1：R-Precision=1/3； 系统2：R-Precision=2/3；

系统&查询	1	2	3	4	5
系统1， 查询1	d3 ✓	d6 ✓	d8	d10	d11
系统1， 查询2	d1 ✓	d4	d7	d11	d13 ✓
系统2， 查询1	d6 ✓	d7	d2	d9 ✓	/
系统2， 查询2	d1 ✓	d2 ✓	d4	d13 ✓	d14

- 面向P、R@N的P-R曲线

- 在有序结果情况下，可以不再采用不同阈值作为P-R值的依据，而是通过依次计算前N个结果对应的P-R值绘制曲线



- 新的不相关文档被检索时，Recall不变，Precision下降

- 更多的评价准则：AP

- 平均准确率 (Average Precision, AP)

- 用于对不同召回率点上的正确率进行平均
- 通常情况下，AP有三种不同的定义与计算方法
 - 未插值AP：某个查询Q共有6个相关结果，排序返回了5篇相关文档，其位置分别是第1，第2，第5，第10，第20位，则 $AP=(1/1+2/2+3/5+4/10+5/20+0)/6$
 - 插值AP：事先选定插值点数并进行插值。例如，当我们计算11点平均时，计算在召回率分别为0,0.1,0.2,...,1.0的十一个点上的正确率求平均
 - 简化AP：只对返回的相关文档进行计算， $AP=(1/1+2/2+3/5+4/10+5/20)/5$ ，倾向那些快速返回结果的系统，**没有考虑召回率和补零的情况**

- 更多的评价准则：AP

- 三种AP的计算实例(假设共有十篇相关文档)

Example

1. d123 • (1/1)	6. d9 • (4/6)	11. d38 • (7/11)
2. d84	7. d511	12. d48
3. d56 • (2/3)	8. d129 • (5/8)	13. d250
4. d6 • (3/4)	9. d187	14. d113 • (8/14)
5. d8	10. d25 • (6/10)	15. d3

- 未插值的AP = $(1/1 + 2/3 + 3/4 + 4/6 + 5/8 + 6/10 + 7/11 + 8/14 + \underline{0} + \underline{0})/10$
- 11点插值的AP = $(\underline{1} + \underline{1/1} + 2/3 + 3/4 + 4/6 + 5/8 + 6/10 + 7/11 + 8/14 + 0 + 0)/\underline{11}$
- 简化的AP = $(1/1 + 2/3 + 3/4 + 4/6 + 5/8 + 6/10 + 7/11 + 8/14)/\underline{8}$

- 单查询评价

- 无序结果评价

- 有序结果评价

- 相关度分级

- 多查询评价

- 结果多样化评价

分级的必要性与考虑相关度加和的度量

- 先前的各种指标，都是基于相关/不相关的二元评判
- 然而在现实中，用户对文档的评价往往更为复杂，无法用二元简单概况
 - 例如，如果用户只是想了解《动物世界》这部电影的概况，那么百度百科即可。
 - 然而，如果用户是想观看《动物世界》这部电影，那么右侧图示的排序就相对较差，因为相关度更高的“在线观看”内容排序靠后。

[动物世界 百度百科](#)



类型：电影作品
 导演：韩延
 简介：《动物世界》是由上海儒意影视制作有限公司、上海火龙果影视制作有限公司、北京光线影业联合出品，韩延执导，李易峰、迈克尔·道格拉斯、周冬雨、曹炳琨、苏可、王戈等...
[剧情简介](#) [演职员表](#) [角色介绍](#) [音乐原声](#) [幕后花絮](#) [更多>>](#)
<https://baike.baidu.com/>

[动物世界](#) [动物世界全集视频 - CCTV1直播网](#)



6天前 - 29:43 [动物世界](#) 20191010 鳄鱼和它的邻居们(上) 发布:2019-10-10 首页上页 1/109 下页末页直播首页全部栏目节目表 ...
www.cctv1zhibo.com/don... - 百度快照

[动物世界 高清视频在线观看 爱奇艺](#)



2018年上映 | 130分钟 | 内地 | 国语
 导演：[韩延](#)
 主演：[李易峰](#) [迈克尔·道格拉斯](#)
 类型：[剧情](#) | [动作](#) | [冒险](#)
 简介：男主角郑开司，因为借给朋友钱而背负上了数百万债务。为偿还欠款，为了相依为命的植物人母亲、青梅... [更多>>](#)
[立即播放](#) 来源：[爱奇艺](#) [腾讯](#) [风行网](#) [优酷](#)

- 基础的相关度加和

- 累计增益 (Cumulative Gain, CG)

- 用于衡量位于位置1 到 p 的检索结果的相关度之和。

$$CG_p = \sum_{i=1}^p rel_i$$

- Rel 用于描述文档相关性，可以根据需求选取多个数值 / 级别。
- 较高的CG表明文档的整体相关性较高。
- 然而，由于CG并未考虑文档位置，并不能体现靠前部分文档的质量。

• 改进的相关度加和

- 如何区别位置的作用？直观想法：对不同位置赋予不同折损
- 折损累计增益 (Discounted Cumulative Gain, DCG)
 - 基本思想：若搜索算法把相关度高的文档排在后面，则应该给予惩罚。
 - 一般用log函数来表示这种惩罚，如 $\log(i+1)$ ， i 为文档位置
 - 往往有以下两种计算公式（后者采用指数，更突出相关性）：

$$• DCG_p = rel_1 + \sum_{i=2}^p \frac{rel_i}{\log_2(i)}$$

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

- 归一化的相关度加和

- 然而，DCG由于其随着长度单调非减的特性，仍具有其局限性
 - DCG与具体查询和结果列表的长度 p 有关，不利于不同算法之间的对比
 - 不同查询的结果有多有少，因此其DCG值无法实现相互比较
- 对DCG进行规范化：归一化折损累计增益 (Normalized DCG, NDCG)
 - 基本思路：将DCG除以完美结果下得到的理想结果，iDCG (ideal DCG)
 - 即： $NDCG = DCG / iDCG$
 - 其中，iDCG是根据文档根据相关性从大到小排序得到理想化的最优序列，并对此序列计算DCG值所得到的。

• NDCG计算实例

- 假设有10个文档，相关度为0-3之间，10个文档的得分依次如下：
 - 3, 2, 3, 0, 0, 1, 2, 2, 3, 0
- 理想的输出结果序列为：3, 3, 3, 2, 2, 2, 1, 0, 0, 0
 - 由此计算 iDCG依次为：3, 6, 7.89, 8.89, 9.75, 10.52, 10.88, 10.88, 10.88, 10.88
- 而与此同时，基本的DCG结果如下：
 - 3, 5, 6.89, 6.89, 6.89, 7.28, 7.99, 8.66, 9.61, 9.61 (单调非减特性)
- 由此可得 NDCG结果如下：1, 0.83, 0.87, 0.76, 0.71, 0.69, 0.73, 0.8, 0.88, 0.88
 - 可以看到任何查询结果位置的NDCG 值都规范化为[0,1]之间的值

- 单查询评价
 - 无序结果评价
 - 有序结果评价
 - 相关度分级
- 多查询评价
 - 结果多样化评价

- **从AP到MAP**

- 从单查询拓展至多查询评价，可以更全面地体现排序算法的综合性能
 - 如何对多查询的结果进行综合？
- MAP (Mean AP) , 对所有查询的AP求算数平均
- 例如：假设有一个检索系统
 - 对查询1 返回4 个相关网页，其rank 分别为1, 2, 4, 7
 - 对查询2 返回3 个相关网页，其rank 分别为1, 3, 5
 - 查询1 共有4 个相关文档，查询2 共有5个相关文档

查询1: $AP = (1/1 + 2/2 + 3/4 + 4/7)/4 = 0.83$

查询2: $AP = (1/1 + 2/3 + 3/5 + 0 + 0)/5 = 0.45$

$MAP = (0.83 + 0.45)/2 = 0.64$

←此处使用了未插值AP

- **MAP的变形: GMAP**

- MAP可以反映全部查询的综合效果，但在查询难度不平衡的条件下有误导

系统	主题	AP	提升	MAP
系统A	主题1	0.02	-	0.113
	主题2	0.03	-	
	主题3	0.29	-	
系统B	主题1	0.08	+300%	0.107
	主题2	0.04	+33.3%	
	主题3	0.20	-31%	

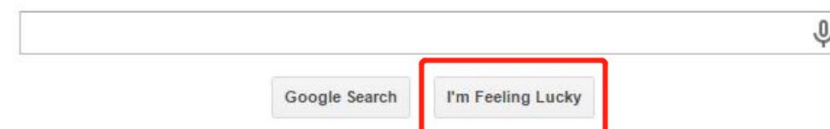
- 单纯从 MAP 来看，系统A 好于系统B 。
- 但是从每个查询来看，3 个 主题中有2 个主题，系统 B 都比 A 有提高，其中一个提高的幅度达到 300%

- **MAP的变形: GMAP**

- 通过引入基于几何平均值的GMAP (Geometric MAP) 解决这一问题
 - 削弱绝对数值的影响, 从而提升相对强弱的影响
- $GMAP = \sqrt[n]{\prod_{i=1}^n AP_i} = \exp(\frac{1}{n} \sum_{i=1}^n \ln AP_i)$
- 基于这一方法, 上述那个例子的结果可修正为:
 - $GMAP_a = 0.056$, $GMAP_b = 0.086$, 因此系统B更为出色
- 从这个例子可以看出, MAP与GMAP各有所长
 - 当各个查询间难度不均, 或存在较难排序的主题时, GMAP或许更合适

- **注重首个相关文档的MRR**

- 在许多查询任务中，用户只关心第一个相关的文档，越靠前越好
 - 该位置的倒数被称作Reciprocal Rank (RR)，数字越大效果越好
- 对多个查询所得的倒数排序求评价，即Mean RR, MRR
- 例如，两个查询，第一个查询的第一个相关文档在位置2，第二个查询的第一个相关文档在位置4
 - 则MRR为 $(1/2 + 1/4) / 2 = 3/8$
 - 即平均在 $8/3$ 的位置上找到第一个相关文档



- **MRR的拓展模型：ERR**

- 当用户发现了第一篇相关文档后，后面的内容可能就不再关注了
- 一篇文档可能被用户点击的概率大致估计为： $PP_r = R_r \prod_{i=1}^{r-1} (1 - R_i)$
 - 其中 R_r 表示位置为 r 的文档的相关度
- 由此，可以定义预期的倒数排序（Expected RR, ERR）如下：
 - $ERR = \sum_{r=1}^n \frac{1}{r} PP_r = \sum_{r=1}^n \frac{1}{r} R_r \prod_{i=1}^{r-1} (1 - R_i)$
 - 表示用户的需求被满足时停止的位置的倒数的期望

- **一个值得关注的指标：方差**
- 对于一个测试文档集合，检索系统常常对有的查询表现的很好，而对有的查询表现很差。
 - 先前的例子可以揭示这一现象，e.g., GMAP时的例子。
 - 通常情况下，一个检索系统对不同查询的方差，往往大于多个检索系统对相同查询的方差。
 - 由此可见，不同查询的难度差异较大，有些查询确实很难。

- 单查询评价
 - 无序结果评价
 - 有序结果评价
 - 相关度分级
- 多查询评价
- 结果多样化评价

为什么要考虑多样性

一方面，用户的单次搜索可能体现出多方面的需求

Query 1: "Free online Tetris"		Query 2: "Tetris game"	
×	Tetris Friends Online Games - Play Free Games Featuri... Play free online games featuring Tetris. Play single-player and ... http://tetrisfriends.com		Tetris Friends Online Games - Play Free Games Featuri... Play free online games featuring Tetris. Play single-player and ... http://tetrisfriends.com
×	Play Free Tetris Game Online Play this classic, original, Flash Tetris Game online for free. http://www.gametetris.com		Tetris game Free online game: Make lines with falling blocks! Russia's finest... http://www.play.vg/games/6-Tetris.html
	Free Tetris Game Free tetris game - Play free tetris games online, learn about tet... http://www.tetrislive.com	×	Tetris (Game Boy) - Wikipedia, the free encyclopedia Tetris was a pack-in title included with the Game Boy at the ha... http://en.wikipedia.org/wiki/Tetris_(handheld_game)
	4FreeOnlineGame.com - Free Online Tetris Game 4FreeOnlineGame - Free Online Tetris Game ... This is the all ... http://www.4freeonlinegame.com/Tetris	×	Tetris - non-stop puzzle action Tetris logo, Tetris theme song and Tetrminos are trademarks of... http://www.tetris.com
	Tetris - Play Tetris. Free online games © Adoption Media, LLC 1995 - 2010 This site should not subst... http://games.adoption.com/free-online-games/Tetris		Free Tetris Game Free tetris game - Play free tetris games online, learn about tetr... http://www.tetrislive.com

另一方面，用户搜索可能存在歧义，需要展示多方面内容加以确认

动物世界 央视网(cctv.com)

2019年9月28日 - CCTV-3综艺频道《动物世界》《动物世界》栏目已经走过20多年,通过专家的讲述、优美的画面、感人的故事去告诉观众、打动观众,使观众认识到我们不能没有...
tv.cctv.com/lm/dw... - 百度快照

动物世界|动物世界全集视频 - CCTV1直播网



栏目标题:动物世界 播放频道:CCTV-1综合 播出时间:每天00:20(除周二) 持续时间:30分钟 栏目介绍:《动物世界》栏目于1981年12月31日开播,主旨在于向电视观众介绍...
www.cctv1zhibo.com/don... - 百度快照

动物世界 百度百科



类型: 电影作品
导演: 韩延
简介: 《动物世界》是由上海儒意影视制作有限公司、上海火龙果影视制作有限公司、北京光线影业有限公司联合出品,韩延执导,李易峰、迈克尔·道格拉斯、周冬雨、曹炳琨、苏可、王戈等...
[剧情简介](#) [演职员表](#) [角色介绍](#) [音乐原声](#) [幕后花絮](#) [更多>>](#)
<https://baike.baidu.com/>

or

- **多样性的形式化定义**

- 基本形式：给定一个查询 q ，返回一个多样化的结果文档集合 $R(q)$ 。
- 其中， $R(q)$ 作为一个整体，应满足以下条件：
 - $R(q)$ 中所有的结果文档都与查询 q 本身有较大的相关性。
 - 总体上要有较小的冗余度，以覆盖 q 的不同方面。
- 核心思想：降低用户无法获得所需信息的风险
 - 尽可能确保排序靠前的结果中至少有一个结果满足用户的需求。

- **多样性的两种衡量方式**

- 总体需求：衡量不同文档之间的主题差异性。
- 一般而言，衡量方式有以下两种：
 - 隐式模型：只计算文档之间的差异性
 - 文档是什么内容，不会也无法进行详细考量
 - 显式模型：更加具体地考量文档所对应的用户意图
 - 会从文档中抽取主题，并显式地实现主题的多样化

- 隐式模型代表：MMR

- 最大边界相关性 (Maximal Marginal Relevance, MMR)
 - 最早的相关性衡量模型，由Carbonell和Goldstein在1998年提出

$$MMR^{def} = \operatorname{Argmax}_{d_i \in R|S} [\lambda P(d_i | q) - (1 - \lambda) \max_{d_j \in S} P(d_i | d_j)]$$

- 该公式由两部分组成，其中：
 - 前半部分表示文档集与查询 q 的相似性
 - 后半部分表示文档之间的多样性
 - λ 用于调节两部分之间的比例
 - 文档可以采用tf-idf，或者word2vec等文本表征工具进行表征

- 显式模型代表：FM-LDA

- Facet model with LDA (FM-LDA)

- 由Carterette与Chandar在2009年提出，考虑文档的不同子主题（Facet）

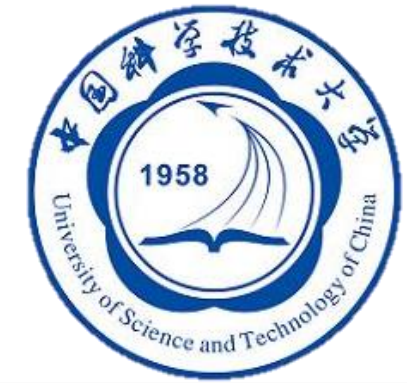
$$L(y_i|F, D) = \prod_{j=1}^m \left(1 - \prod_{i=1}^n (1 - p(f_j \in d_i))^{y_i} \right)$$

- 其中， $p(f_j \in d_i)$ 表示文档 D_i 包含主题 F_j 的概率， $Y_i=1$ 表示某文档被选中
- 由此，联乘部分表示主题 F_j 至少被一个文档所涵盖的概率
- 当有约束 $\sum y_i \leq l$ 时，返回的文档数量被限定为 l 篇

本章小结

结果评价

- 面向单次查询的结果评估
 - 无序结果评估：准确率、召回率、F值等
 - 有序结果评估：P@N、R@N、NDCG等
- 面向多次查询的结果评估
 - MAP、GMAP、MRR、ERR等
- 结果多样化评估：隐式与显式评估



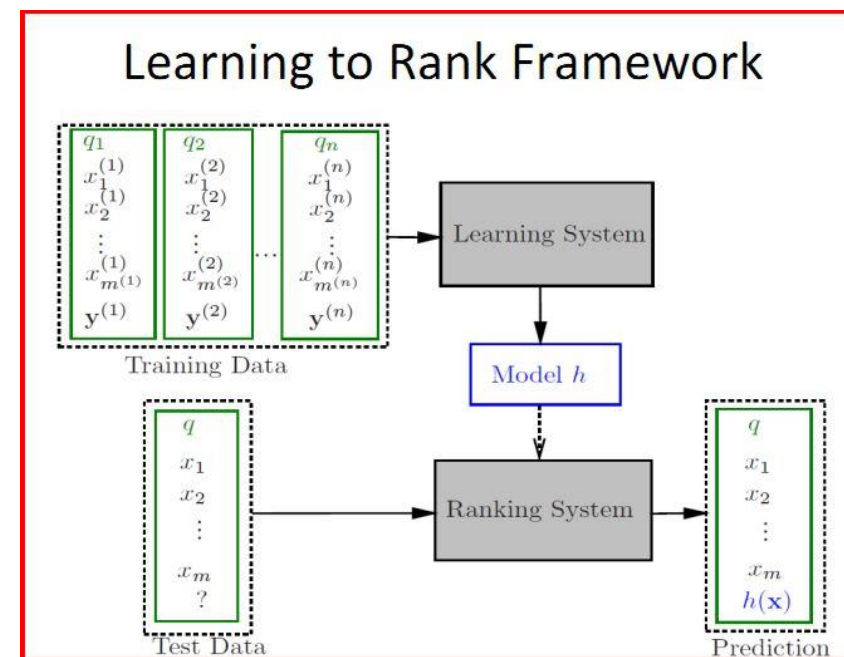
Web信息处理与应用

第六节 排序（下）

徐童 2021.10.25

- 从网页排序到排序学习

- 对于网页内容的排序问题，本质上是一个学习排序器（Ranker）的问题
 - 输入网页（文档）内容及查询，输出该网页合适的排序位置
 - 排序学习是个有监督学习问题
 - 基于已知的排序（如用户反馈）
 - 为新网页-查询对给出排序



- 从网页排序到排序学习

- 三类常见的排序学习算法

- Pointwise, 将排序退化为分类或回归问题
 - 输出: 网页对应的分类 (有序)、回归值或有序回归值
- Pairwise: 比较一对网页之间的相关度, e.g., 相关 > 不相关
 - 输出: 网页对之间的偏序关系
- Listwise: 对整个网页集合进行排序
 - 输出: 整个集合的完整排序, 往往依赖特定排序指标

- **Pointwise 算法**
- Pairwise 算法
- Listwise 算法

- **Pointwise类排序算法**

- 基本假设：训练样本中的任何一个查询-文档对，都可以映射到一个分值或一个有序类别（如优良中差）
- 相应的，给定一个查询-文档对，Pointwise LTR将试图预测其分值/类别。
- 常见的模型类别包括：
 - 回归，将查询-文档对映射到具体分数
 - 分类，将排序问题转化为一个面向有序类别的二分类/多分类问题
 - 有序回归，在映射到具体分数的同时保持样本之间的有序关系

- **Pointwise类排序算法**

- 第一类方法：回归方法 (Regression)
- 基于回归算法的一般损失函数如下所示：

$$L(f; x_j, y_j) = (y_j - f(x_j))^2$$

- 该类方法实现较为简单，但缺陷也较为明显。
 - 最大的缺陷在于没有对文档间的排序添加约束。
 - 因此，可能为降低Loss所误导而错判文档间的顺序。

- **Pointwise类排序算法**

- 第二类方法: 分类方法 (Classification)
- 采用分类方法, 将查询-文档对映射到二分类 (相关/不相关) 或多分类
 - 代表性算法: Discriminative Model for IR (R. Nallapati, SIGIR 2004)
 - 通过抽取特征表征文档, 相关文档被视作正样本, 不相关为负样本

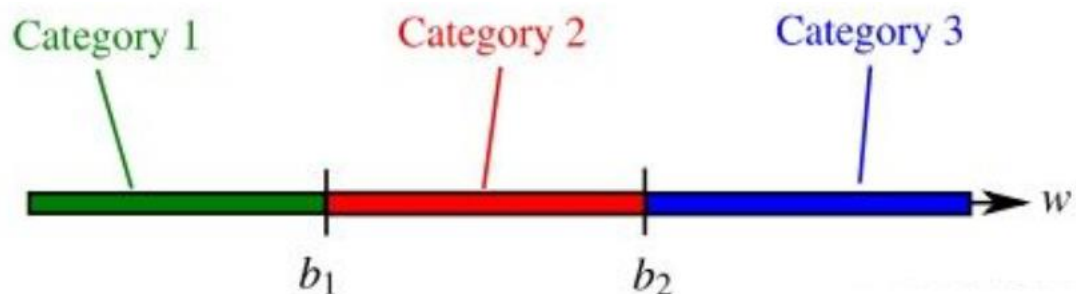
	Feature		Feature
1	$\sum_{q_i \in Q \cap D} \log(c(q_i, D))$	4	$\sum_{q_i \in Q \cap D} (\log(\frac{ C }{c(q_i, C)}))$
2	$\sum_{i=1}^n \log(1 + \frac{c(q_i, D)}{ D })$	5	$\sum_{i=1}^n \log(1 + \frac{c(q_i, D)}{ D } idf(q_i))$
3	$\sum_{q_i \in Q \cap D} \log(idf(q_i))$	6	$\sum_{i=1}^n \log(1 + \frac{c(q_i, D)}{ D } \frac{ C }{c(q_i, C)})$

- 有关分类的常用技术将在 “第12讲: 分类算法” 介绍

- **Pointwise类排序算法**

- 第三类方法：有序回归方法 (Ordinal Regression)

- 某种意义上，相当于利用回归方法求解有序多分类问题。



- 其中，文档的标签可通过如下公式推测：

$$\hat{y}_j = \arg \min_k \{w^T x_j - b_k < 0\}.$$

- 投影后，位于数轴上的区间决定了文档的标签。

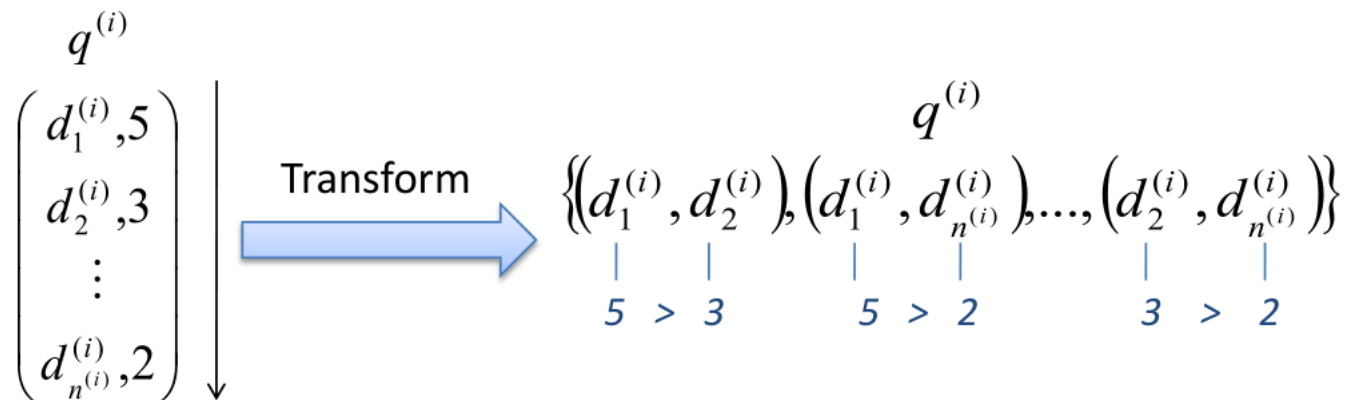
- **Pointwise类排序算法**

- Pointwise类排序算法可以简单且广泛地套用已有回归、分类算法
- 然而，其局限性也较为明显
 - 首先，Pointwise类方法往往更为注重文档的相关度得分，而并不注重文档之间的相关性排序
 - Pairwise类方法的出现，为解决这一问题提供了新的手段
 - 其次，不同查询所对应的文档，尤其相关文档数量不同，对损失函数的贡献也各不相同，一定程度上影响效果

- Pointwise 算法
- **Pairwise 算法**
- Listwise 算法

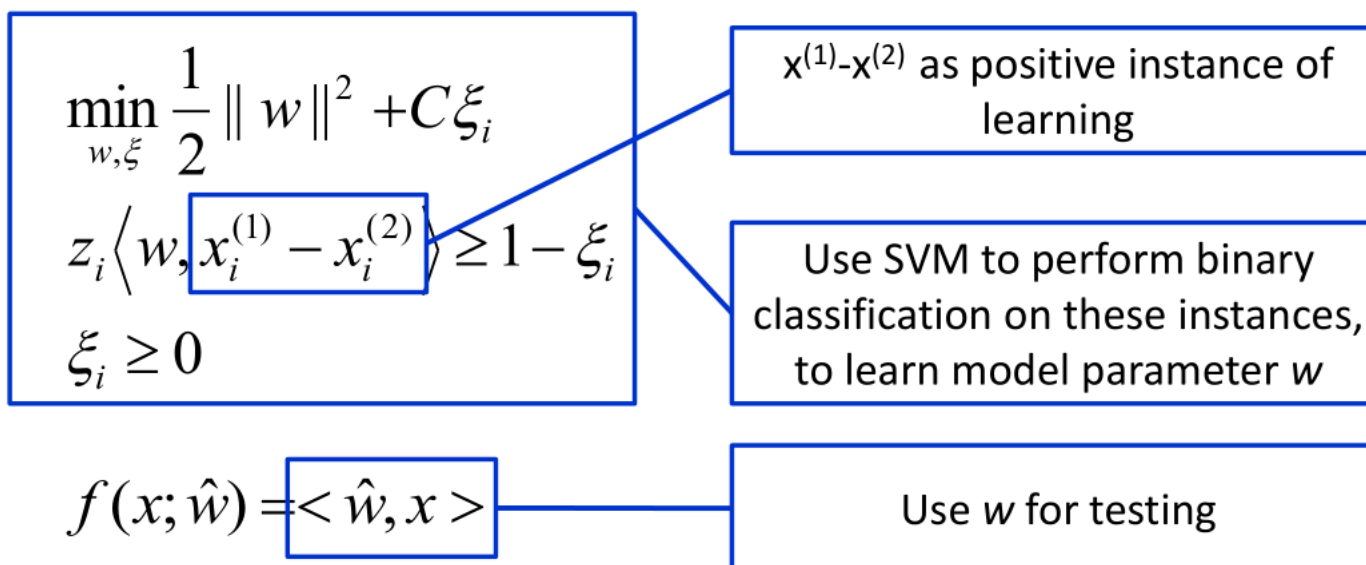
• Pairwise类排序算法

- 基本假设：同样是将排序问题转化为分类问题（二分类或三分类）
 - 每次比较一个查询与两个文档，衡量两个文档的偏序（Partial Order）
 - 分类器的目的在于判断哪个文档应该排在前面
 - 对应的标签为 $\{1, -1\}$ 或 $\{1, 0, -1\}$ （0表示两个文档可以并列）



- **Pairwise类排序算法**

- 代表性方法: Ranking SVM (R. Herbrich, et al. ICANN 1999)
 - 转化为分类问题后, 该类算法同样可以借鉴常用的分类方法
 - 例如, 通过改进经典的SVM算法实现二分类 (如下, 当D1优于D2)



- **Pairwise类排序算法**

- 相比于Pointwise类算法，Pairwise类排序算法通过衡量样本之间的偏序关系，实现了从绝对相关性（分值）到相对偏序的改进
- 然而，Pairwise类算法也具有自己的缺陷：
 - 首先，两两成对导致样本数大为提升，计算资源开支增加
 - 其次，Pairwise类算法仍然受样本不平衡问题的影响
 - 最后，Pairwise类算法无法体现全局排序的合理性
 - 由此引出了最后一类算法：Listwise类算法

- Pairwise类排序算法

- 局限性：样本不平衡性的影响
 - 如果不同Query的Doc数量相差很大，从偏序的对来说，Doc（对）数量较多的Query将掩盖其他的Query，从而产生干扰

		Case 1	Case 2
Document pairs of q_1	correctly ranked	770	780
	wrongly ranked	10	0
	Accuracy	98.72%	100%
Document pairs of q_2	correctly ranked	10	0
	wrongly ranked	0	10
	Accuracy	100%	0%
overall accuracy	document level	98.73%	98.73%
	query level	99.36%	50%

- **Pairwise类排序算法**

- 局限性：算法无法体现全局排序
 - 某个成对文档排序错误，发生在不同位置，在Pairwise类条件下是等价的
 - 例如：7篇文档，相关性从1到3，理想状态下的排序为3 2 2 1 1 1 1
 - 假如有如下两个排序：
 - Rank A：2 3 2 1 1 1 1 Rank B：3 2 1 2 1 1 1
 - 这两个排序，在Pairwise类算法中是等价的，然而，后者显然更合理
 - 如前所述，用户往往更关心排在靠前位置的文档

- Pointwise 算法
- Pairwise 算法
- **Listwise 算法**

- **Listwise类排序算法**

- 基本思路：直接面向整体排序结果进行优化
 - 直接将排序的完整队列作为学习的对象
- 通常情况下，解决这一问题采用以下两种思路
 - 直接采用某种IR指标对排序进行优化
 - 直接设计面向完整排序的损失函数

- **Listwise类排序算法**

- 代表性思路：采用某种IR指标对排序进行优化
 - 此类方法较为直观，且可以直接优化所获得排序结果的衡量指标
 - 然而，该类方法也面临明显的挑战：
 - 大多数排序指标，如NDCG等，因为与排序相关，属于非光滑、不可微的函数。
 - 因此，传统的优化方法很难直接应用于该问题。

- **Listwise类排序算法**

- 一种有趣的解决方案：AdaRank (J. Xu, et al. SIGIR 2007)
 - AdaBoost的变形，采用弱排序器的组合来实现全局排序。
 - AdaBoost是解决分类问题的经典算法，其思路大致如下：
 1. 生成一组弱分类器（如仅用一维特征进行分类）
 2. 根据结果更新分类器和样本的权重
 - 效果更好的分类器，权重更高；分类错误的样本，权重更高
 3. 面向权重更新后的样本，训练新的弱分类器
 4. 最终，按照权重整合所有的弱分类器，形成一个完整的强分类器

• Listwise类排序算法

- 将AdaBoost的思路迁移至排序学习问题，即AdaRank
 - 借鉴了AdaBoost的设定及步骤。
 - 基于部分特征和指定指标（如NDCG），得到弱排序器。
 - 通过调节权重，重点修正那些在先前排序中排序错误的文档。
 - 回避了直接优化全局NDCG的问题。

Input: $S = \{(q_i, \mathbf{d}_i, \mathbf{y}_i)\}_{i=1}^m$, and parameters E and T
Initialize $P_1(i) = 1/m$.

For $t = 1, \dots, T$

- Create weak ranker h_t with weighted distribution P_t on training data S .
- Choose α_t

$$\alpha_t = \frac{1}{2} \cdot \ln \frac{\sum_{i=1}^m P_t(i) \{1 + E(\pi(q_i, \mathbf{d}_i, h_t), \mathbf{y}_i)\}}{\sum_{i=1}^m P_t(i) \{1 - E(\pi(q_i, \mathbf{d}_i, h_t), \mathbf{y}_i)\}}.$$

- Create f_t

$$f_t(\vec{x}) = \sum_{k=1}^t \alpha_k h_k(\vec{x}).$$

- Update P_{t+1}

$$P_{t+1}(i) = \frac{\exp\{-E(\pi(q_i, \mathbf{d}_i, f_t), \mathbf{y}_i)\}}{\sum_{j=1}^m \exp\{-E(\pi(q_j, \mathbf{d}_j, f_t), \mathbf{y}_j)\}}.$$

End For

Output ranking model: $f(\vec{x}) = f_T(\vec{x})$.

- **Listwise类排序算法**

- 通常情况下，由于全局优化的作用，Listwise类排序算法可以取得相比于Pointwise和Pairwise类算法更好的效果。
- 然而，Listwise类算法也会面临一些小的挑战，例如两个网页并列的情况
 - 相当于Pairwise中文档对标签为0的情况

- **Learning To Rank 工具包**
- TensorFlow: TF-Ranking <https://github.com/tensorflow/ranking>
- PyTorch: allRank <https://github.com/allegro/allRank>
- 提供不同损失函数, pointwise, pairwise, listwise
- 提供不同排序指标, MRR, NDCG 等

本章小结

排序学习

- Pointwise 算法
 - 回归、分类（二分类/多分类）、有序回归、关联规则
- Pairwise 算法
 - 从绝对分值到相对偏序
- Listwise 算法
 - 面向IR指标的优化、直接定义Listwise Cost的优化