

第1章: Recommender System Strategies

两大策略: content filtering、collaborative filtering

一、content filtering 内容过滤

基于内容推荐的原理是根据用户感兴趣的物品A, 找到和A内容信息相近的物品B。内容信息是指用户和物品本身的内容特征, 如用户的地理位置、性别、年龄, 电影物品的导演、演员、发布时间等。比如用户喜欢看《神探夏洛克第一季》, 那么就给他推荐《神探夏洛克第二季》。

基于内容推荐的优点如下:

通过 profile 获得

- 简单、有效, 推荐结果直观, 容易理解, 不需要领域知识;
- 不需要用户的历史数据, 如对对象的评价等;
- 没有物品冷启动的问题;
- 没有稀疏问题;
- 算法成熟, 如数据挖掘、聚类分析等。

基于内容的推荐的缺点如下:

- 特征提取能力有限
比如图像、视频, 没有有效的特征提取方法。即便是文本资源, 特征提取也只能反应一部分内容, 难以提取内容质量, 会影响用户满意度。
- 很难出现新的推荐结果
根据用户兴趣的喜好进行推荐, 很难出现惊喜。对于时间敏感的内容, 如新闻, 推荐内容基本相同, 体验度较差。
- 存在用户冷启动的问题
当新用户出现时, 系统较难获得该用户的兴趣偏好, 无法进行有效推荐。
- 推荐对象内容分类方法需要的数据量较大

想要为每个用户、物品都创建一个“profile”

二、Collaborative filtering 协同过滤

仅依赖过去的行为, 无需创建“profile”, 但有冷启动问题

1. neighborhood methods 基于邻域的模型

(1) 基于用户的协同过滤算法 UserCF

在一个在线个性化推荐系统中, 当一个用户A需要个性化推荐时, 可以先找到和A有相似兴趣的其他用户, 然后把那些用户喜欢的、而用户A没有关注过的物品推荐给A。这种方法称为基于用户的协同过滤算法 (UserCF)。

基于用户的协同过滤算法主要包括两个步骤:

- 找到和目标用户兴趣相似的用户集合
- 找到这个集合中的用户喜欢的, 且目标用户没有听说过的物品, 推荐给目标用户

算法的关键是计算两个用户的兴趣相似度。协同过滤算法主要利用用户兴趣列表的相似度计算用户兴趣的相似度, 给定用户 u 、 v , 令 $N(u)$ 表示用户 u 曾经有过兴趣的物品集合, $N(v)$ 表示用户 v 曾经有过兴趣的物品集合。

(2) 基于物品的协同过滤算法 ItemCF

基于物品的协同过滤算法用于给用户推荐那些与他们之前喜欢的物品相似的物品。

ItemCF算法主要通过分析用户的行为记录来计算物品之间的相似度。该算法认为, 物品A和物品B具有很大的相似度是因为喜欢物品A的用户大都喜欢物品B。

基于物品的协同过滤算法主要分为两步:

- 计算物品之间的相似度;
- 根据物品的相似度和用户的历史行为给用户生成推荐列表。

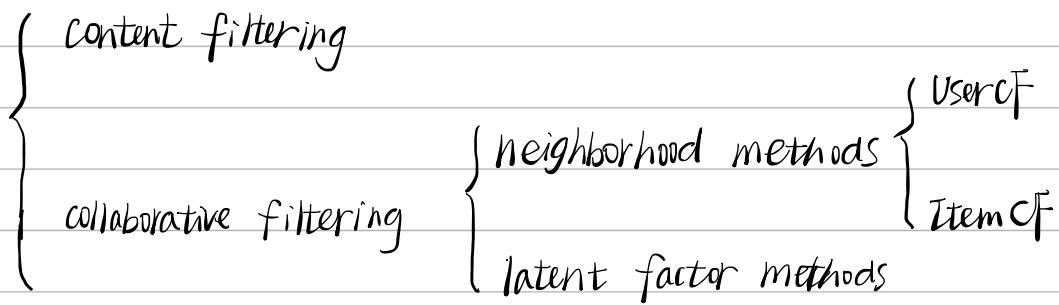
我们给定物品 i 、 j , 设 $N(i)$ 为喜欢物品 i 的用户数, $N(j)$ 为喜欢物品 j 的用户数, 则物品 i 、 j 的相似度可以表达为:

2. Latent factor method 隐语义模型

LFM (Latent Factor Model) 隐语义模型的核心思想是通过隐含特征(Latent Factor)联系用户兴趣和物品，它采取基于用户行为统计的自动聚类，让用户和物品的分类自动化。

推断很多行因子来表征用户和物品，比如电影是否为喜剧、是否对儿童等，还可能有一些无法解释的维度

总结



Content-filtering 介绍

基于内容的推荐算法

虽然协同过滤是目前较为流行的推荐算法，在学术界和工业界都有广泛的研究和使用，但同样作为推荐系统领域的基础算法——基于内容的推荐也很重要，其他它还是最早出现的推荐算法。其基本原理是根据用户之前对物品的历史行为（如用户购买过什么物品、对什么物品收藏过、评分过等等，然后再根据计算与这些物品相似的物品，并把它们推荐给用户。例如用户之前购买过金庸的武侠小说，这可以说明用户可能是一个金庸迷或武侠迷，这时就可以给用户推荐一些金庸的其他武侠小说。基于内容的推荐算法之前也成为基于内容的过滤（搜索）算法，早期主要应用在信息检索和信息过

基于内容的推荐算法一般包括以下三步：

- 1、为每个物品抽取一些特征用来表示这个物品。→ 物品特征
- 2、使用用户的历历史行为数据分析物品的这些特征，从而学习出用户的喜好特征或者兴趣。→ 用户特征
- 3、通过比较上一步得到的用户兴趣和待推荐物品的特征，确定一组相关性最大的物品作为推荐列表。→ 对比用户、物品特征

MF 模型是协同过滤中的常见技术，将 user 和 item 插入一个共享的低维空间中

MF 基本算法

一、总体目标：用不完整矩阵中的已有评价来预测未知评价

将 item 和 user 都映射到 \mathbb{R}^d 向量空间

item 向量 $q_i \in \mathbb{R}^d$, user 向量 $p_u \in \mathbb{R}^d$, $\tilde{r}_{ui} = q_i^T p_u$

难点在于如何得到各 q_i 和 p_u

二、优化思路

输入的矩阵存在稀疏问题，不采用以往的“填补缺数据”法，而是利用已有数据

在已知评价集合上最小化平方误差，正规化避免过拟合

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

三、法一：SGD 随机梯度下降

记误差 $e_{ui} = r_{ui} - q_i^T p_u$, 则转化为 $\min_{q^*, p^*} \sum_{(u, i) \in R} e_{ui}^2 + \lambda (\|q_i\|^2 + \|p_u\|^2)$

对 R 中的每一个 (u, i) , 更新参数

$$q_i = q_i - \gamma [z e_{ui} \cdot (-p_u) + 2\lambda p_u] = q_i + \gamma (e_{ui} \cdot p_u - \lambda p_u)$$

$$p_u = p_u - \gamma [z e_{ui} \cdot (-q_i) + 2\lambda q_i] = p_u + \gamma (e_{ui} \cdot q_i - \lambda q_i)$$

此法速度很快，但某些场景下不如 ALS

四、法二：ALS 算法最小二乘

因为在原评级矩阵中，大量未知元是我们想推断的，所以这个重构误差是包含未知数的。而 ALS 算法的解决方案很简单：只计算已知打分的重构误差。

即固定 user 和 item 中的一者，更新另一者

ALS 的实现原理是迭代式求解一系列最小二乘回归问题。在每一次迭代时，固定用户因子矩阵或是物品因子矩阵中的一个，然后用固定的这个矩阵以及评级数据来更新另一个矩阵。之后，被更新的矩阵被固定住，再更新另外一个矩阵。如此迭代，直到模型收敛（或是迭代了预设好的次数）。

ALS 算法推导

ALS 对于显式矩阵分解的损失函数：

$$\min_{U, I} \sum_{i, j} (r_{i, j} - U_i^T I_j)^2 + \lambda (\|U_i\|^2 + \|I_j\|^2)$$

其中， r 是打分（原始）矩阵， $r(i, j)$ 表示用户 i 对物品 j 的实际打分。

$U_{i, j}$ 是根据用户和商品的隐藏因子矩阵算得的值，

所以：某一个物品评分的误差 = (实际打分 - 计算值) $\wedge 2$

λ 是正则化的参数。正规化是为了防止过拟合的情况发生。

在算法执行中，比如先随机化 I (物品隐藏因子矩阵)，并固定之，然后对 U (用户隐藏因子矩阵) 在损失函数 $L(U, I)$ 上求偏导，因为 ALS 算法本质是
最小二乘法，所以令其导数 = 0

最小二乘法、求偏导 = 0

$$\frac{\partial L(U, I)}{\partial U} = -2 \sum I (r - U^T I) + 2U \lambda$$

令其 = 0，得：

$$U = (I^T I + \lambda E)^{-1} I r$$

根据对称性，当固定 U ，求 I 时：

$$I = (U U^T + \lambda E)^{-1} U r$$

两种方法对比

{ -一般情况下，SGD有 fast 的优点
但有两点场景下，ALS更优：并行化、矩阵非常疏
解释：针对 R 中的每一对 (u, i), SGD 都要更新计算
而 ALS 一次计算可以更新所有的 U 或 I

五. 回归完善

1. 引入偏差

整体平均评分 μ , item 偏差 b_i , user 偏差 b_u

$$b_{ui} = \mu + b_i + b_u$$

$$\tilde{r}_{ui} = b_{ui} + q_i^T p_u = \mu + b_i + b_u + q_i^T p_u$$

$$\min_{p^*, q^*, b^*} \sum_{(u, i) \in E} [r_{ui} - (\mu + b_i + b_u + q_i^T p_u)]^2 + \lambda (||p_u||^2 + ||q_i||^2 + b_u^2 + b_i^2)$$

2. 针对冷启动，引入外部信息

隐式反馈 $|N(u)|^{-0.5} \sum_{i \in N(u)} x_i^{4.5}$

用户属性 $\sum_{a \in A(u)} y_a$

$$\tilde{r}_{ui} = \mu + b_i + b_u + q_i^T [p_u + |N(u)|^{-0.5} \sum_{i \in N(u)} x_i^{4.5} + \sum_{a \in A(u)} y_a]$$

3. 站点初表本质

b_i 动态 $\rightarrow b_i(t)$, b_u 动态 $\rightarrow b_u(t)$, p_u 动态 $\rightarrow p_u(t)$, q_i 静态

$$\tilde{r}_{ui}(t) = \mu + b_i(t) + b_u(t) + q_i^T p_u(t)$$

4. 引入置信度 c_{ui}

$$\min_{p^*, q^*, b^*} \sum_{(u, i) \in E} c_{ui} [r_{ui} - (\mu + b_i + b_u + q_i^T p_u)]^2 + \lambda (||p_u||^2 + ||q_i||^2 + b_u^2 + b_i^2)$$

六. MF求解方法总结

① SVD, 但要先处理缺失值

② SGD > 只利用已知数据, 不处理缺失值

③ ALS