推荐系统实践

一、好的推荐系统

1.1 什么是推荐系统

推荐系统就是自动联系用户和物品的一种工具，它能够在信息过载的环境中帮助用户发现令他们感兴趣的信息，也能将信息推送给对它们感兴趣的用户。

1.2 个性化推荐系统的应用

1. 电子商务（如亚马逊）
2. 电影和视频网站（如Netflix、YouTube）
3. 个性化音乐网络电台（如Pandora）
4. 社交网络（如Facebook和Twitter）
5. 个性化阅读（Google Reader）
6. 基于位置的服务（Foursquare）
7. 个性化邮件（Gmail）
8. 个性化广告

1.3 推荐系统评测

推荐系统实验方法

- 离线实验
    1. 通过日志系统获得用户行为数据，并按照一定格式生成一个标准的数据集；
    2. 将数据集按照一定的规则分成训练集和测试集；
    3. 在训练集上训练用户兴趣模型，在测试集上进行预测；
    4. 通过事先定义的离线指标评测算法在测试集上的预测结果。

表1-2　离线实验的优缺点

优　　点	缺　　点
不需要有对实际系统的控制权	无法计算商业上关心的指标
不需要用户参与实验	离线实验的指标和商业指标存在差距
速度快，可以测试大量算法	

- 用户调查
    - 需要有一些真实用户，让他们在需要测试的推荐系统上完成一些任务。在他们完成任务时，我们需要观察和记录他们的行为，并让他们回答一些问题。最后，我们需要通过分析他们的行为和答案了解测试系统的性能。
    - 它的优点是可以获得很多体现用户主观感受的指标，相对在线实验风险很低，出现错误后很容易弥补。缺点是招募测试用户代价较大，很难组织大规模的测试用户，因此会使测试结果的统计意义不足。此外，在很多时候设计双盲实验非常困难，而且用户在测试环境下的行为和真实环境下的行为可能有所不同，因而在测试环境下收集的测试指标可能在真实环境下无法重现。
- 在线实验
    - AB测试是一种很常用的在线评测算法的实验方法。它通过一定的规则将用户随机分成几组，并对不同组的用户采用不同的算法，然后通过统计不同组用户的各种不同的评测指标比较不同算法，比如可以统计不同组用户的点击率，通过点击率比较不同算法的性能。
    - AB测试的优点是可以公平获得不同算法实际在线时的性能指标，包括商业上关注的指标。

AB测试的缺点主要是周期比较长，必须进行长期的实验才能得到可靠的结果。

评测指标

1. 用户满意度：用户满意度没有办法离线计算，只能通过用户调查或者在线实验获得。
2. 预测准确度：度量一个推荐系统或者推荐算法预测用户行为的能力。这个是最重要的推荐系统离线评测指标
    1. 均方根误差（RMSE）：$$\mathrm{RMSE}=\frac{\sqrt{\sum_{u,i\in T}(r_{ui}-\hat{r}_{ui})^2}}{|T|}$$
    2. 平均绝对误差（MAE）：$$\mathrm{MAE}=\frac{\sum_{u,i\in T}|r_{ui}-\hat{r}_{ui}|}{|T|}$$
    3. TopN推荐：准确率（precision）/召回率（recall）度量
    4. 覆盖率：推荐系统能够推荐出来的物品占总物品集合的比例
    5. 多样性：多样性描述了推荐列表中物品两两之间的不相似性。
    6. 新颖性：新颖的推荐是指给用户推荐那些他们以前没有听说过的物品
    7. 惊喜度：如果推荐结果和用户的历史兴趣不相似，但却让用户觉得满意，就可以说推荐的惊喜度很高，
    8. 信任度：对推荐系统的信任程度
    9. 实时性：新闻、微博等具有很强的时效性
    10. 健壮性：robust,鲁棒性指标衡量了一个推荐系统抗击作弊的能力。
    11. 商业目标：不同的网站具有不同的商业目标

测评维度

- 用户维度 主要包括用户的人口统计学信息、活跃度以及是不是新用户等。
- 物品维度 包括物品的属性信息、流行度、平均分以及是不是新加入的物品等。
- 时间维度 包括季节，是工作日还是周末，是白天还是晚上等。

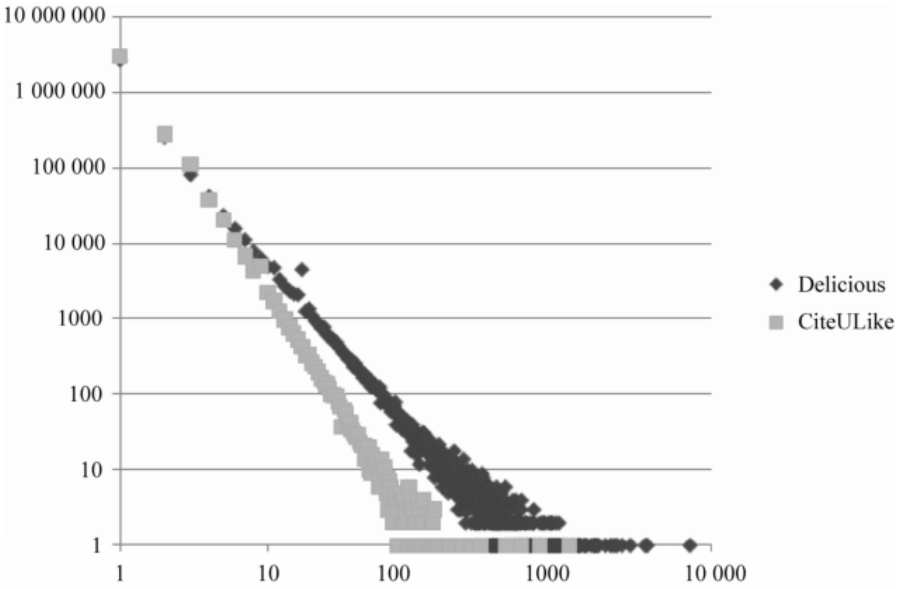
二、利用用户行为数据

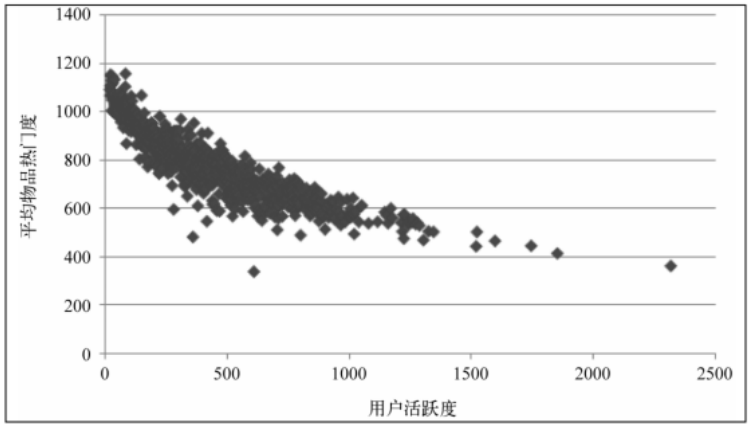
基于用户行为分析的推荐算法是个性化推荐系统的重要算法，学术界一般将这种类型的算法称为协同过滤算法。顾名思义，协同过滤就是指用户可以齐心协力，通过不断地和网站互动，使自己的推荐列表能够不断过滤掉自己不感兴趣的物品，从而越来越满足自己的需求。

2.1 用户行为数据简介

- 无上下文信息的隐性反馈数据集 每一条行为记录仅仅包含用户ID和物品ID。
- 无上下文信息的显性反馈数据集 每一条记录包含用户ID、物品ID和用户对物品的评分。
- 有上下文信息的隐性反馈数据集 每一条记录包含用户ID、物品ID和用户对物品产生行为的时间戳。
- 有上下文信息的显性反馈数据集 每一条记录包含用户ID、物品ID、用户对物品的评分和评分行为发生的时间戳。

2.2 用户行为分析

- 用户活跃度和物品流行度的分布（符合长尾分布）
  - 令*f***u*(*k*)为对*k*个物品产生过行为的用户数，令*f***i*(*k*)为被*k*个用户产生过行为的物品数
  - $$f_i(k) = \alpha_i k^{\beta_i}$$
  - $$f_u(k) = \alpha_u k^{\beta_u}$$
  - 

  - 横坐标是物品的流行度*K*/用户的活跃度*K*，纵坐标是流行度为*K*的物品的总数/活跃度为*K*的用户总数
- 用户活跃度和物品流行度的关系
  - 用户越活跃，越倾向于浏览冷门的物品
  - 

  - 基于邻域的方法
    - 基于用户的协同过滤算法 这种算法给用户推荐和他兴趣相似的其他用户喜欢的物品。
    - 基于物品的协同过滤算法 这种算法给用户推荐和他之前喜欢的物品相似的物品。

	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
	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和对它产生行为的用户兴趣相似的其他用户	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

- 隐语义模型：通过隐含特征(latent factor)联系用户兴趣和物品

三、推荐系统冷启动问题

3.1 冷启动简介

- 用户冷启动：用户冷启动主要解决如何给新用户做个性化推荐的问题
- 物品冷启动：物品冷启动主要解决如何将新的物品推荐给可能对它感兴趣的用户这一问题
- 系统冷启动：系统冷启动主要解决如何在一个新开发的网站上（还没有用户，也没有用户行为，只有一些物品的信息）设计个性化推荐系统

解决方法：

- 提供非个性化的推荐：非个性化推荐的最简单例子就是热门排行榜，
- 利用用户注册时提供的年龄、性别等数据做粗粒度的个性化。
- 利用用户的社交网络账号登录（需要用户授权），导入用户在社交网站上的好友信息，然后给用户推荐其好友喜欢的物品。
- 要求用户在登录时对一些物品进行反馈，收集用户对这些物品的兴趣信息，然后给用户推荐那些和这些物品相似的物品
- 对于新加入的物品，可以利用内容信息，将它们推荐给喜欢过和它们相似的物品的用户。
- 在系统冷启动时，可以引入专家的知识，通过一定的高效方式迅速建立起物品的相关度表

3.2 利用用户注册信息

- 人口统计学信息
- 用户兴趣的描述
- 从其他网站导入的用户站外行为数据

3.3 选择合适的物品启动用户的兴趣

新用户第一次访问推荐系统时，不立即给用户展示推荐结果，而是给用户提供一些物品，让用户反馈他们对这些物品的兴趣，然后根据用户反馈给提供个性化推荐。

一般来说，能够用来启动用户兴趣的物品需要具有以下特点

- 比较热门
- 具有代表性和区分性
- 启动物品集合需要有多样性

3.4 利用物品的内容信息

如何将新加入的物品推荐给对它感兴趣的用户。物品冷启动在新闻网站等时效性很强的网站中非常重要

- UserCF算法对物品冷启动问题并不非常敏感，需要解决第一推动力的问题
- 对于ItemCF算法来说，物品冷启动就是一个严重的问题了，频繁地更新相关表

四、利用用户标签数据

让普通用户给物品打标签，也就是UGC（User Generated Content，用户生成的内容）的标签应用。UGC的标签系统是一种表示用户兴趣和物品语义的重要方式

4.1 UGC 标签系统的代表应用

- Delicious
- 论文书签网站CiteULike
- 音乐网站Last.fm
- 视频网站Hulu
- 书和电影评论网站豆瓣

4.2 标签系统中的推荐问题

- 如何利用用户打标签的行为为其推荐物品（基于标签的推荐）？
- 如何在用户给物品打标签时为其推荐适合该物品的标签（标签推荐）？

解决：

1. 用户为什么要打标签?

   首先是社会维度，有些用户标注是给内容上传者使用的（便于上传者组织自己的信息），而有些用户标注是给广大用户使用的（便于帮助其他用户找到信息）。另一个维度是功能维度，有些标注用于更好地组织内容，方便用户将来的查找，而另一些标注用于传达某种信息，比如照片的拍摄时间和地点等。

2. 用户如何打标签?
3. 用户打什么样的标签

五、利用上下文信息

5.1 时间上下文信息

时间效应简介

- 用户兴趣是变化的
- 物品也是有生命周期的
- 季节效应

5.2 地点上下文信息

- 兴趣本地化
- 活动本地化

六、利用社交网络数据

6.1 获取社交网络数据的途径

- 电子邮件
- 用户注册信息
- 用户的位置数据
- 论坛和讨论组
- 即时聊天工具
- 社交网站

6.2 社交网络数据简介

- 双向确认的社交网络数据
- 单向关注的社交网络数据
- 基于社区的社交网络数据

6.3 基于社交网络的推荐

优点：

- 好友推荐可以增加推荐的信任度
- 社交网络可以解决冷启动问题

算法：

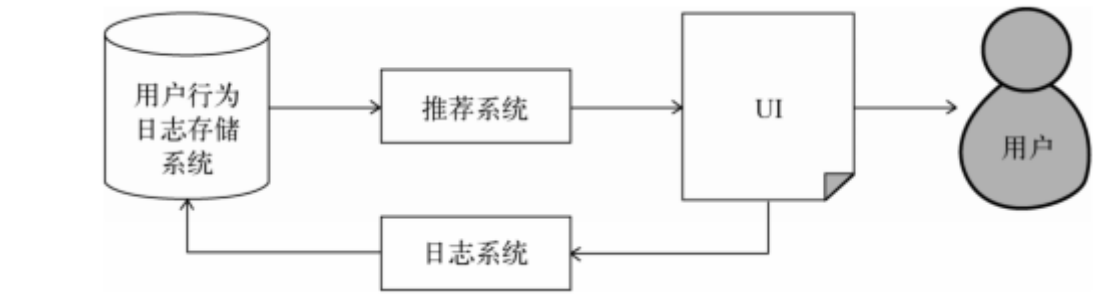
- 基于邻域的社会化推荐算法
- 基于图的社会化推荐算法
- 实际系统中的社会化推荐算法
- 社会化推荐系统和协同过滤推荐系统
- 信息流推荐

6.4 给用户推荐好友

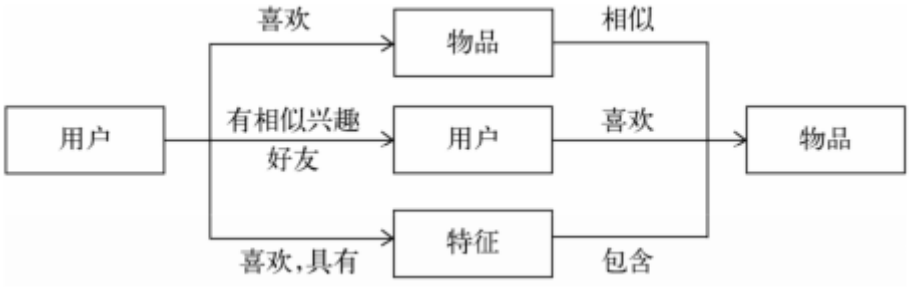
- 基于内容的匹配
- 基于共同兴趣的好友推荐
- 基于社交网络图的好友推荐
- 基于用户调查的好友推荐算法对比

七、推荐系统实例

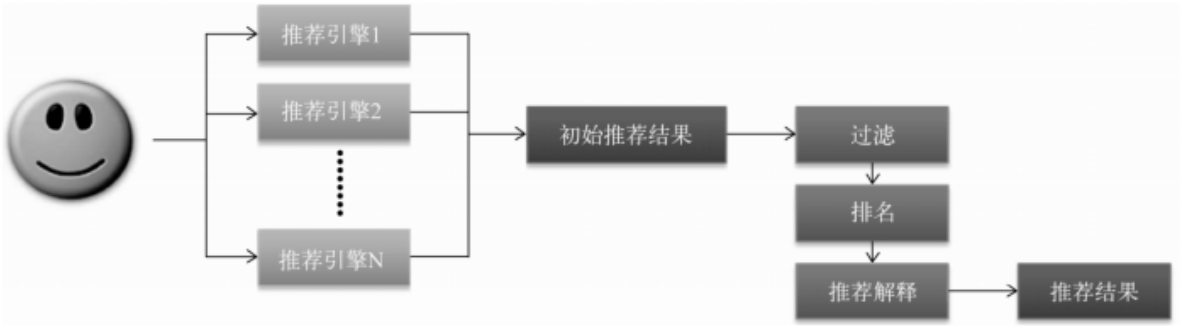
推荐系统和其他系统之间的关系



3种联系用户和物品的推荐系统

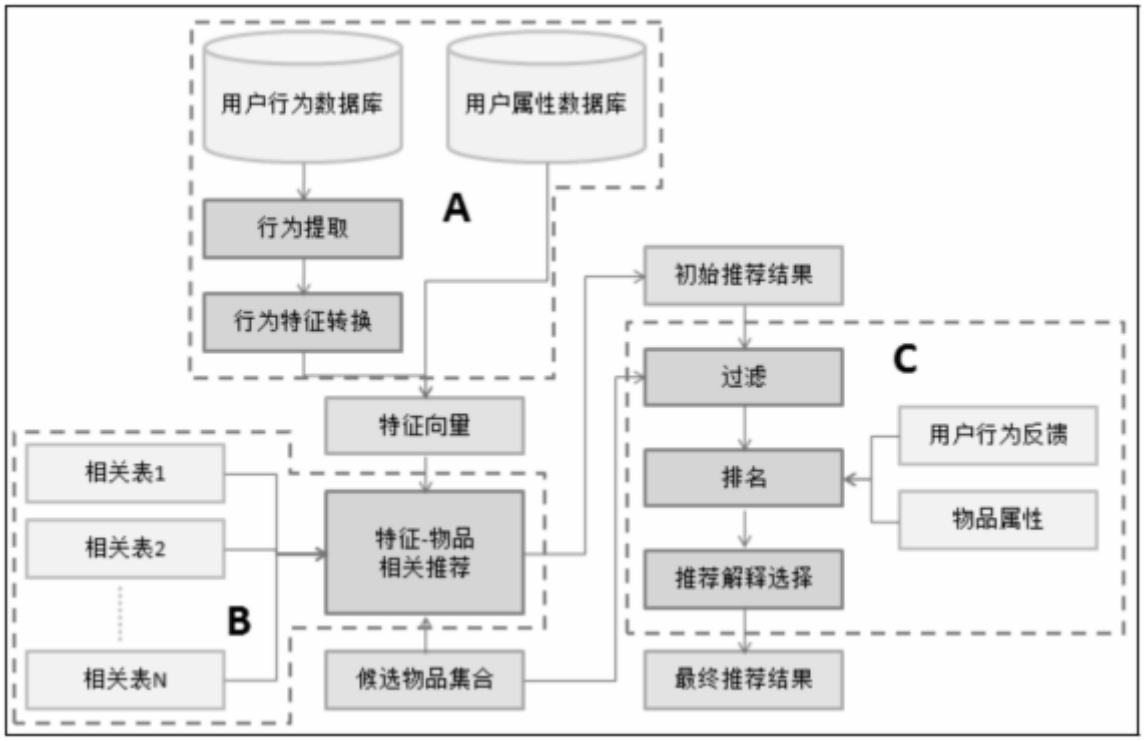


推荐系统的架构图



推荐引擎架构

- 该部分负责从数据库或者缓存中拿到用户行为数据，通过分析不同行为，生成当前用户的特征向量。不过如果是使用非行为特征，就不需要使用行为提取和分析模块了。该模块的输出是用户特征向量。
- 该部分负责将用户的特征向量通过特征-物品相关矩阵转化为初始推荐物品列表。
- 该部分负责对初始的推荐列表进行过滤、排名等处理，从而生成最终的推荐结果。



A生成用户特征向量：

- 用户行为的种类
- 用户行为产生的时间
- 用户行为的次数
- 物品的热门程度

B特征—物品相关推荐

src_id	dst_id	weight
特征ID	物品ID	权重

C过滤模块

- 用户已经产生过行为物品
- 候选物品以外的物品
- 某些质量很差的物品

C排名模块

- 新颖性排名
- 多样性
- 时间多样性
- 用户反馈

八、评分预测问题

8.2 评分预测算法

1. **平均值**
   - 全局平均值：训练集中所有评分记录的评分平均值

$$\mu = \frac{\sum_{(u,i)\in \text{Train}} r_{ui}}{\sum_{(u,i)\in \text{Train}} 1}$$

   - 用户评分平均值：用户u的评分平均值 $\overline{r}_u$ 定义为用户u在训练集中所有评分的平均值：

$$\overline{r}_u = \frac{\sum_{i\in N(u)} r_{ui}}{\sum_{i\in N(u)} 1}$$

   - 物品评分平均值：物品i的评分平均值 $\overline{r}_u$ 定义为物品i在训练集中接受的所有评分的平均值

$$\overline{r}_i = \frac{\sum_{u\in N(i)} r_{ui}}{\sum_{u\in N(i)} 1}$$

   - 用户分类对物品分类的平均值：利用训练集中同类用户对同类物品评分的平均值预测用户对物品的评分

$$\hat{r}_{ui} = \frac{\sum_{(v,j)\in \text{Train},\phi(u)=\phi(v),\varphi(i)=\varphi(j)} r_{vj}}{\sum_{(v,j)\in \text{Train},\phi(u)=\phi(v),\varphi(i)=\varphi(j)} 1}$$

   - 用户和物品的平均分
   - 用户活跃度和物品流行度
2. **基于邻域的方法**
   1. 基于用户的邻域算法：认为预测一个用户对一个物品的评分，需要参考和这个用户兴趣相似的用户对该物品的评分

$$\hat{r}_{ui} = \overline{r}_u + \frac{\sum_{v \in S(u,K) \cap N(i)} w_{uv}(r_{vi} - \overline{r}_v)}{\sum_{v \in S(u,K) \cap N(i)} |w_{uv}|}$$

$$w_{uv} = \frac{\sum_{i \in I}(r_{ui} - \overline{r}_u) \cdot (r_{vi} - \overline{r}_v)}{\sqrt{\sum_{i \in I}(r_{ui} - \overline{r}_u)^2 \sum_{i \in I}(r_{vi} - \overline{r}_v)^2}}$$

2. 基于物品的邻域算法：在预测用户u对物品i的评分时，会参考用户u对和物品i相似的其他物品的评分

$$\hat{r}_{ui} = \overline{r}_i + \frac{\sum_{j \in S(u,K) \cap N(u)} w_{ij}(r_{uj} - \overline{r}_i)}{\sum_{j \in S(i,K) \cap N(u)} |w_{ij}|}$$

第一种是普通的余弦相似度（cosine similarity）：

$$w_{ij} = \frac{\sum_{u \in U} r_{ui} \cdot r_{uj}}{\sqrt{\sum_{u \in U} r_{ui}^2 \sum_{u \in U} r_{uj}^2}}$$

第二种是皮尔逊系数（pearson correlation）：

$$w_{ij} = \frac{\sum_{u \in U}(r_{ui} - \overline{r}_i) \cdot (r_{uj} - \overline{r}_j)}{\sqrt{\sum_{u \in U}(r_{ui} - \overline{r}_i)^2 \sum_{u \in U}(r_{uj} - \overline{r}_j)^2}}$$

第三种被Sarwar称为修正的余弦相似度（adjust cosine similarity）：

$$w_{ij} = \frac{\sum_{u \in U}(r_{ui} - \overline{r}_u) \cdot (r_{uj} - \overline{r}_u)}{\sqrt{\sum_{u \in U}(r_{ui} - \overline{r}_u)^2 \sum_{u \in U}(r_{uj} - \overline{r}_u)^2}}$$

3. **隐语义模型与矩阵分解模型**
    1. 传统的SVD分解
    2. Simon Funk的SVD分解
    3. 加入偏置项后的LFM
    4. 考虑邻域影响的LFM
4. **加入时间信息**
    1. 基于邻域的模型融合时间信息
    2. 基于矩阵分解的模型融合时间信息
5. **模型融合**
    1. 模型级联融合
    2. 模型加权融合