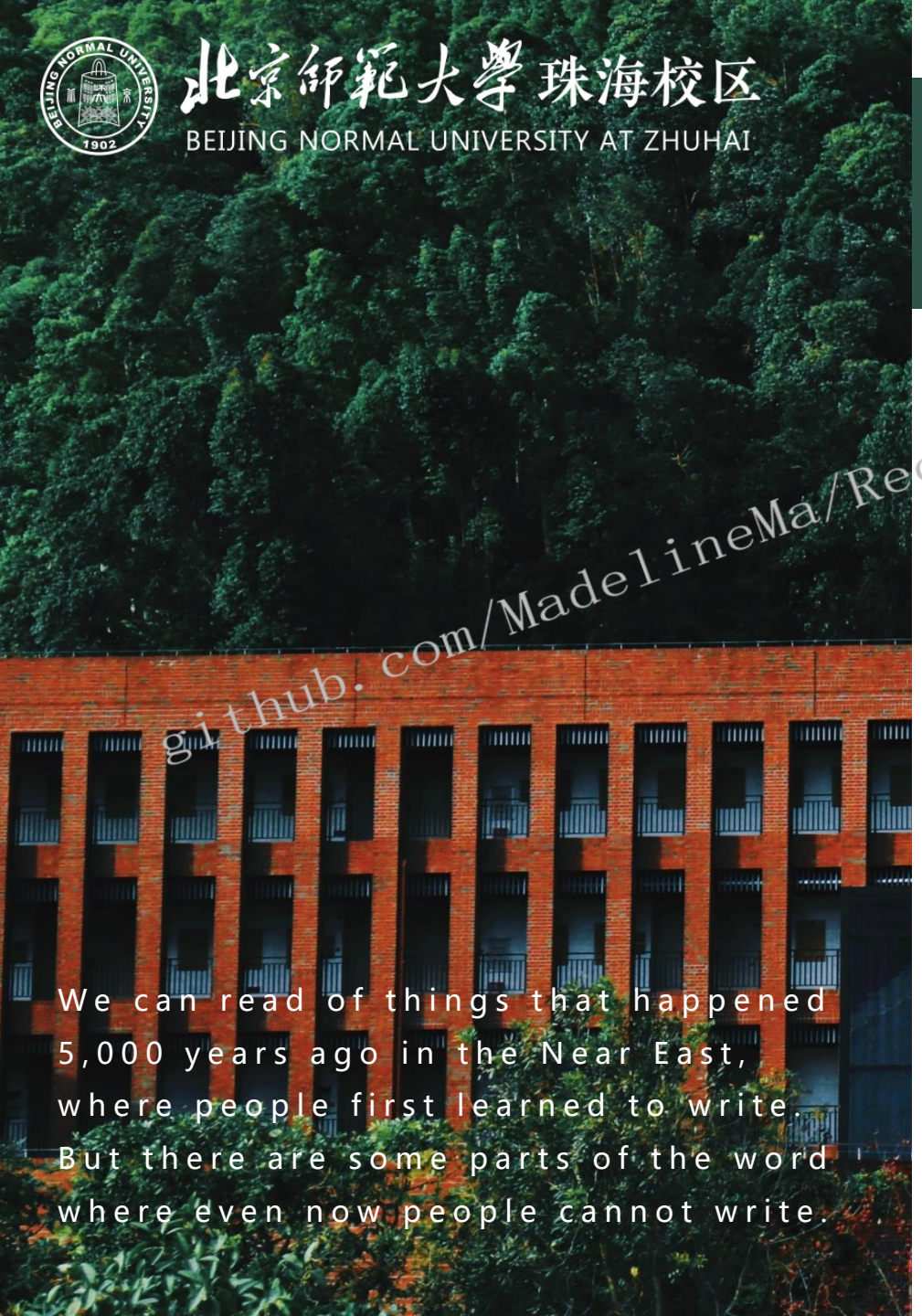

北京師範大學珠海校区
BEIJING NORMAL UNIVERSITY AT ZHUHAI
前深度学习时代
github.com/MadelineMa/Recommender-System
github.com/MadelineMa/Recommender-System
马静

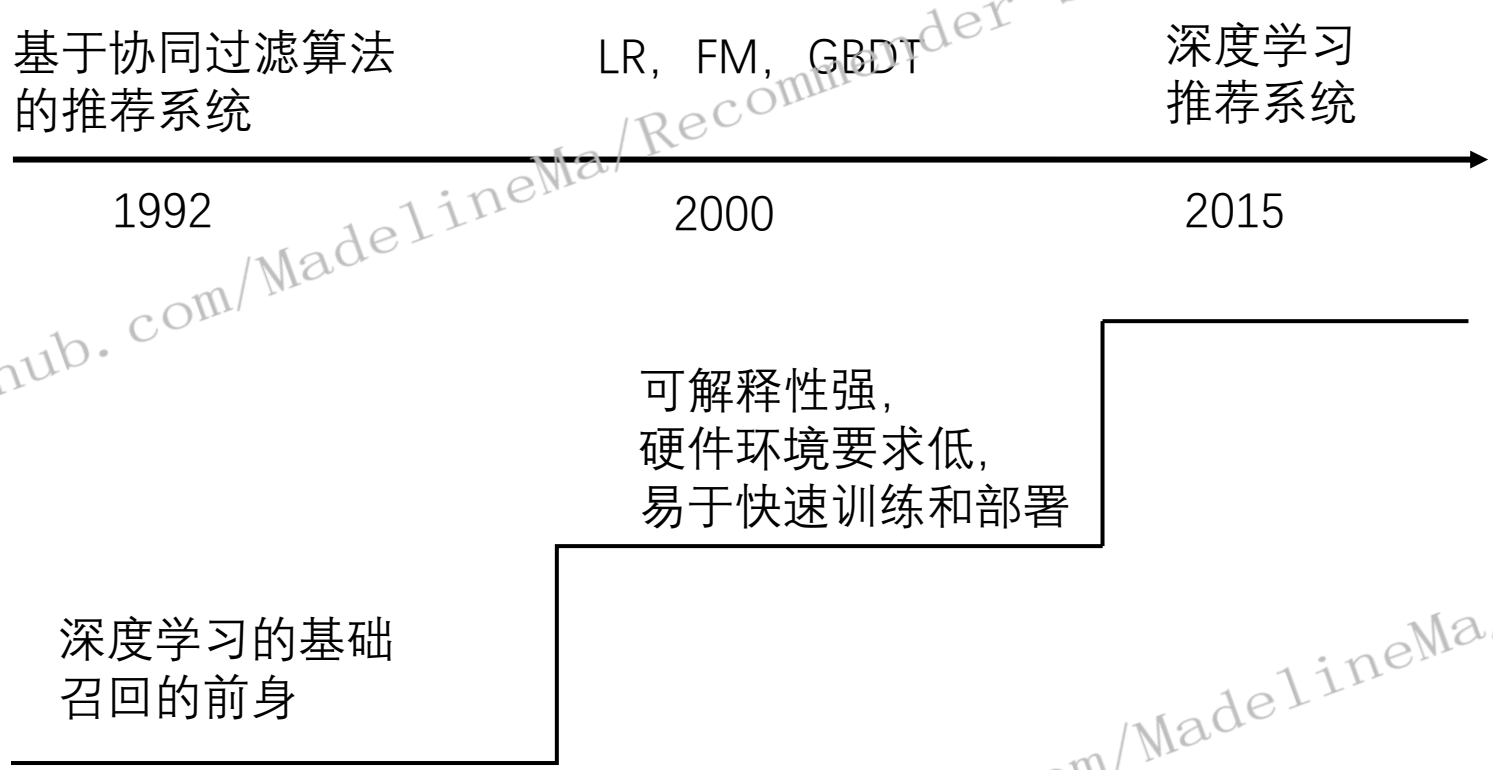


CONTENT

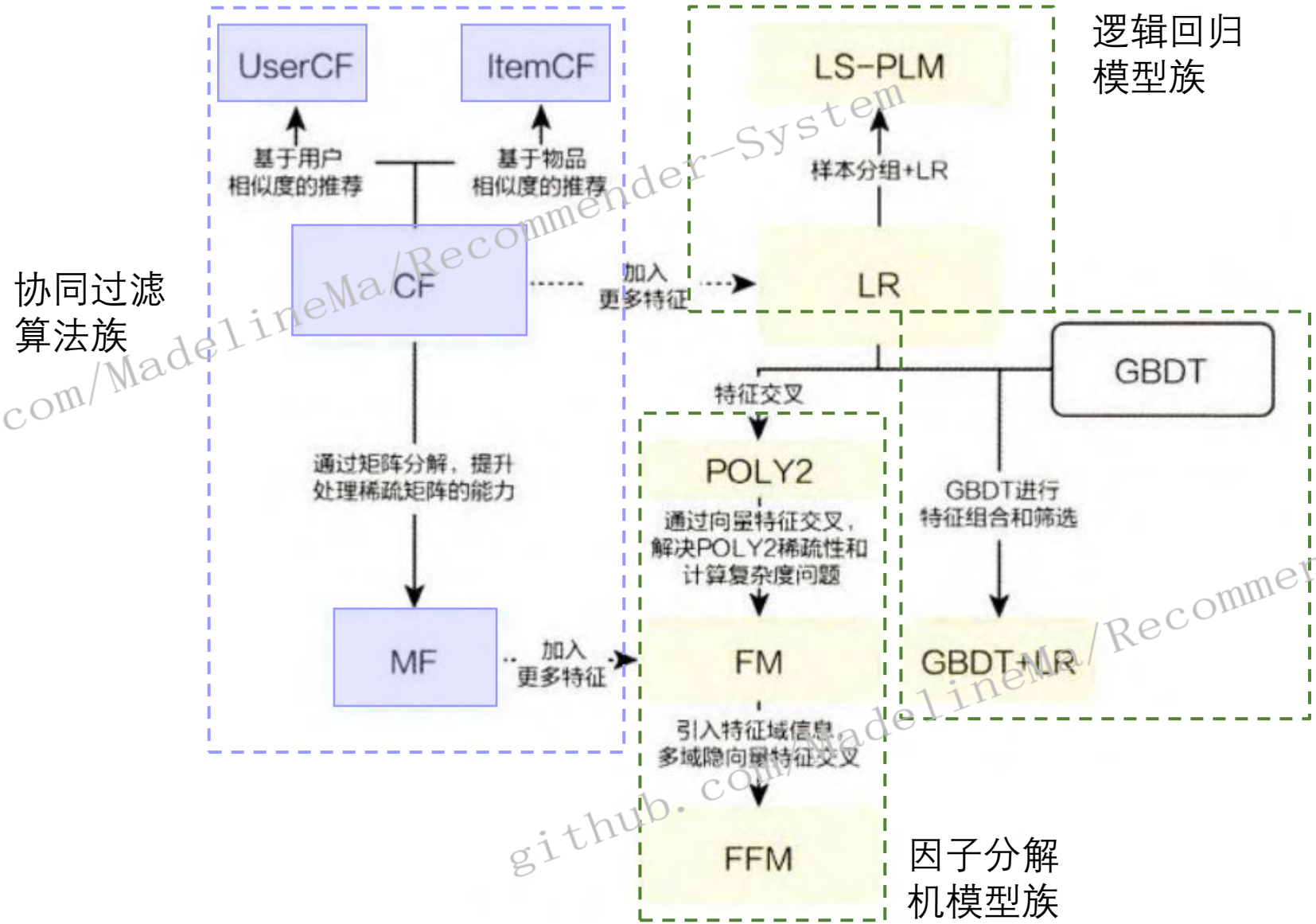
北京師範大學珠海校区
BEIJING NORMAL UNIVERSITY AT ZHUHAI

01传统推荐模型的演化关系

传统推荐模型的演化关系



传统推荐模型的演化关系

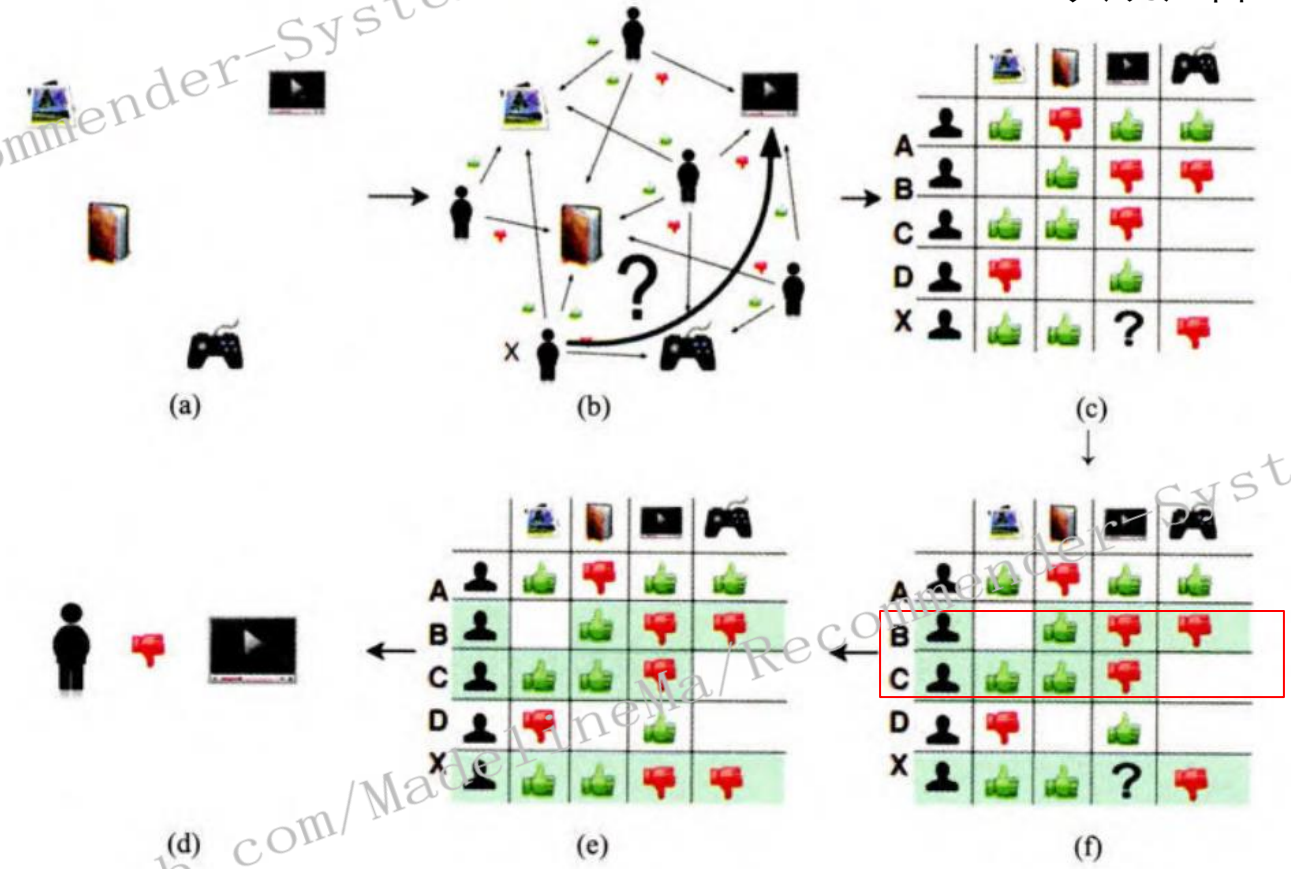


02
协同过滤-经典的推荐算法

协同过滤

- 1992年，应用于邮件筛选系统，过滤不感兴趣的无用邮件；
- 2003年，Amazon应用于推荐系统；
- 定义：协同大家的反馈、评价和意见一起对海量的信息进行过滤，从中筛选出目标用户可能感兴趣的信息的推荐过程；
- 用户相似度计算与最终结果的排序不严谨.

共现矩阵



用户相似度计算->向量之间相似度计算

余弦相似度

$$\mathrm{sim}(i,j) = \cos(i,j) = \frac{i \cdot j}{\|i\| \cdot \|j\|}$$

用户打分纠偏的皮尔逊相似度

$$\mathrm{sim}(i,j) = \frac{\sum_{p \epsilon P}(R_{i,p} - \bar{R}_i)(R_{j,p} - \bar{R}_j)}{\sqrt{\sum_{p \epsilon P}(R_{i,p} - \bar{R}_i)^2}\sqrt{\sum_{p \epsilon P}(R_{j,p} - \bar{R}_j)^2}}$$

物品打分纠偏的皮尔逊相似度

$$\mathrm{sim}(i,j) = \frac{\sum_{p \epsilon P}(R_{i,p} - \overline{R_p})(R_{j,p} - \overline{R_p})}{\sqrt{\sum_{p \epsilon P}(R_{i,p} - \overline{R_p})^2}\sqrt{\sum_{p \epsilon P}(R_{j,p} - \overline{R_p})^2}}$$

最终结果排序

假设：目标用户与其相似用户的喜好是相似的

- 目标货品池打分：

$$R_{u,p} = \frac{\sum_{s \epsilon S}(w_{u,s} \cdot R_{s,p})}{\sum_{s \epsilon S} w_{u,s}}$$

- 对打分结果进行排序，选取高分货品进行推荐.

阿里解决方式：离线计算Top200相似用户喜欢的相似货品，存为在线表格，以便实时快速查询

userId	SimiUserId	itemId	info
1110	2314	11348	…

x UserCF 需要维护用户相似度矩阵表以便快速找出TopN相似用户，存储以及存储的增长对在线存储系统提出挑战；

x 历史数据稀疏，对冷启动用户以及反馈难获取的场景并不适用（酒店预订，大件商品低频购买应用）

货品相似度计算->共现矩阵的列向量相似度

1. 基于历史数据，构建以用户（假设用户总数为m)为行坐标，物品（物品总数为n)为列坐标的mxn维的共现矩阵。
2. 计算共现矩阵两两列向量间的相似性（相似度的计算方式与用户相似度的计算方式相同），构建维的物品相似度矩阵。
3. 获得用户历史行为数据中的正反馈物品列表。
4. 利用物品相似度矩阵，针对目标用户历史行为中的正反馈物品，找出相似的TopA个物品，组成相似物品集合。
5. 对相似物品集合中的物品，利用相似度分值进行排序，生成最终的推荐列表。

UserCF 与 ItemCF应用差异：

- 面向的对象：快速得知与自己兴趣相似的人最近喜欢什么？ UserCF：新闻，微博，热点发现.
  一段时间内更倾向于某类货品. ItemCF：电商，视频推荐.

CF的缺点：

- 泛化能力差，热门商品以及反馈多的商品容易更热门，新品和小众的商品快速沉底
  （长尾问题以及处理稀疏向量能力弱）

$$\begin{matrix} A[0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1] \\ B[0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0] \\ C[0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0] \\ D[1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1] \end{matrix} \Rightarrow \begin{matrix} & A & B & C & D \\ A & - & 0.00 & 0.00 & 0.71 \\ B & 0.00 & - & 0.00 & 0.18 \\ C & 0.00 & 0.00 & - & 0.18 \\ D & 0.71 & 0.18 & 0.18 & - \end{matrix}$$

- 仅利用用户和物品的交互信息，无法有效引入用户基本信息，商品基本信息，上下文特征.

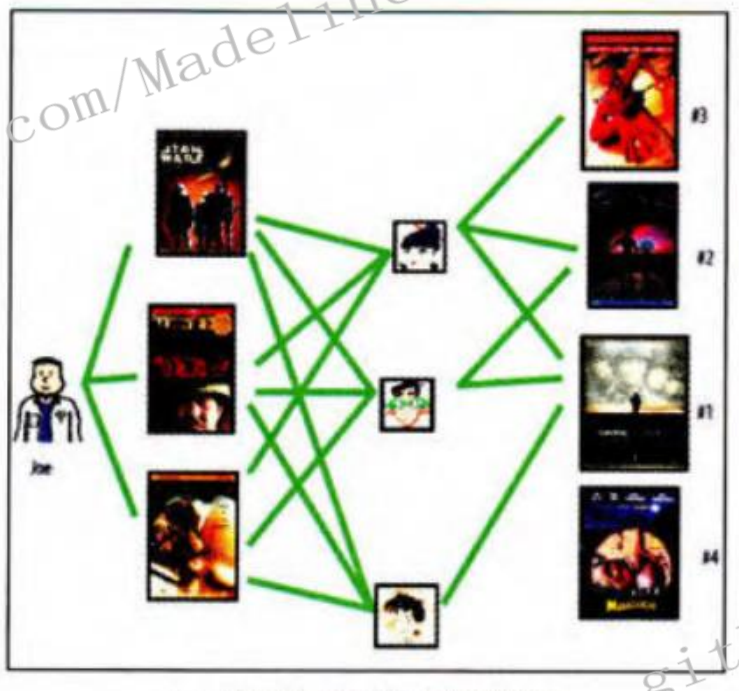
北京師範大學珠海校区
BEIJING NORMAL UNIVERSITY AT ZHUHAI

03 矩阵分解法

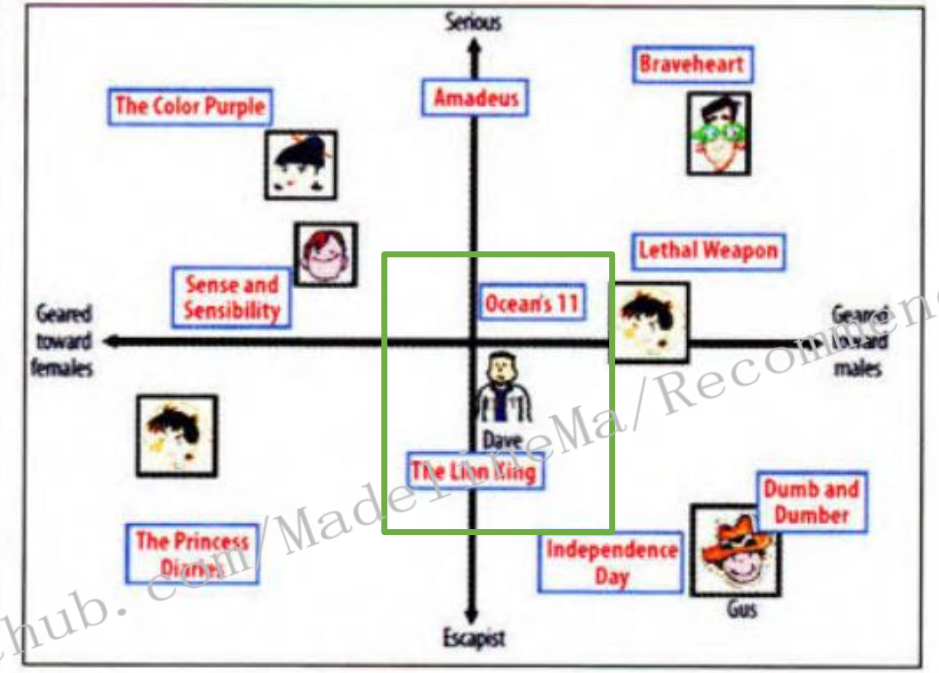
We can read of things that happened 5,000 years ago in the Near East, where people first learned to write.

矩阵分解

- CF的缺点：泛化能力差，长尾问题以及处理稀疏向量能力弱；
- 2006年，Netflix Prize Challenge中，MF算法大放异彩；
- 隐向量的引入；



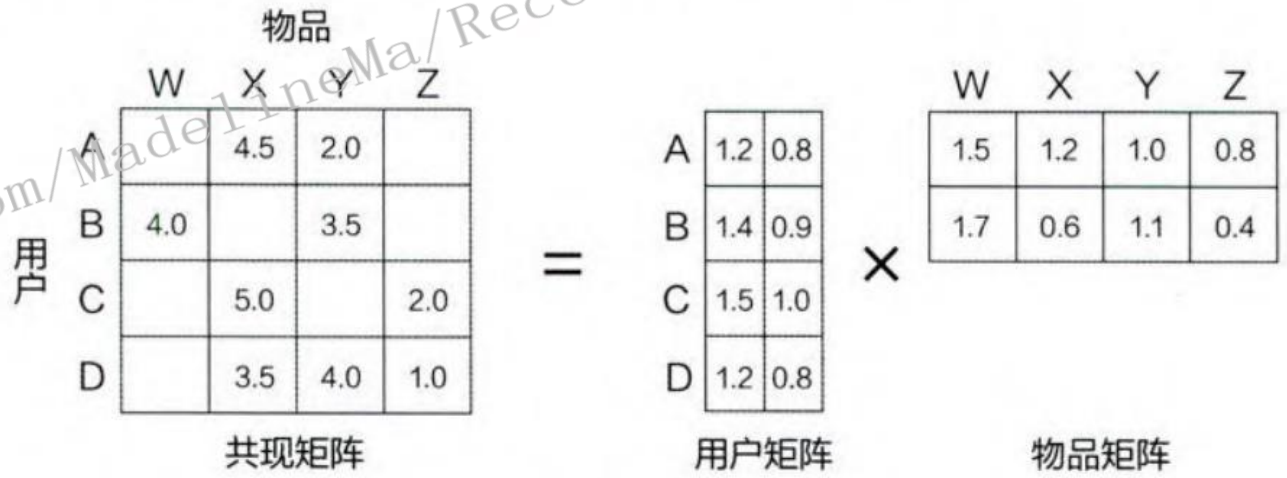
(a) 协同过滤算法原理图



(b) 矩阵分解算法原理图

矩阵分解概览

- 目标：用隐向量表达用户和物品，还要保证相似的用户及用户可能喜欢的物品的距离相近.
- **用户和物品的隐向量是通过分解协同过滤生成的共现矩阵得到的.**



- 将mxn维的共现矩阵及分解为mxk维的用户矩阵U, kxn维的物品矩阵V相乘的形式.
- k的大小决定了隐向量表达能力的强弱和泛化能力强弱?
- 最终打分 $\hat{r}_{ui} = q_i^T p_u$

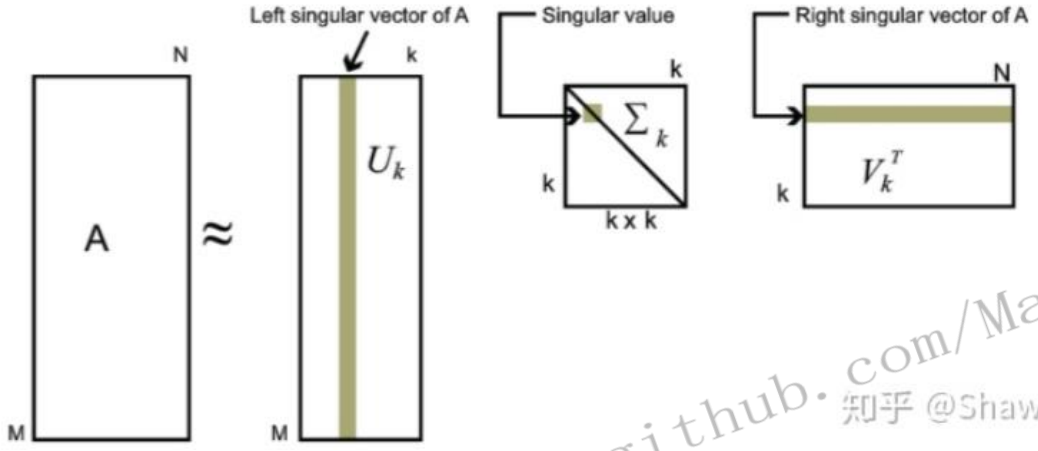
矩阵分解过程

x 特征值分解：只能用于方阵，不适用于互联网场景，why?

x 奇异值分解：要求原始的共现矩阵是稠密的，复杂度为$O(mn^2)$，不适用，Why?

假设矩阵M是一个mxn的矩阵，则一定存在一个分解$M = U\sum V^T$,其中U是mxm的正交矩阵，V是nxn的正交矩阵，$\sum$是mxn的对角阵.

取对角阵$\sum$中较大的k个元素作为隐含特征，删除其他维度及U和V中对应的维度，矩阵M被分解为$M \approx U_{m\times k}\sum V^T_{k\times n}$



★

U_k：用户矩阵的第k列被对角阵的第k行k列的元素进行整行放缩；

U_m：取用户矩阵第m行隐向量，每个元素j逐个被对角阵的j行j列元素放缩，再与货品矩阵的列相乘。

矩阵分解过程

✓梯度下降法：**让原始评分与用户向量和物品向量之积的差尽量小**

$$\min_{q^*,p^*} \sum_{(u,i)\in K} (r_{ui} - q_i^T p_u)^2$$

$$\min_{q^*,p^*} \sum_{(u,i)\in K} (r_{ui} - q_i^T p_u)^2 + \lambda(\|q_i\| + \|p_u\|)^2$$ 正则化的overfitting, Why?

✓面试常见题：什么是overfitting? 哪些降低overfitting的方法？L1和L2优缺点是什么？两种方法在收敛的角度看差异在哪里？L2好在哪里？

1）对用户求偏导，得到的结果为 $2(r_{ui} - q_i^T p_u)p_u - 2\lambda q_i$

2）对货品求偏导，得到的结果为 $2(r_{ui} - q_i^T p_u)q_i - 2\lambda p_u$

3）更新梯度： $p_u \leftarrow p_u - \gamma\left((r_{ui} - q_i^T p_u)q_i - \lambda p_u\right)$ $q_i \leftarrow q_i - \gamma\left((r_{ui} - q_i^T p_u)p_u - \lambda q_i\right)$

4）终止条件: n次或损失的变化几乎不变.

矩阵分解过程

- ✓梯度下降法：**让原始评分与用户向量和物品向量之积的差尽量小**

- ✓将正则化项加入损失函数来保持模型稳定的做法也可以做如下理解。对于加入了正则化项的损失函数来说，模型权重的值越大，损失函数越大。梯度下降是朝着损失（Loss)小的方向发展的，因此正则化项其实是希望在尽量不影响原模型与数据集之间损失的前提下，使模型的权重变小，权重的减小自然会让模型的输出波动更小，从而达到让模型更稳定的目的。

MF解决了CF的缺点?

- CF的缺点：泛化能力差，长尾问题以及处理稀疏向量能力弱；
- 2006年，Netflix Prize Challenge中，MF算法大放异彩；
- 隐向量的引入，使任意的用户和物品之间都可以得到预测分值；
- 而隐向量的生成过程其实是对共现矩阵进行全局拟合的过程，因此隐向量其实是利用全局信息生成的，有更强的泛化能力；
- 而对协同过滤来说，如果两个用户没有相同的历史行为，两个物品没有相同的人购买，那么这两个用户和两个物品的相似度都将为0. Why?

✓泛化能力强.
✓空间复杂度低，存储低.
✓Embedding思想的前身
x 不方便加入用户、商品、上下文的基本信息进行综合考虑.

本周作业

1. 正则化的资料阅读与深入思考.
2. CF和MF的资料阅读和深入思考.
3. Embedding课题小组可趁热打铁组织报告或论文.

Reference：

L1正则化与L2正则化 - 知乎 (zhihu.com)

l1 相比于 l2 为什么容易获得稀疏解？ - 知乎 (zhihu.com)

最小推荐系统：Neural Collaborative Filtering (NeuralCF) - 知乎 (zhihu.com)

万物皆Embedding，从经典的word2vec到深度学习基本操作item2vec - 知乎 (zhihu.com)

一个值得讨论的问题:word2vec与SVD/LSA等的关系 - 知乎 (zhihu.com)

[NLP] 秒懂词向量Word2vec的本质 - 知乎 (zhihu.com)


北京師範大學珠海校区
BEIJING NORMAL UNIVERSITY AT ZHUHAI
github.com/MadelineMa/Recommender-System
THANKS
github.com/MadelineMa/Recommender-System
DESIGNED BY 2xh