# 协同过滤

**最大限度的利用用户之间，或者物品之间的相似相关性**

1. 基于用户的推荐：你朋友喜欢，你可能喜欢；K-邻居算法
2. 基于项目的推荐：你喜欢A,A与C相似，你可能也喜欢C
3. 基于模型的推荐：基于**样本**用户的喜好信息**构造推荐模型**，然后根据**实时的**用户喜好信

息**预测推荐；**可见，这种方法使用离线的历史数据，进行模型训练和

评估，需要耗费较长的时间，依赖于实际的数据集规模、机器学习算

法计算复杂度

推荐引擎分类：

1. 协同过滤：基于*相似用户*或者*相似项目*的协同过滤推荐
2. 基于内容分析的推荐：

优点

 1）不需要其它用户的数据，没有冷启动问题和稀疏问题。

2）能为具有特殊兴趣爱好的用户进行推荐。  
 3）能推荐新的或不是很流行的项目，没有新项目问题。  
 4）通过列出推荐项目的内容特征，可以解释为什么推荐那些项目。  
 5）已有比较好的技术，如关于分类学习方面的技术已相当成熟。

缺点

1. 要求特征内容有良好的结构性 2）不能显式地得到其它用户的判断情况

**协同过滤推荐步骤：**

1. 收集用户偏好（用户行为数据）
2. 对用户行为数据进行减噪归一化 **得到一个用户偏好的二维矩阵（用户x物品）**

减噪：误操作产生，可用数据挖掘算法去除 归一化：简单的做法，各类数据除以此类中的最大值

1. 计算相似用户或者相似物品的相似度

找物品间的相似度：用户不变，找多个用户对物品的评分；

找用户间的相似度：物品不变，找用户对某些个物品的评分

**计算相似度**

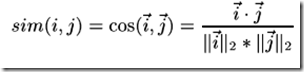
* 1. **欧几里得距离表示相似度：**

当用欧几里德距离表示相似度，距离越小，相似度越大（同时，避免除数为0）

https://ss.csdn.net/p?https://mmbiz.qpic.cn/mmbiz_png/pu7ghYhibpSicgMm9uK2ZLEKU2FtPibPRY02AQib3p5AtI0icib3bQOCu7gCd7HOprQopCsy0Sh9PnV6u5UC7l4PbQuw/640?wx_fmt=png https://ss.csdn.net/p?https://mmbiz.qpic.cn/mmbiz_png/pu7ghYhibpSicgMm9uK2ZLEKU2FtPibPRY0JsibZjial76rzHMsXQNpokp1Wo99LibdYDBXEBDd2I1pQcMSblDIvHCHw/640?wx_fmt=png

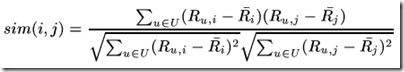
* 1. **余弦相似度：**

两个项目 i ，j 视作为两个m维用户空间向量，相似度计算通过计算两个向量的余弦夹角；对于m\*n的评分矩阵，i ，j的相似度sim( i , j ) 计算公式：

（其中 " · "记做两个向量的内积）

* 1. **皮尔逊相关系数：**

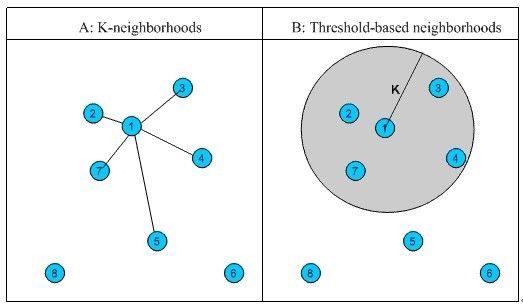
用户集U为既评论了 i 又评论了 j 的用户集



其中Ru,i 为用户u 对项目 i 的评分，对应带横杠的为这个用户集U对项目i的平均评分

1. **计算邻居**

**固定数量的邻居 相似度门槛的邻居**

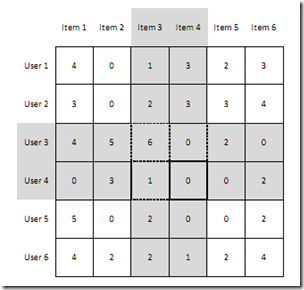


1. 基于用户的CF(基于用户推荐之用：**通过共同口味与偏好找相似邻居用户**，K-邻居算法，你朋友喜欢，你也可能喜欢) ，基于物品的CF(基于项目推荐之用：**发现物品之间的相似度，推荐类似的物品**，你喜欢物品A，C与A相似，那么你可能也喜欢C

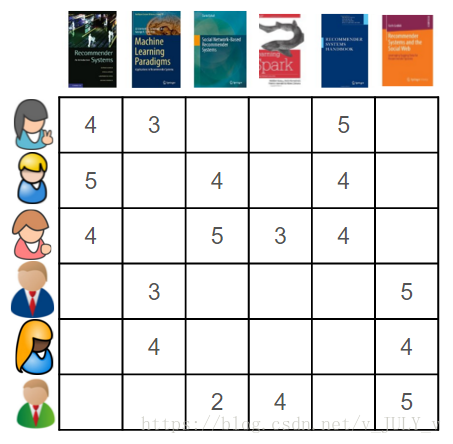
基于用户相似度与项目相似度根本区别

***基于项目相似度计算式计算如Item3，Item4两列向量相似度；***

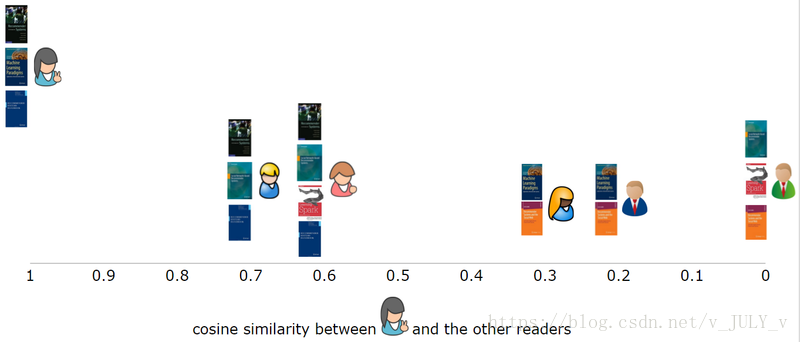
***基于用户相似度计算式计算如User3，User4量行向量相似度。***



**基于用户相似度例子：**

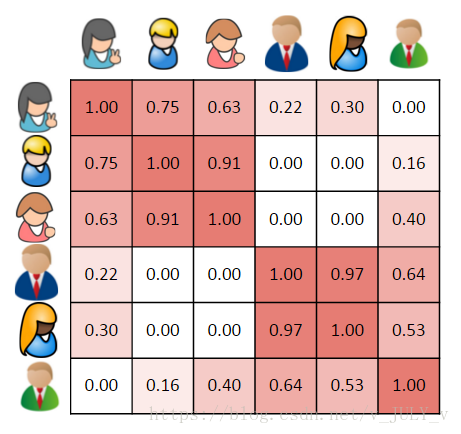


当我们把第一个用户和其他五个用户进行比较时，就能直观的看到他和其他用户的相似程度

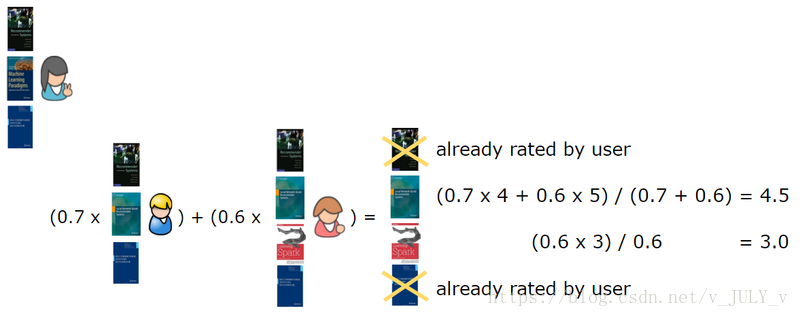


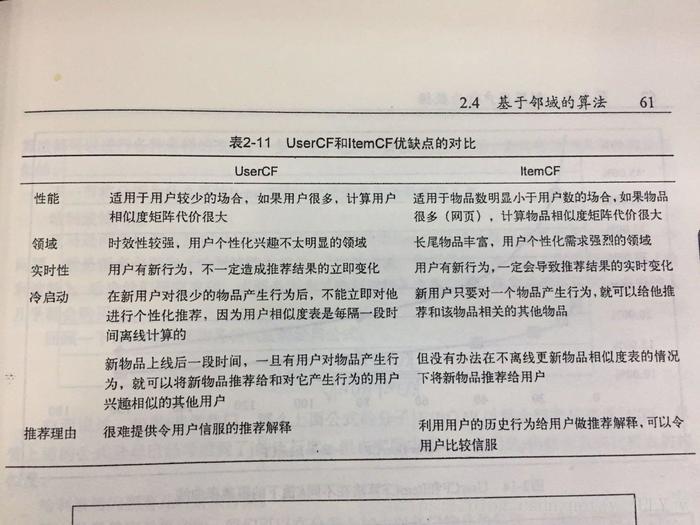
对于大多数相似度量，向量之间相似度越高，代表彼此更相似。本例中，第一个用户第二、第三个用户非常相似，有两本共同书籍，与第四、第五个用户的相似度低一些，只有一本共同书籍，而与最后一名用户完全不相似，因为没有一本共同书籍。

计算出每个用户的相似性：



所以，我们找到了与第一个用户最相似的K个用户，删除第一个用户已经评价过的书籍，给最相似的K个用户正在阅读的书籍加权，然后计算出总和。





基于用户的协同过滤适用于用户较少的情况

# 冷启动问题

1. 非个性化推荐-》数据积累，个性化推荐
2. **用户注册的信息将用户分类**
3. 选择合适的物品启动用户兴趣，利用用户登陆时的物品反馈
4. **利用物品内容**

**UserCF：第一个用户反馈后，与他兴趣相似的推荐列表中会出现**

**第一个用户，先投放给曾喜欢与它类似物品的用户**

**ItemCF：计算新物品与曾出现物品之间的内容相似度**

1. 专家标注
2. **利用用户在其他地方已经沉淀的数据**
3. **利用用户的手机等兴趣偏好**

# 隐语义建模

隐语义模型在Top-N推荐中的应用十分广泛

1. 将物品映射到隐含分类**（不同的类别、主题）**
2. 确定用户对隐含分类的兴趣
3. 选择用户感兴趣的分类中的物品推荐给用户

模型方法：pLSA，LDA，隐含类别模型（latent class model）, 隐含主题模型（latent topic model）, 矩阵分解（matrix factorization）

1. **向用户推荐物品**

通过隐含语义模型将用户（user）和物品（item）自动分类，假设现在有一个大小为m×n的***评分矩阵V***，包含了m个用户对n个物品的评分，评分从0到5，值越大代表越喜欢，0代表没有打分。设定共有***r个隐含的分类***。

**V = W\*H**

W的大小为m×r，H的大小为r×n。在隐语义模型中，***W(i,j)被解释为用户i属于类别j的权重，H(a,b)被解释为物品b属于类别a的的权重。***

**如果用户u对物品i没有评分，可以将这个评分r(u,i)预测为：**

**r(u,i) = sum(W(i, :) .\* H(:, i)) // 向量点乘**

1. **文本分类**

文档-词矩阵：如果共有m个文本，n个单词，那么V的大小为m\*n，V（i,j）表示文档i中出现单词j的次数。设定共有r个隐含的分类。将V展开为两个相乘的矩阵：

**V = W\*H**

W的大小为m×r，H的大小为r×n。在隐语义模型中，***W(i,j)被解释为文档i属于类别j的权重，H(a,b)被解释为单词b属于类别a的的权重。*** 对于一个文档，其权重最大的类别被看作是该文档的类别。由于设定共有r个隐含的分类，分类结果也是r个份分类

pureSVD

SVD：将一个任意实矩阵分解为桑格矩阵**U,S,V**，其中***U , V 是两个正交矩阵***，称为左右奇异矩阵，***S是个对称阵***，称为奇异值矩阵。

Latent Factor Model**（LFM缺点：它很难实现实时推荐）**

推荐系统结果的目标是将那个稀疏的评分矩阵分解成为两个矩阵，***一个表示user的feature，一个表示item的feature***，然后做内积得到预测

目标函数（包含用户平分的偏见+置信权重等） + 约束条件 + 优化（随机梯度下降）

NMF非负矩阵分解

考虑SVD和Latent Factor Model会得到负值的情况，非负矩阵分解；*应用：协同过滤中的预测打分值。*

分解的两个小矩阵：***其中一个是基矩阵W，另一个是投影矩阵H***；即R（nm）=W(nr)H(r\*m)。

W：每一列包含一个基向量，这组基向量构成一个r维的空间。

H： 每一列则近似为原始数据对应的列向量在该r维空间的投影。

NMF步骤:

1 将评分矩阵转置然后分解称为两个矩阵W和H。

2 根据基矩阵W，可以计算得到目标用户评分向量a对基矩阵W的投影向量h.

3 计算投影向量h与投影矩阵H各行之间的欧氏距离，将其中距离最小的前k个用户组成目标用户a的最近邻集合。

4 然后用皮尔逊讲最近邻集合中的数据进行加权计算，然后拍下进行推荐。

# 奇异值分解

将一个比较复杂的矩阵用更小更简单的几个子矩阵的相乘来表示

1. **特征值与奇异值**

**特征值**：

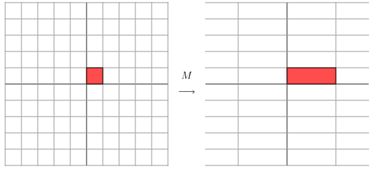
一个向量v是方阵A的特征向量，那么λ就被称为特征向量v对应的特征值

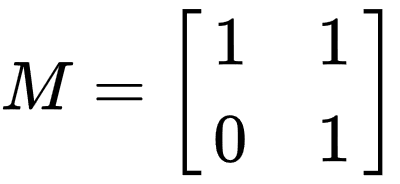
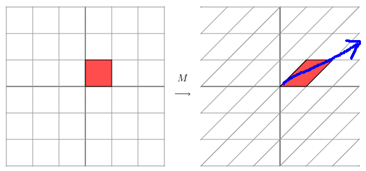
https://julyedu-img.oss-cn-beijing.aliyuncs.com/quesbase64154817358412663868.png

一个矩阵的一组特征向量是一组正交向量。特征值分解是将一个矩阵分解成下面的形式： https://julyedu-img.oss-cn-beijing.aliyuncs.com/quesbase64154817361328500520.png

Q是这个矩阵A的特征向量组成的矩阵，Σ是一个对角阵，每一个对角线上的元素就是一个特征值

一个矩阵其实就是一个线性变换，因为一个矩阵乘以一个向量后得到的向量，其实就相当于将这个向量进行了线性变换

对称的https://julyedu-img.oss-cn-beijing.aliyuncs.com/quesbase64154817370111456813.png

非对称

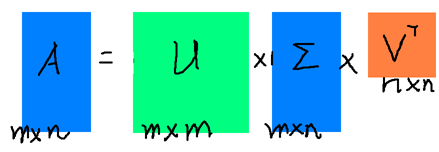
总结：特征值分解可以得到特征值与特征向量，***特征值表示的是这个特征到底有多重要，而特征向量表示这个特征是什么***，可以将每一个特征向量理解为一个线性的子空间。

特征值分解的局限：变换的矩阵必须是方阵。

**奇异值**

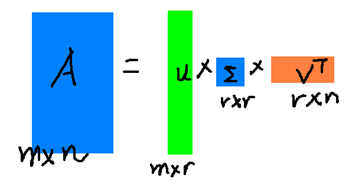
分解任意一个矩阵：

https://julyedu-img.oss-cn-beijing.aliyuncs.com/quesbase64154817385860669940.png



在很多情况下，前10%甚至1%的奇异值的和就占了全部的奇异值之和的99%以上了，我们也可以用前r大的奇异值来近似描述矩阵，r远小于m,n；，*r越接近于n，则相乘的结果越接近于A*

https://julyedu-img.oss-cn-beijing.aliyuncs.com/quesbase6415481739764428773.png



1. **应用：**

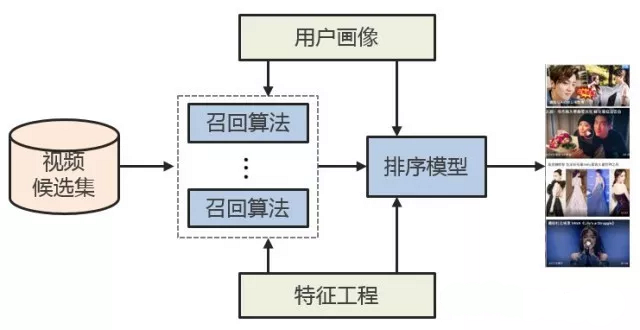
**图像压缩、图像去噪**

# 推荐系统中的排序算法

推荐系统：召回阶段+排序阶段

召回阶段：根据用户的兴趣与行为，从千万级视频库中挑选出一个小的候选集（百千级别）

排序阶段：在此基础上进行更精准的计算，打分，找出少量高质量内容（十级别）



召回算法：包含多通道召回模型；比如协同过滤，主题模型，内容召回等

排序模型：对多个召回通道的内容进行同一个打分排序（统一），选出最优的少量结果

推荐机器学习算法的推进：

1. 前期

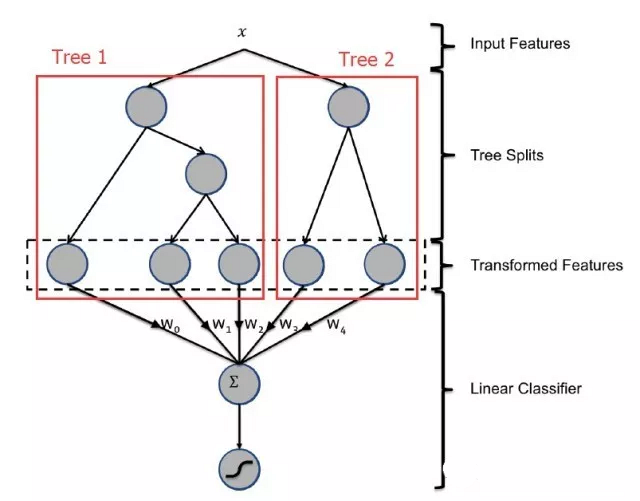
评测指标：ctr不好，偏向短视频，标题党；离线评测指标是按照业务来制定的

优化目标：给样本加权重，使得优化目标与评测目标尽可能的一致

LR：线性分类模型；而实际稠密特征往往非线性，需要特征转换成线性，可以是同两两Interaction

1. 中期

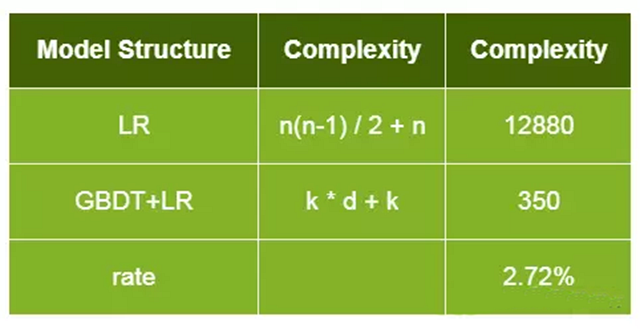
为了解决LR存在的问题，升级模型为GBDT+LR



GBDT基于boosting的ensemble模型，有多颗决策树，优点

* 对输入特征分布没有要求
* 根据熵增益自动特征转换，离散组合
* 预测复杂度与特征的个数无关

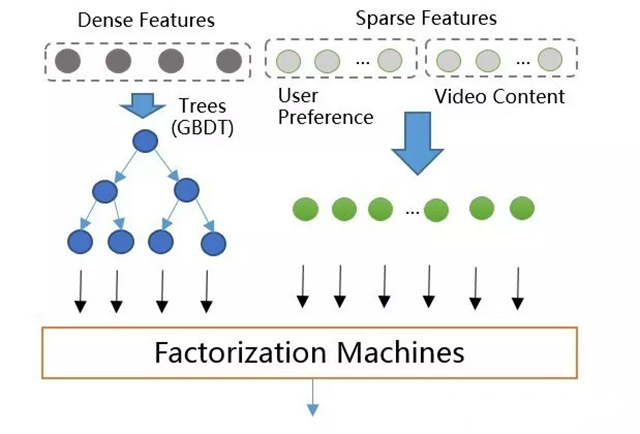
前后期对比：



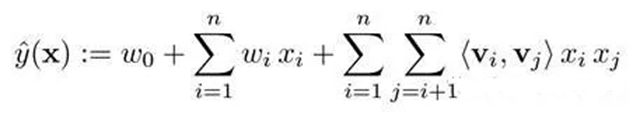
1. 后期

GBDT+LR输入几百维特征（稠密），泛化好，记忆能力差；需要增加高维特征增加记忆能力，GBDT不支持高维稀疏特征。

GBDT+FM



FM模型公式：



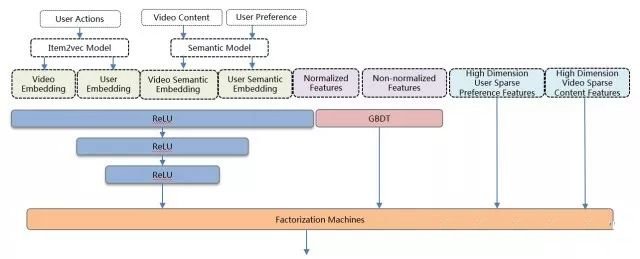
1. 前两项为一个线性模型，相当于LR模型的作用
2. 第三项为一个二次交叉项，能够自动对特征进行交叉组合
3. 增加隐向量，使模型训练和预测的计算复杂度降为了O(N)
4. 支持稀疏特征

GBDT+FM具有良好的***稀疏特征支持***，FM使用***GBDT的叶子结点和稀疏特征（内容特征）作为输入***。GBDT+FM模型上线后相比GBDT+LR在各项指标的效果提升在4%~6%之间。

1. 当代

GBDT+FM模型，对embedding等具有结构信息的深度特征利用不充分，而深度学习（Deep Neural Network）能够对嵌入式（embedding）特征和普通稠密特征进行学习

DNN+GBDT+FM的ensemble模型架构：（相对于（三）后期有4%的提升）



***FM融合层 FM模型与DNN模型作为同一个网络同时训练。 将DNN特征，GBDT输出和稀疏特征进行融合并交叉。***

DNN+GBDT+FM的ensemble模型使用的是Adam优化器。**Adam**结合了The Adaptive Gradient Algorithm（AdaGrad）和Root Mean Square Propagation（RMSProp）算法。**具有更优的收敛速率，每个变量有独自的下降步长，整体下降步长会根据当前梯度进行调节，能够适应带噪音的数据。**

*补充：deeepFM*

*神经网络：提取低阶特征 因子分解机：提取高阶特征*

# 评估指标

## 精度与召回率

精度：查准率； **检索出的相关文档 / 检索出的文档总数**

召回率：查全率； **检索出的相关文档 / 文档中所有相关文档**

F值：反映整体的指标； **2 \* 精度 \* 召回率 /（精度 + 召回率）**

例子：某池塘有1400条鲤鱼，300只虾，300只鳖。现在以捕鲤鱼为目的。撒一大网，逮着了700条鲤鱼，200只虾，100只鳖。那么，这些指标分别如下：

代表查准率的正确率 = 700 / (700 + 200 + 100) = 70%

代表查全率的召回率 = 700 / 1400 = 50%

F值 = 70% \* 50% \* 2 / (70% + 50%) = 58.3%

## 推荐系统中的常用评价标准

常用评价标准：

线上：点击率，网站流量，A/Btest;

线下：

1. 精度

P(k) = c/k

我们给某个用户推荐了k个链接，他／她点击了其中的c个链接，那么精度就是c/k。

1. 平均精度 ap@n

n是被预测的链接的总数，m是用户点击的链接的总数

例子1： 我们一共推荐了10个链接，用户实际上点击了我们推荐当中的第1个和第4个链接，以及另外两个其他的链接，那么对于这个用户， ap@10=(1/1+2/4)/4≈0.38

例子2： 我们一共推荐了10个链接，用户实际上点击了我们推荐当中的第2个,第3个和第5个链接，以及另外三个其他的链接，那么对于这个用户，ap@10=(1/2+2/3+3/5)/6≈0.29

例子3： 我们一共推荐了10个链接，用户实际上点击了我们推荐当中的第2个,第7个，此外没有点击其他联系，那么对于这个用户， ap@10=(1/2+2/7)/2≈0.39

例子4： 我们一共推荐了5个链接，用户实际上点击了我们推荐当中的第1个,第2个和第4个，以及另外6个其他链接，那么对于这个用户， ap@5=(1/1+2/2+3/4)/5≈0.55

1. 平均精度均值 Map@n

，N是用户的数量

比如说我们三个用户甲、乙、丙分别推荐了10个链接，

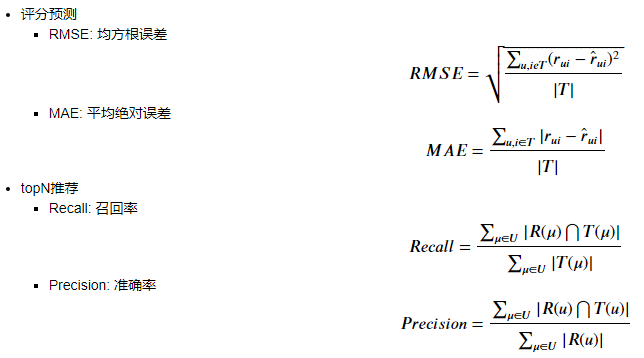
甲点击了我们推荐当中的第1个和第4个链接，以及另外两个其他的链接，那么(ap@10)1=(1/1+2/4)/4≈0.38.

乙点击了我们推荐当中的第3个链接，以及另外一个其他的链接，那么(ap@10)2=(1/3)/2≈0.17.

丙点击了我们推荐当中的第1个链接，第7个链接，以及另外三个其他的链接，那么(ap@10)3=(1/1+2/7)/5≈0.26.

***那么这个模型的平均精度均值 MAP@10 = (0.38+0.17+0.26)/3 ≈ 0.27***

## 评估函数



## 各种指标

多样性：推荐列表两两不相似

新颖性：推荐结果的平均流行度

惊喜性：历史不相似（惊）；满意（喜）

Exploi**ta**tion: 选择现在可能最佳的方案(可能会让推荐的结果范围越来越小)

Explo**ra**tion: 选择现在不确定的一些方案,但未来可能会有高收益的方案(会让推荐的面越来越广)

在做两类决策的过程中,不断更新对所有决策的不确定性的认知,优化长期的目标函数

推荐：准确 + 多样

搜索： 带着query ,无需多样

Ensemble：中每个模型需要单独训练，并且各个**模型之间是相互独立**的，模型之间互相不感知，当预测样本时，每个模型的结果用于投票，最后选择得票最多的结果。

Join train：这种方式**模型之间不是独立**的，是相互影响的，可以同时优化模型的参数。