

# 信息组织与检索

## 检索评价

主讲人：张蓉

华东师范大学 数据科学与工程学院

# 本讲内容

---

- 信息检索的评价指标
  - 不考虑序的检索评价指标(即基于集合)
  - 考虑序的评价指标

# 关于评价

---

- 评价无处不在，也很必要
  - 工作、生活、娱乐、找对象、招生
- 评价很难，但是似乎又很容易
  - 人的因素、标准、场景
- 评价是检验学术进步的唯一标准，也是杜绝学术腐败的有力武器
- 小明宣称自己是某项目世界最厉害的，却从不参加公开比赛

# 从竞技体育谈起

---

- 世界记录
  - 男子110米栏世界记录： 12.80 Aries Merritt
  - 男子马拉松世界记录： 2小时01分39秒 Eliud Kipchoge
- 评价要公平！
  - 环境要基本一致： 天气、风速、跑道等等
  - 比赛过程要一样： 竞走中的犯规
  - 指标要一样： 速度、耐力
- 国际田联规定了顺风风速的界限，即每秒2米，顺风超过这个速度称为超风速，运动员在超风速下所创成绩有效，不过不列为任何纪录。

# 为什么要评估IR?

---

- 通过评估可以评价不同技术的优劣，不同因素对系统的影响，从而促进本领域研究水平的不断提高
  - 类比：110米栏各项技术---起跑、途中跑、跨栏、步频、冲刺等等
- 信息检索系统的目标是较少消耗情况下尽快、全面返回准确的结果。

IR中评价什么?

# IR中评价什么？

---

- 效率 (Efficiency)—可以采用通常的评价方法
  - 时间开销
  - 空间开销
  - 响应速度
- 效果 (Effectiveness)
  - 返回的文档中有多少相关文档
  - 所有相关文档中返回了多少
  - 返回得靠不靠前
- 其他指标
  - 覆盖率 (Coverage)
  - 访问量
  - 数据更新速度

# 如何评价效果？

---

- 相同的文档集合，相同的查询主题集合，相同的评价指标，不同的检索系统进行比较。
  - **The Cranfield Experiments**, Cyril W. Cleverdon, 1957 – 1968 (上百篇文档集合)
  - **SMART System**, Gerald Salton, 1964-1988 (数千篇文档集合)
  - **TREC(Text REtrieval Conference)**, Donna Harman, 美国标准技术研究所, 1992 - (上百万篇文档), 信息检索的“奥运会”

# 评价任务的例子

- 两个系统，一批查询，对每个查询每个系统分别得到一些结果。目标：哪个系统好？

系统&查询	1	2	3	4	...
系统1， 查询1	d3	d6	d8	d10	
系统1， 查询2	d1	d4	d7	d11	
系统2， 查询1	d6	d7	d3	d9	
系统2， 查询2	d1	d2	d4	d13	



# 评价的几部分

---

- 评价指标：某个或某几个可衡量、可比较的值
- 评价过程：设计上保证公平、合理  
(benchmarking)

# 评价指标分类

---

- 对单个查询进行评估的指标
  - 在单个查询上检索系统的得分
- 对多个查询进行评估的指标
  - 在多个查询上检索系统的得分

# 评价指标分类

---

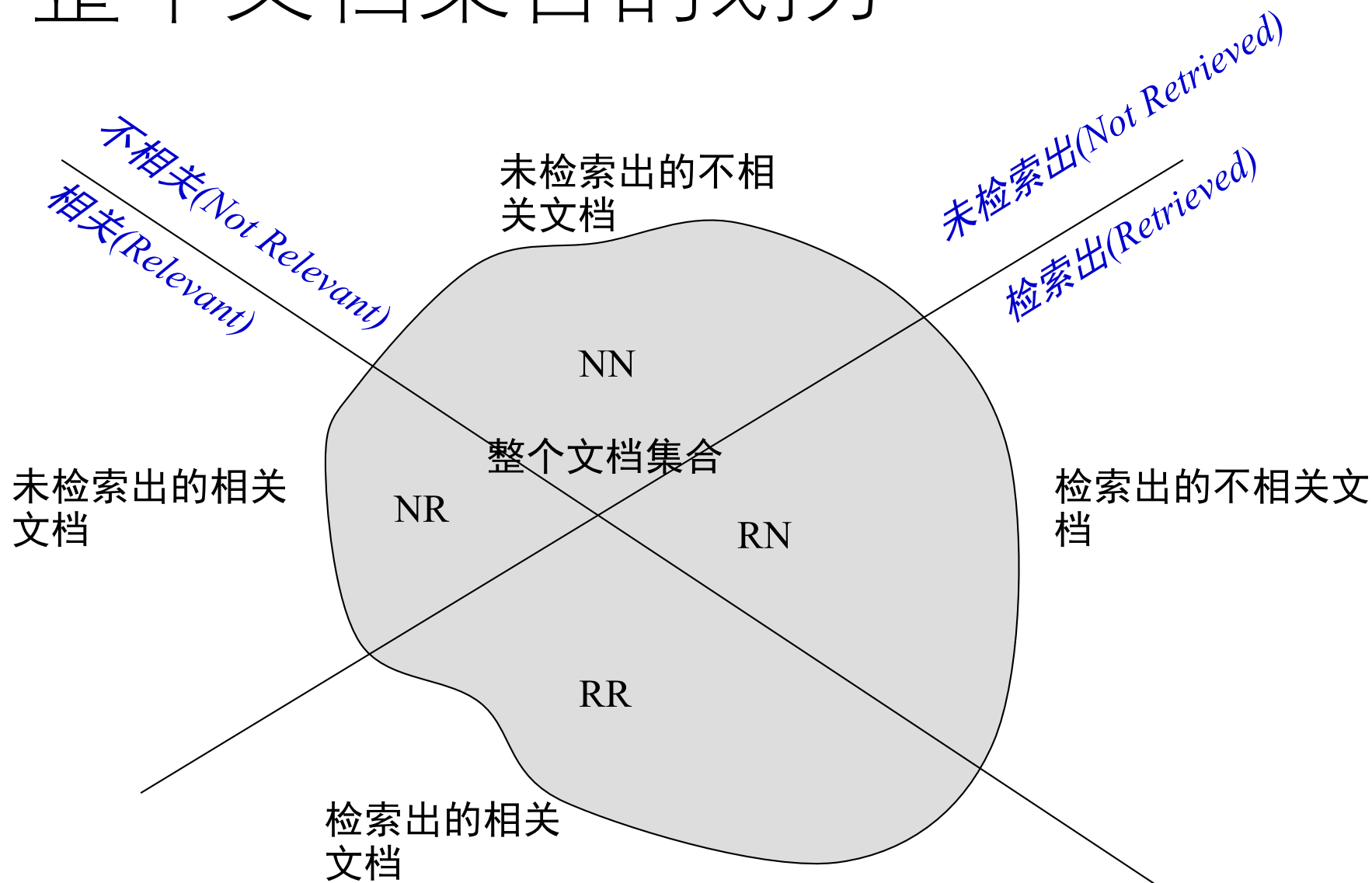
- 对单个查询进行评估的指标 ←
  - 在单个查询上检索系统的得分
- 对多个查询进行评估的指标
  - 在多个查询上检索系统的得分

## 回到例子

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

对于查询1的标准答案集合 {d3,d4,d6,d9}

# 整个文档集合的划分



# 评价指标

---

- **召回率 (Recall)**:  $RR / (RR + NR)$ , 返回的相关结果数占实际相关结果总数的比率, 也称为**查全率**,  $R \in [0,1]$
- **正确率 (Precision)**:  $RR / (RR + RN)$ , 返回的结果中真正相关结果的比率, 也称为**查准率**,  $P \in [0,1]$
- 两个指标分别度量检索效果的某个方面, 忽略任何一个方面都有失偏颇。
  - **两个极端情况**: 返回有把握的1篇,  $P=100\%$ , 但R极低; 全部文档都返回,  $R=1$ , 但P极低

# 四种关系的矩阵表示

真正相关文档  $RR+NR$     真正不相关文档

系统判定相关  
 $RR+RN$  (检索出)

系统判定不相关  
(未检索出)

$RR$	$RN$
$NR$	$NN$

$Ret = RR+RN$

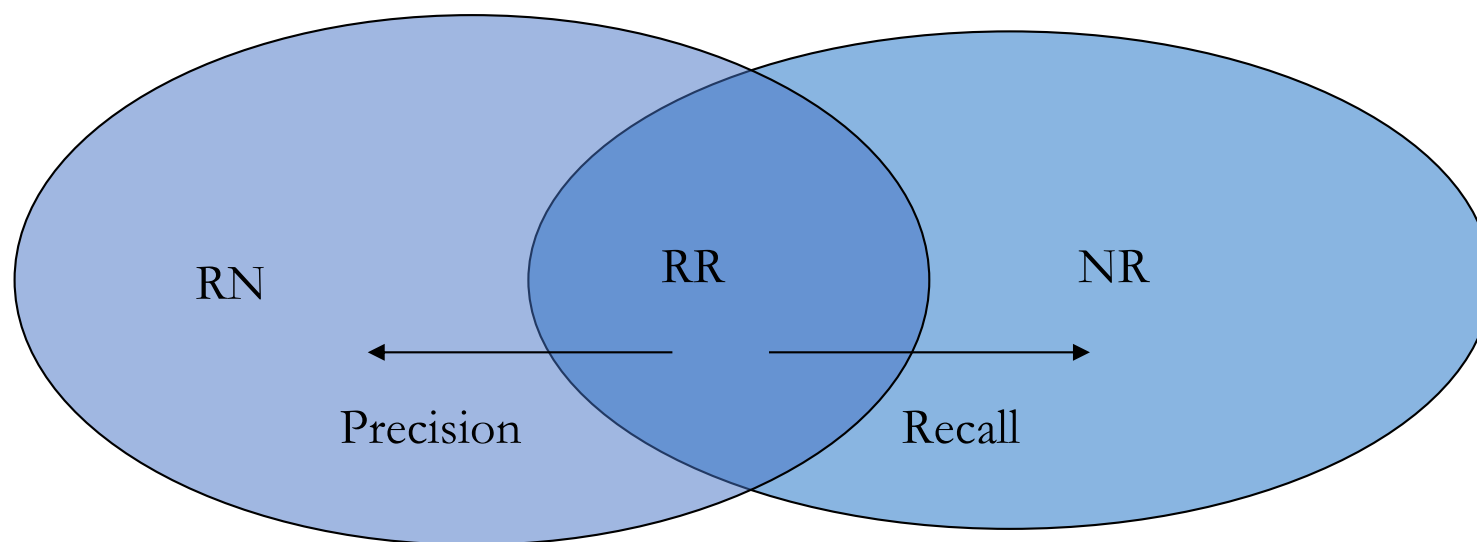
Precision

Recall

$Ans = RR+NR$

# 基于集合的图表示

---



返回结果Ret

标准答案Ans



# 回到例子

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

对于查询1的标准答案集合 {d3,d4,d6,d9}

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

# 提问

---

- 查询Q，本应该有100篇相关文档，某个系统返回200篇文档，其中80篇是真正相关的文档
- 提问：P/R怎么计算？

# 提问

---

- 一个例子：查询Q，本应该有100篇相关文档，某个系统返回200篇文档，其中80篇是真正相关的文档
- $\text{Recall} = 80/100 = 0.8$
- $\text{Precision} = 80/200 = 0.4$
- 结论：召回率较高，但是正确率较低

# 正确率和召回率的应用领域

---

- 拼写校对
- 中文分词
- 文本分类
- 人脸识别
- .....

# 关于正确率和召回率的讨论(1)

---

- “宁可错杀一千，不可放过一人” → 偏重召回率，忽视正确率。冤杀太多。
- 判断是否有罪：
  - 如果没有证据证明你无罪，那么判定你有罪。 → 召回率高，有些人受冤枉
  - 如果没有证据证明你有罪，那么判定你无罪。 → 召回率低，有些人逍遥法外

# 关于正确率和召回率的讨论(2)

---

- 虽然Precision和Recall都很重要，但是不同的应用、不同的用户可能会对两者的要求不一样。因此，实际应用中应该考虑这点。
  - 垃圾邮件过滤：宁愿漏掉一些垃圾邮件，但是尽量少将正常邮件判定成垃圾邮件。
  - 有些用户希望返回的结果全一点，他有时间挑选；有些用户希望返回结果准一点，他不需要结果很全就能完成任务。

## P/R指标的方差

- 对于一个测试文档集来说，某些信息需求上效果很差 (比如,在  $R = 0.1$  点上  $P = 0.2$ )，但是在一些其他需求上又相当好 (如在  $R = 0.1$  点上  $P = 0.95$ )
- 实际上，同一系统在不同查询上的结果差异往往高于不同系统在同一查询上的结果
- 也就是说，存在容易的信息需求和难的信息需求
- 好比有的人擅长回答几何问题，有的擅长回答数论问题

# 提问

- 正确率和召回率的定义或者计算有什么问题或不足或者难点？

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

对于查询1的标准答案集合 {d3,d4,d6,d9}

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



# 正确率和召回率的问题

---

- 召回率难以计算
  - 解决方法：Pooling方法，或者不考虑召回率
- 两个指标分别衡量了系统的某个方面，但是也为比较带来了难度，究竟哪个系统好？大学最终排名也只有一个指标。
  - 解决方法：单一指标，将两个指标融成一个指标
- 两个指标都是基于(无序)集合进行计算，并没有考虑序的作用
  - 举例：两个系统，对某个查询，返回的相关文档数目一样都是10，但是第一个系统是前10条结果，后一个系统是最后10条结果。显然，第一个系统优。但是根据上面基于集合的计算，显然两者指标一样。
  - 解决方法：引入序的作用

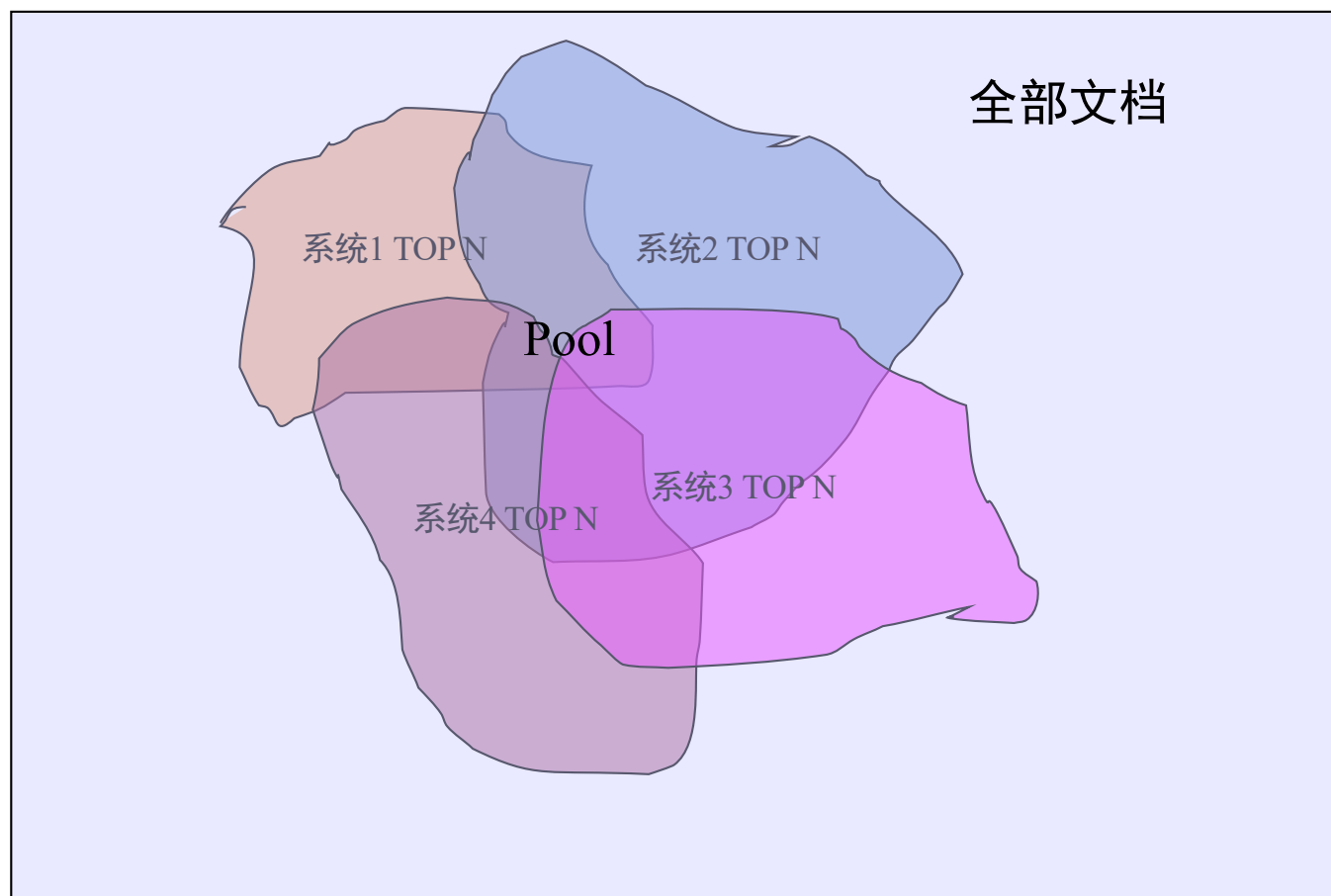
# 关于召回率的计算

---

- 对于大规模语料集合，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
- 缓冲池(Pooling)方法：对多个检索系统的Top N个结果组成的集合进行人工标注，标注出的相关文档集合作为整个相关文档集合。这种做法被验证是可行的(可以比较不同系统的相对效果)，在TREC会议中被广泛采用。

# 4个系统的Pooling

---



# P和R融合

$$P=0.4 \ R=0.4 \ F1=?$$

$$P=0.6 \ R=0.2 \ F2$$

$$P=0.5 \ R=0.3 \ F3$$

$$F1 > F3 > F2$$

- F值(F-measure): 召回率R和正确率P的调和平均值, if  $P=0$  or  $R=0$ , then  $F=0$ , else 采用下式计算:

$$F = \frac{2}{\frac{1}{P} + \frac{1}{R}} = \frac{2PR}{P+R} \quad (P \neq 0, R \neq 0)$$

- $F_\beta$ : 表示召回率的重要程度是正确率的 $\beta(>=0)$ 倍,  $\beta > 1$ 更重视召回率,  $\beta < 1$ 更重视正确率

$$F_\beta = \frac{(1+\beta^2)PR}{\beta^2 P + R} \quad (P \neq 0, R \neq 0)$$

- E(Effectiveness)值: 召回率R和正确率P的加权平均值,  $b > 1$ 表示更重视P,  $E = 1 - F_\beta$ ,  $b^2 = 1/\beta^2$

$$E = 1 - \frac{1+b^2}{\frac{b^2}{P} + \frac{1}{R}} \quad (P \neq 0, R \neq 0)$$

# 为什么使用调和平均计算F值

- 为什么不使用其他平均来计算F，比如算术平均
- 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F值就不低于50%，这有些过高。
- 做法：不管是P还是R，如果十分低，那么结果应该表现出来，即这样的情形下最终的F值应该有所惩罚
- 采用P和R中的最小值可能达到上述目的
- 但是最小值方法不平滑而且不易加权
- 基于调和平均计算出的F值可以看成是平滑的最小值函数

# 精确率(Accuracy)

- 精确率是所有判定中正确的比率

- $\text{accuracy} = (\text{RR} + \text{NN}) / (\text{RN} + \text{RR} + \text{NR} + \text{NN})$

- 为什么通常使用P、R、F而不使用精确率？
- Web信息检索当中精确率为什么不可用？

# 课堂练习

- 计算P、R、F1

	相关	不相关
返回	18	2
未返回	82	1,000,000,000

- 下面的一个搜索引擎无论对于什么查询都返回0结果，为什么该引擎例子表明使用精确率是不合适的？



# 精确率不适合IR的原因

- 由于和查询相关毕竟占文档集的极少数，所以即使什么都不返回也会得到很高的精确率
- 什么都不返回可能对大部分查询来说可以得到 99.99%以上的精确率
- 信息检索用户希望找到某些文档并且能够容忍结果中有一定的不相关性
- 返回一些即使不好的文档也比不返回任何文档强
- 因此，实际中常常使用P、R和F1，而不使用精确率



# 引入序的作用(1)

---

- **R-Precision**: 检索结果中, 在所有相关文档总数位置上的准确率, 如某个查询的相关文档总数为80, 则计算检索结果中在前80篇文档的正确率。

系统1, 查询1 d3 ✓ d6 ✓ d8 d10 d11  
系统2, 查询1 d6 ✓ d7 d2 d9 ✓

对于查询1的标准答案集合 {d3,d4,d6,d9}

$$R-P1=2/4 \quad R-P2=2/4$$

# 引入序的作用(2)

---

- 正确率-召回率 曲线(precision versus recall curve)
  - 检索结果以排序方式排列，用户不可能马上看到全部文档，因此，在用户观察的过程中，正确率和召回率在不断变化(vary)。
  - 可以求出在召回率分别为0%,10%,20%,30%,...,90%,100%上对应的正确率，然后描出图像
  - 在上面的曲线对应的系统结果更好

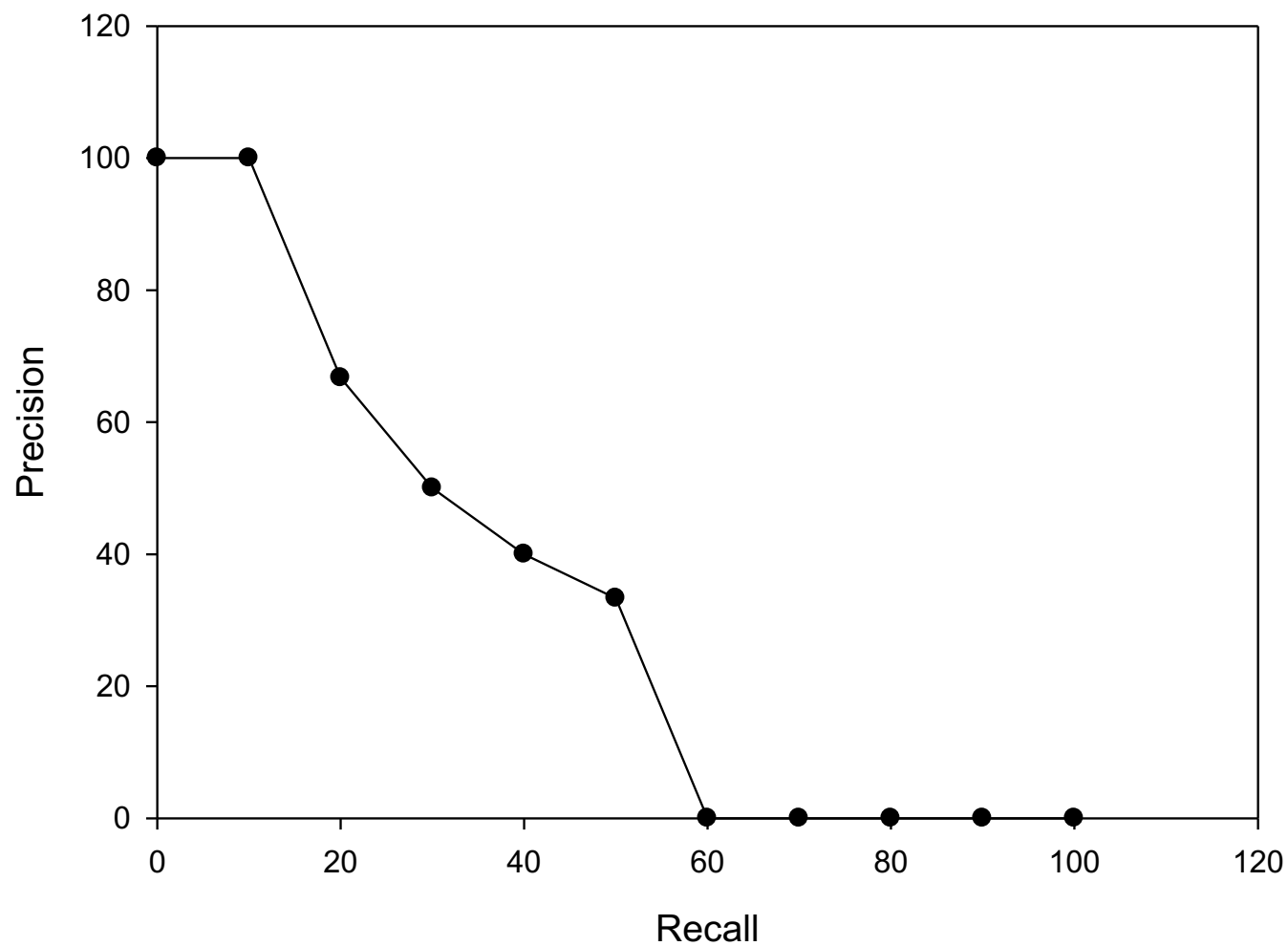
# P-R曲线的例子

- 某个查询q的10个标准答案集合为：  
 $R_q = \{d_3, d_5, d_9, d_{25}, d_{39}, d_{44}, d_{56}, d_{71}, d_{89}, d_{123}\}$
- 某个IR系统对q的检索结果如下：

1. d123 R=0.1,P=1	6. d9 R=0.3,P=0.5	11. d38
2. d84	7. d511	12. d48
3. d56 R=0.2,P=0.67	8. d129	13. d250
4. d6	9. d187	14. d113
5. d8	10. d25 R=0.4,P=0.4	15. d3 R=0.5,P=0.33

# P-R 曲线

Precision-recall « 曲线



# P-R 曲线的插值问题

---

- 对于前面的例子，假设  $R_q = \{d3, d56, d129\}$ 
  - 3. d56  $R=0.33, P=0.33$ ; 8. d129  $R=0.67, P=0.25$ ; 15. d3  $R=1, P=0.2$

1. d123	6. d9	11. d38
2. d84	7. d511	12. d48
3. d56 $R=0.33, P=0.33$	8. d129 $R=0.67, P=0.25$	13. d250
4. d6	9. d187	14. d113
5. d8	10. d25	15. d3 $R=1, P=0.2$

- 
- 不存在10%, 20%, ..., 90%的召回率点，而只存在33.3%, 66.7%, 100%三个召回率点
  - 在这种情况下，需要利用存在的召回率点对不存在的召回率点进行插值(interpolate)
  - 对于 $t\%$ ，如果不存在该召回率点，则定义 $t\%$ 的正确率为从 $t\%$ 到100%( $\geq t\%$ )中最大的正确率值。

- 
- 对于示例，插值后准确率为多少？
    - 0%,10%,20%,30%上正确率为0.33,
    - 40%~60%对应0.25,
    - 70%以上对应0.2

1. d123	6. d9	11. d38
2. d84	7. d511	12. d48
3. d56 R=0.33,P=0.33	8. d129 R=0.67,P=0.25	13. d250
4. d6	9. d187	14. d113
5. d8	10. d25	15. d3 R=1,P=0.2

# P-R 曲线图

插值前:

No3. d56  $R=0.33, P=0.33$ ;

No8. d129  $R=0.66, P=0.25$ ;

No15. d3  $R=1, P=0.2$

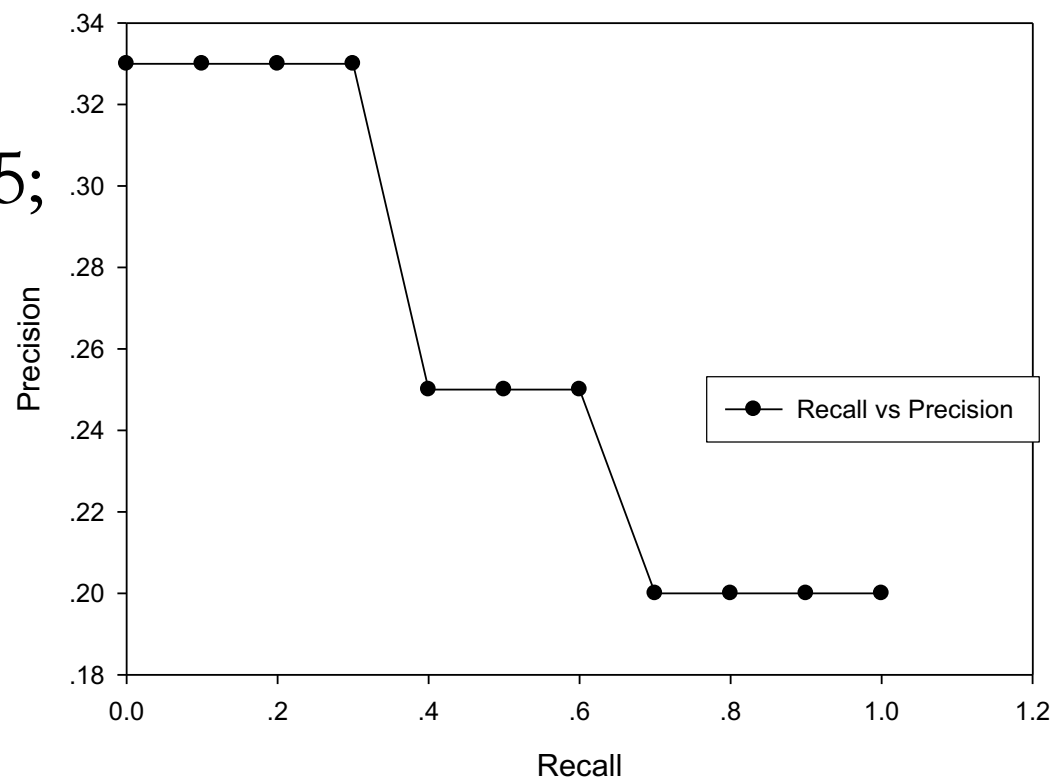
插值后:

0%~30% 正确率为 0.33

40%~60% 对应 0.25

70% 以上 对应 0.2

Precision-Recall « 06





# P-R的优缺点

---

- 优点：
  - 简单直观
  - 既考虑了检索结果的覆盖度，又考虑了检索结果的排序情况
- 缺点：
  - 单个查询的P-R曲线虽然直观，但是难以明确表示两个查询的检索结果的优劣

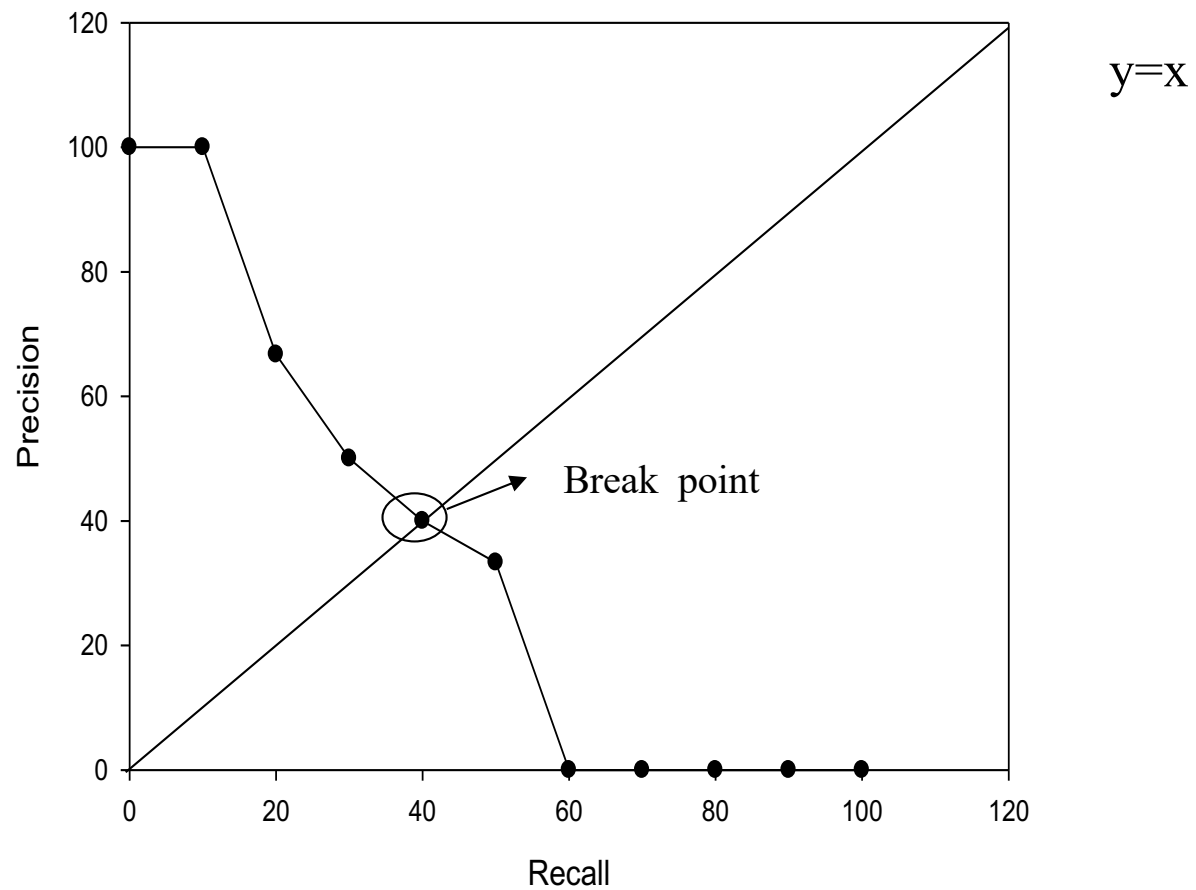
# 基于P-R曲线的单一指标

---


- Break Point: P-R曲线上  $P=R$  的那个点
  - 这样可以直接进行单值比较
- 11点平均正确率(11 point average precision): 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, 等价于插值的AP

# P-R曲线中的break point

Precision-recall « 图



# 引入序的作用(3)

- 平均正确率(Average Precision, AP): 对不同召回率点上的正确率进行平均
  - 未插值的AP: 某个查询Q共有6个相关结果, 某系统排序返回了5篇相关文档, 其位置分别是第1, 第2, 第5, 第10, 第20位, 则
$$AP = (1/1 + 2/2 + 3/5 + 4/10 + 5/20 + 0) / 6$$
第6文档
  - 插值的AP: 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, 等价于11点平均
  - 只对返回的相关文档进行计算的AP,
$$AP = (1/1 + 2/2 + 3/5 + 4/10 + 5/20) / 5$$
, 倾向那些快速返回结果的系统, 没有考虑召回率

# 不考虑召回率

---

- Precision@N: 在第N个位置上的正确率，对于搜索引擎，大量统计数据表明，大部分搜索引擎用户只关注前一、两页的结果，因此，P@10, P@20对大规模搜索引擎来说是很好的评价指标

# 回到例子

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

查询1及查询2的标准答案集合分别为  $\{d3, d4, d6, d9\}$   $\{d1, d2, d13\}$

系统1查询1:  $P@2=1$ ,  $P@5=2/5$ ; 系统1查询2:  $P@2=1/2$ ,  $P@5=2/5$ ;

系统2查询1:  $P@2=1/2$ ,  $P@5=2/5$ ; 系统2查询2:  $P@2=1$ ,  $P@5=3/5$

# 评价指标分类

---

- 对单个查询进行评估的指标
  - 对单个查询得到一个结果
- 对多个查询进行评估的指标 ←
  - 在多个查询上检索系统的得分求平均

# 评价指标(9)

---

- 平均的求法：
  - 宏平均(Macro Average): 对每个查询求出某个指标，然后对这些指标进行算术平均
  - 微平均(Micro Average): 将所有查询视为一个查询，将各种情况的文档总数求和，然后进行指标的计算
    - 如：Micro Precision=(对所有查询检出的相关文档总数)/(对所有查询检出的文档总数)
  - 宏平均对所有查询一视同仁，微平均受返回相关文档数目比较大的查询影响(宏平均保护弱者，类比：乒乓球参赛资格限制)
- MAP(Mean AP): 对所有查询的AP求宏平均



## 回到例子 采用未插值AP算法

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

查询1及查询2的标准答案集合分别为  $\{d3, d4, d6, d9\}$   $\{d1, d2, d13\}$

系统1查询1:  $P=2/5$ ,  $R=2/4$ ,  $F=4/9$ ,  $AP=1/2$ ; 系统1查询2:  $P=2/5$ ,  $R=2/3$ ,  $F=1/2$ ,  $AP=7/15$ ;

系统2查询1:  $P=2/4$ ,  $R=2/4$ ,  $F=1/2$ ,  $AP=3/8$ ; 系统2查询2:  $P=3/5$ ,  $R=3/3$ ,  $F=3/4$ ,  $AP=11/12$ ;

系统1: MacroP?, MacroR?, MacroF? MAP?, MicroP? MicroR=? MicroF?

系统1的MacroP= $2/5$ , MacroR= $7/12$ , MacroF= $17/36$ , MAP= $29/60$ ,

MicroP= $4/10$ (不去重), MicroR= $4/7$ , MicroF= $8/17$

系统2的MacroP= $11/20$ , MacroR= $3/4$ , MacroF= $5/8$ , MAP= $31/48$ ,

MicroP= $5/9$ , MicroR= $5/7$ , MicroF= $5/8$

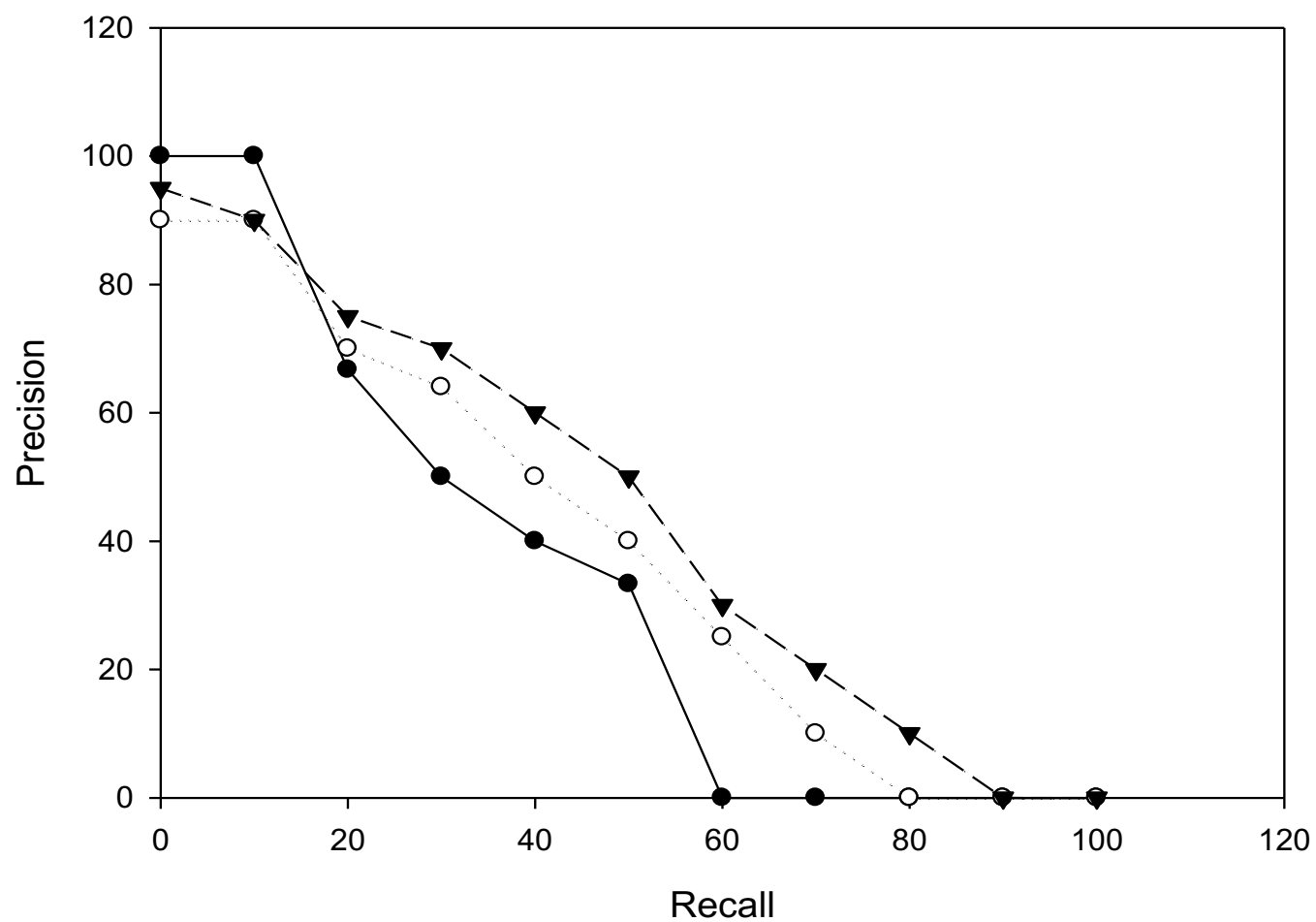
# 整个IR系统的P-R曲线

---

- 在每个召回率点上，对所有的查询在此点上的正确率进行算术平均，得到系统在该点上的正确率的平均值。
- 两个检索系统可以通过P-R曲线进行比较。位置在上面的曲线代表的系统性能占优。

# 几个IR系统的P-R曲线比较

° IIII qf P-R « ' d ± » Ω œ



# 面向用户的评价指标

---

- 前面的指标都没有考虑用户因素。而相关不相关由用户判定。
- 假定用户已知的相关文档集合为 $U$ ，检索结果和 $U$ 的交集为 $R_u$ ，则可以定义覆盖率(Coverage)  $C = |R_u| / |U|$ ，表示系统找到的用户已知的相关文档比例。
- 假定检索结果中返回一些用户以前未知的相关文档 $R_k$ ，则可以定义出新率(Novelty Ratio)  $N = |R_k| / (|R_u| + |R_k|)$ ，表示系统返回的新相关文档的比例。

# 其他评价指标

---

- 不同的信息检索应用或者任务还会采用不同的评价指标
- MRR(Mean Reciprocal Rank) 平均倒排序值：对于某些IR系统(如问答系统或主页发现系统)，只关心第一个标准答案返回的位置(Rank)，越前越好，这个位置的倒数称为RR，对问题集合求平均，则得到MRR
  - 例子：两个问题，系统对第一个问题返回的标准答案的Rank是2，对第二个问题返回的标准答案的Rank是4，则系统的MRR为  $(1/2 + 1/4) / 2 = 3/8$

# 其他评价指标

---

- Adaptive & Batch filtering

	Relevant	Not Relevant
Retrieved	$R^+ / A$	$N^+ / B$
Not Retrieved	$R^- / C$	$N^- / D$

- Utility =  $A * R^+ + B * N^+ + C * R^- + D * N^-$
- $T11U = 2 * R^+ - N^+$
- $P = R^+ / (R^+ + N^+)$ ,  $R = R^+ / (R^+ + R^-)$
- $T11F = 1.25 / (0.25 / R + 1 / P)$
- 归一化平均

# 近几年出现的新的评价指标

---

- Bpref
- GMAP
- NDCG

\*增加于2007年9月20日

# Bpref

---

- Bpref: Binary preference, 2005年首次引入到TREC的 Terabyte任务中
- 基本的思想：在相关性判断(Relevance Judgement) 不完全的情况下，计算在进行了相关性判断的文档集合中，在判断到相关文档前，需要判断的不相关文档的篇数
- 相关性判断完全的情况下，利用Bpref和MAP进行评价的结果很一致，但是相关性判断不完全的情况下，Bpref更鲁棒

\*Buckley, C. & Voorhees, E.M. Retrieval Evaluation with Incomplete Information, Proceedings of SIGIR 2004



# 原始定义

- 对每个Topic，已判定结果中有  $R$  个相关结果

$$bpref = \frac{1}{R} \sum_r \left( 1 - \frac{|n \text{ 排在 } r \text{ 前面}|}{R} \right)$$

- $r$  是相关文档， $n$  是  $r$  前 **不相关** 文档集合的大小

假设检索结果集  $S$  为：

$S = \{D1, D2 \bullet, D3 *, D4 *, D5 \bullet, D6, D7 \bullet, D8, D9, D10\}$

其中  $D2$ 、 $D5$  和  $D7$  是相关文档，

$D3$  和  $D4$  为未经判断的文档。对这个例子来说， $R=3$ ;

$bpref = 1/3 [(1 - 1/3) + (1 - 1/3) + (1 - 2/3)]$ 。

# 特定情况

- 当R很小(1 or 2)时，原公式不合适

$$bpref10 = \frac{1}{R} \sum_r \left( 1 - \frac{|n \text{ 排在 } r \text{ 前面}|}{10+R} \right)$$

- $r$ 是相关文档， $n$ 是Top 10+ $R$ 篇不相关文档集合的子集

# 最新定义

- 对每个Topic，已判定结果集合中有R个相关文档，N个不相关文档，则

$$bpref = \frac{1}{R} \sum_r 1 - \frac{|n \text{排在} r \text{前面}|}{\min(R, N)}$$

Bpref can be thought of as the inverse of the fraction of judged irrelevant documents that are retrieved before relevant ones. Bpref and mean average precision are very highly correlated when used with complete judgments. But when judgments are incomplete, rankings of systems by bpref still correlate highly to the original ranking, whereas rankings of systems by MAP do not.

\*参看trec\_eval工具8.0修正说明(bpref\_bug文件)

# GMAP

- GMAP(Geometric MAP): TREC2004 Robust 任务引进
- 先看一个例子

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

- 从MAP来看，系统A好于系统B，但是从每个查询来看，3个查询中有2个Topic B比A有提高，其中一个提高的幅度达到300%

# GMAP

- 几何平均值

$$GMAP = \sqrt[n]{\prod_{i=1}^n AP_i} = \exp\left(\frac{1}{n} \sum_{i=1}^n \ln AP_i\right)$$

- 上面那个例子  $GMAP_a=0.056$ ,  $GMAP_b=0.086$
- $GMAP_a < GMAP_b$
- GMAP和MAP各有利弊，可以配合使用，**如果存在难Topic时**，GMAP更能体现细微差别

# NDCG Normalized Discounted Cumulative Gain

---

## 归一化折损累计增益

- 每个文档不仅仅只有相关和不相关两种情况，而是有相关度级别，比如0,1,2,3。我们可以假设，对于返回结果：
  - 相关度级别越高的结果越多越好
  - 相关度级别越高的结果越靠前越好

\*Jarvelin, K. & Kekalainen, J. Cumulated Gain-based Evaluation of IR Techniques. *ACM Transactions on Information Systems*, 2002, 20, 422-446

# NDCG

---

- Directed Gain

$$G' = \langle 3, 2, 3, 0, 0, 1, 2, 2, 3, 0, \dots \rangle.$$

- Cumulated Gain(CG) vector

$$CG[i] = \begin{cases} G[1], & \text{if } i = 1 \\ CG[i - 1] + G[i], & \text{otherwise.} \end{cases} \quad (1)$$

$$CG' = \langle 3, 5, 8, 8, 8, 9, 11, 13, 16, 16, \dots \rangle.$$

- Discounted CG vector( $\log_b i$ 表示以 $b$ 为底对 $i$ 取对数)

$$DCG[i] = \begin{cases} CG[i], & \text{if } i < b \\ DCG[i - 1] + G[i]/^b \log i, & \text{if } i \geq b. \end{cases} \quad (2)$$

$$b=2, \quad DCG' = \langle 3, 5, 6.89, 6.89, 6.89, 7.28, 7.99, 8.66, 9.61, 9.61, \dots \rangle.$$

# NDCG

---

- BV(Best Vector) : 假定 $m$ 个3,  $l$ 个2,  $k$ 个1, 其他都是0

$$\text{BV}[i] = \begin{cases} 3, & \text{if } i \leq m, \\ 2, & \text{if } m < i \leq m + l, \\ 1, & \text{if } m + l < i \leq m + l + k, \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

$$I' = \langle 3, 3, 3, 2, 2, 2, 1, 1, 1, 1, 0, 0, 0, \dots \rangle.$$

$$\text{CG}'_I = \langle 3, 6, 9, 11, 13, 15, 16, 17, 18, 19, 19, 19, 19, \dots \rangle$$

$$\text{DCG}'_I = \langle 3, 6, 7.89, 8.89, 9.75, 10.52, 10.88, 11.21, 11.53, 11.83, 11.83, 11.83, \dots \rangle.$$



# NDCG

---

- Normalized (D)CG

$$\textit{norm-vect}(V, I) = \langle v_1/i_1, v_2/i_2, \dots, v_k/i_k \rangle. \quad (5)$$

$$\begin{aligned} \text{nCG}' &= \textit{norm-vect}(\text{CG}', \text{CG}'_I) \\ &= \langle 1, 0.83, 0.89, 0.73, 0.62, 0.6, 0.69, 0.76, 0.89, 0.84, \dots \rangle. \end{aligned}$$

# 例子

- 假设搜索回来的6个结果，其相关性分数分别是 3、2、3、0、1、2
- 那么  $CG = 3+2+3+0+1+2=11$
- 可以看到只是对相关的分数进行了一个关联的打分，并没有召回的所在位置对排序结果评分对影响。而我们看  $DCG = 3+1.26+1.5+0+0.38+0.71 = 6.86$

i	rel <sub>i</sub>	$\log_2(i+1)$	$rel_i / \log_2(i+1)$
1	3	1	3
2	2	1.58	1.26
3	3	2	1.5
4	0	2.32	0
5	1	2.58	0.38
6	2	2.8	0.71

- 归一化需要先结算 IDCG，假如我们实际召回8个物品，除了上面的6个，还有两个结果，假设第7个相关性为3，第8个相关性为0。
- 在理想情况下的相关性分数排序应该是：3、3、3、2、2、1、0、0。计算  $IDCG@6 = 3 + 1.89 + 1.5 + 0.86 + 0.77 + 0.35 = 8.37$
- $NDCG = 6.86 / 8.37 = 81.96\%$

i	rel <sub>i</sub>	$\log_2(i+1)$	$rel_i / \log_2(i+1)$
1	3	1	3
2	3	1.58	1.89
3	3	2	1.5
4	2	2.32	0.86
5	2	2.58	0.77
6	1	2.8	0.35

# NDCG

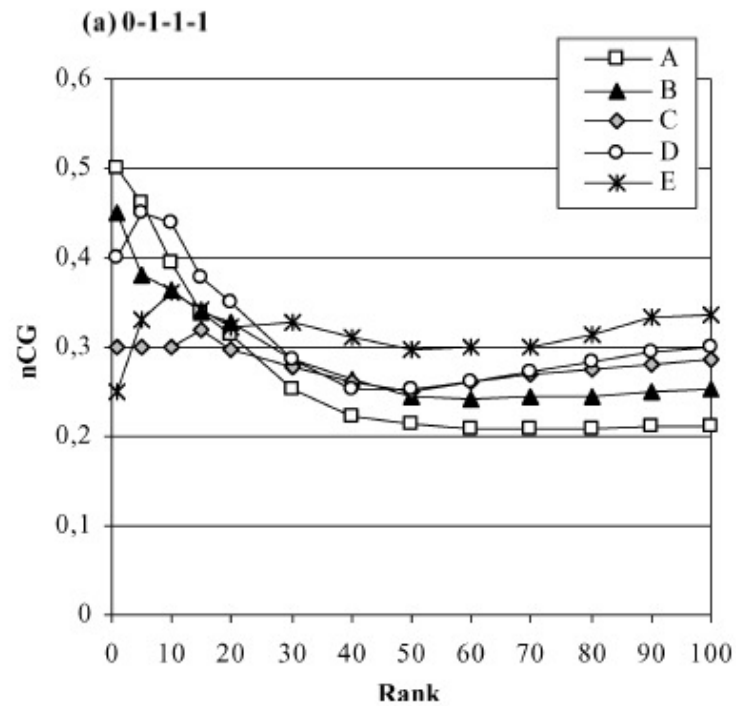


Fig. 3(a). Normalized cumulated gain (nCG) curves, binary weighting.

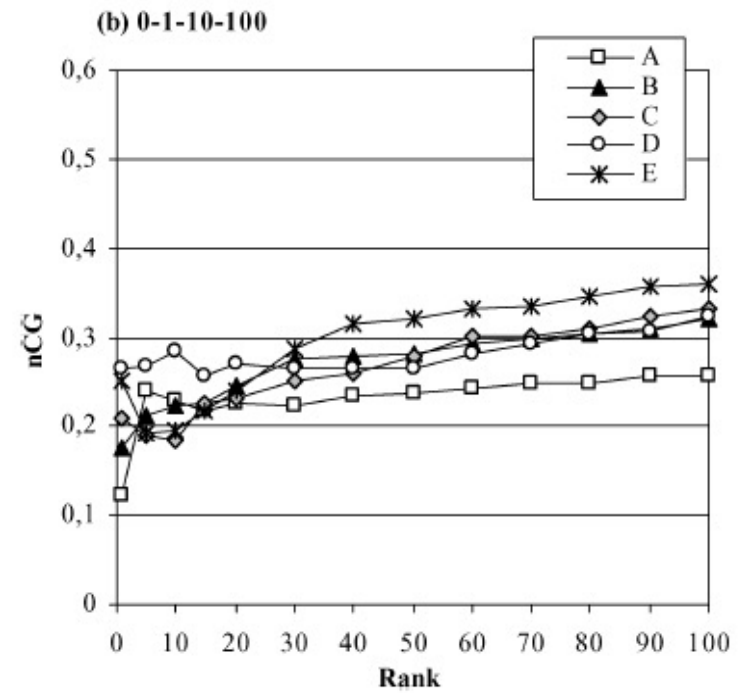


Fig. 3(b). Normalized cumulated gain (nCG) curves, nonbinary weighting.

# NDCG

---

- 优点：
  - 图形直观，易解释
  - 支持非二值的相关度定义，比P-R曲线更精确
  - 能够反映用户的行为特征(如：用户的持续性 persistence)
- 缺点：
  - 相关度的定义难以一致
  - 需要参数设定

\*Ruihua Song, Evaluation in Information Retrieval, 中科院研究生院微软系列讲座,  
<http://tjluo.gucas.ac.cn/sites/wism2006/PPT/Forms/AllItems.aspx>

# 另一种NDCG的计算方法

---

- 加大相关度本身的权重，原来是线性变化，现在是指数变化，相关度3、2、1 在计算时用 $2^3$ 、 $2^2$ 、 $2^1$

$$\text{NDCG}(Q, k) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} Z_{j,k} \sum_{m=1}^k \frac{2^{R(j,m)} - 1}{\log(1 + m)} \quad (8-9)$$

- 据说搜索引擎公司常用这个公式