

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：

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
3. 计算相似用户或者相似物品的相似度

计算相似度

a) 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）

$$d(x, y) = \sqrt{(\sum (x_i - y_i)^2)} \quad \text{sim}(x, y) = \frac{1}{1 + d(x, y)}$$

b) 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 $m \times n$ 的评分矩阵， i ， j 的相似度 $\text{sim}(i, j)$ 计算公式：

$$\text{sim}(i, j) = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{\|\vec{i}\|_2 * \|\vec{j}\|_2} \quad \left(\text{其中 } \vec{i} \cdot \vec{j} \text{ 记做两个向量的内积} \right)$$

c) 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

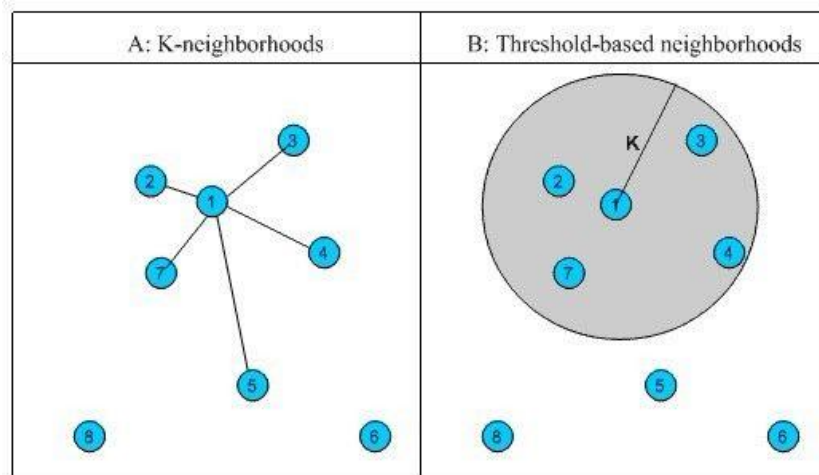
$$sim(i, j) = \frac{\sum_{u \in U} (R_{u,i} - \bar{R}_i)(R_{u,j} - \bar{R}_j)}{\sqrt{\sum_{u \in U} (R_{u,i} - \bar{R}_i)^2} \sqrt{\sum_{u \in U} (R_{u,j} - \bar{R}_j)^2}}$$

其中 $R_{u,i}$ 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

4. 计算邻居

固定数量的邻居

相似度门槛的邻居



5. 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。

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
User 1	4	0	1	3	2	3
User 2	3	0	2	3	3	4
User 3	4	5	6	0	2	0
User 4	0	3	1	0	0	2
User 5	5	0	2	0	0	2
User 6	4	2	2	1	2	4

表2-11 UserCF和ItemCF优缺点的对比

	UserCF	ItemCF
性能	适用于用户较少的场合, 如果用户很多, 计算用户相似度矩阵代价很大	适用于物品数明显小于用户数的场合, 如果物品很多(网页), 计算物品相似度矩阵代价很大
领域	时效性较强, 用户个性化兴趣不太明显的领域	长尾物品丰富, 用户个性化需求强烈的领域
实时性	用户有新行为, 不一定造成推荐结果的立即变化	用户有新行为, 一定会导致推荐结果的实时变化
冷启动	在新用户对很少的物品产生行为后, 不能立即对他进行个性化推荐, 因为用户相似度表是每隔一段时间离线计算的 新物品上线后一段时间, 一旦有用户对物品产生行为, 就可以将新物品推荐给对它产生行为的用户兴趣相似的其他用户	新用户只要对一个物品产生行为, 就可以给他推荐和该物品相关的其他物品 但没有办法在不离线更新物品相似度表的情况下将新物品推荐给用户
推荐理由	很难提供令用户信服的推荐解释	利用用户的历史行为给用户做推荐解释, 可以令用户比较信服

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

5. 专家标注
6. 利用用户在其他地方已经沉淀的数据
7. 利用用户的收集等兴趣偏好

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）

（一）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 $m \times n$ 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。

$$V = W \cdot H$$

W 的大小为 $m \times r$ ， H 的大小为 $r \times n$ 。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：

$$r(u,i) = \text{sum}(W(i, :) \cdot H(:, i)) // \text{向量点乘}$$

（二）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 $m \times n$ ， $V(i,j)$ 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：

$$V = W \cdot H$$

W 的大小为 $m \times r$ ， H 的大小为 $r \times n$ 。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 个分类

pureSVD

SVD：将一个任意实矩阵分解为三个矩阵 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 随机梯度下降

NMF 非负矩阵分解

分解的两个小矩阵：**其中一个基矩阵 W ，另一个是投影矩阵 H** ，即 $R(nm) = W(nr)H(rm)$ 。

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：

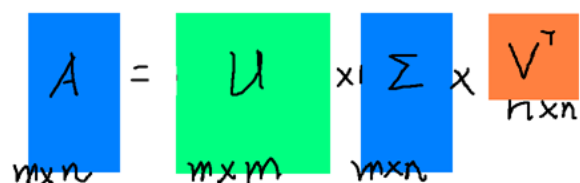
- 1 将评分矩阵转置然后分解称为两个矩阵 W 和 H 。
- 2 根据基矩阵 W ，可以计算得到目标用户评分向量 a 对基矩阵 W 的投影向量 h 。
- 3 计算投影向量 h 与投影矩阵 H 各行之间的欧氏距离，将其中距离最小的前 k 个用户组成目标用户 a 的最近邻集合。
- 4 然后用皮尔逊将最近邻集合中的数据进行加权计算，然后拍下进行推荐。

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

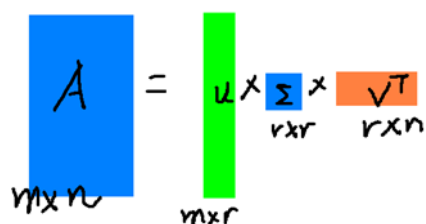
奇异值

分解任意一个矩阵：

$$A = U \Sigma V^T$$


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

$$A_{m \times n} \approx U_{m \times r} \Sigma_{r \times r} V_{r \times n}^T$$


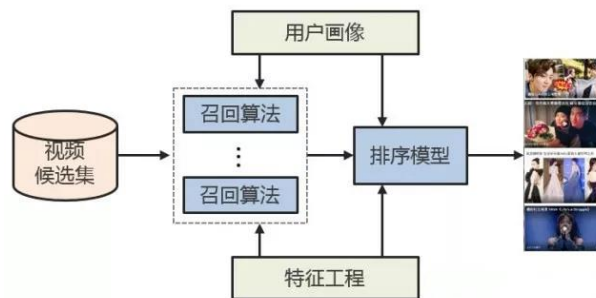
应用：图像压缩、图像去噪

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

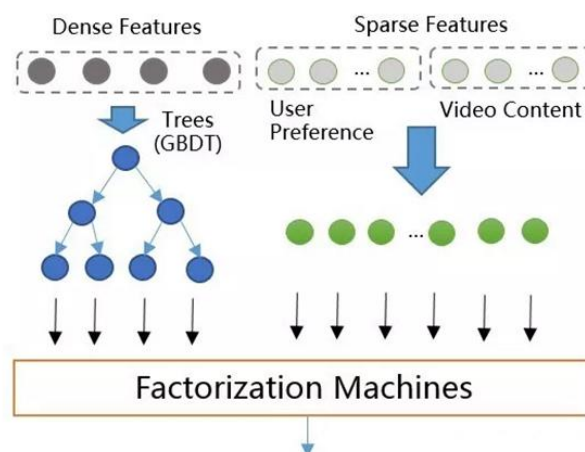
排序模型：对多个召回通道的内容进行同一个打分排序（统一），选出最优的少量结果

推荐机器学习算法的推进：

(一) 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

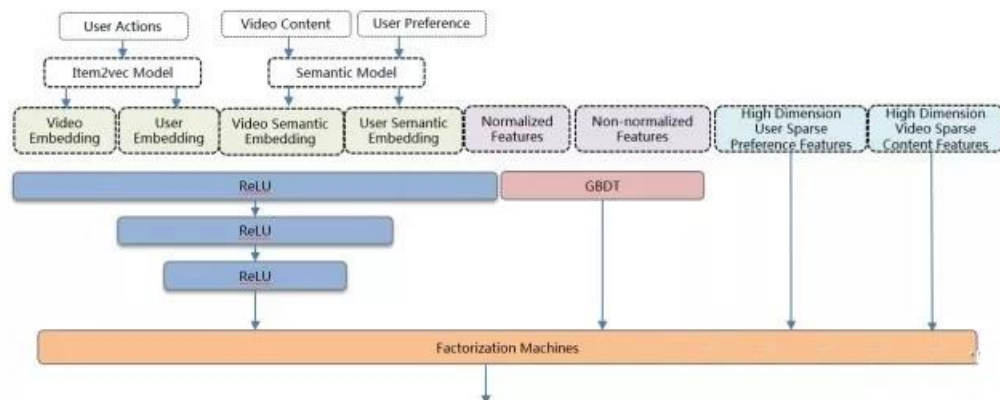


GBDT+FM 具有良好的**稀疏特征支持**，FM 使用 **GBDT 的叶子结点和稀疏特征（内容特征）作为输入**。GBDT+FM 模型上线后相比 GBDT+LR 在各项指标的效果提升在 4%~6%之间。

(二) 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: *deeeFM*

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； $\frac{\text{检索出的相关文档}}{\text{检索出的文档总数}}$
召回率：查全率； $\frac{\text{检索出的相关文档}}{\text{文档中所有相关文档}}$
F 值：反映整体的指标； $2 * \text{精度} * \text{召回率} / (\text{精度} + \text{召回率})$

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。

2. 平均精度 $ap@n$

$$ap@n = \sum_{k=1}^n \frac{P(k)}{\min(m, n)}$$

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， $ap@10 = (1/1 + 2/4)/4 \approx 0.38$

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， $ap@10 = (1/2 + 2/3 + 3/5)/6 \approx 0.29$

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， $ap@10 = (1/2 + 2/7)/2 \approx 0.39$

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， $ap@5 = (1/1 + 2/2 + 3/4)/5 \approx 0.55$

3. 平均精度均值 $Map@n$

$$MAP@n = \sum_{i=1}^N \frac{(ap@n)_i}{N}$$

，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 $(ap@10)_1=(1/1+2/4)/4 \approx 0.38$.

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 $(ap@10)_2=(1/3)/2 \approx 0.17$.

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 $(ap@10)_3=(1/1+2/7)/5 \approx 0.26$.

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

评估函数

- 评分预测

- RMSE: 均方根误差

$$RMSE = \sqrt{\frac{\sum_{u,i \in T} (r_{ui} - \hat{r}_{ui})^2}{|T|}}$$

- MAE: 平均绝对误差

$$MAE = \frac{\sum_{u,i \in T} |r_{ui} - \hat{r}_{ui}|}{|T|}$$

- topN推荐

- Recall: 召回率

$$Recall = \frac{\sum_{\mu \in U} |R(\mu) \cap T(\mu)|}{\sum_{\mu \in U} |T(\mu)|}$$

- Precision: 准确率

$$Precision = \frac{\sum_{\mu \in U} |R(u) \cap T(u)|}{\sum_{\mu \in U} |R(u)|}$$

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)

Exploitation: 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)

Exploration: 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。