北京交通大学

基于大数据的智能推荐系统设计与实现

个性化推荐

张迪
dizhang@bjtu.edu.cn

信息过载

- 互联网技术的飞速发展，信息呈现爆炸式的增长

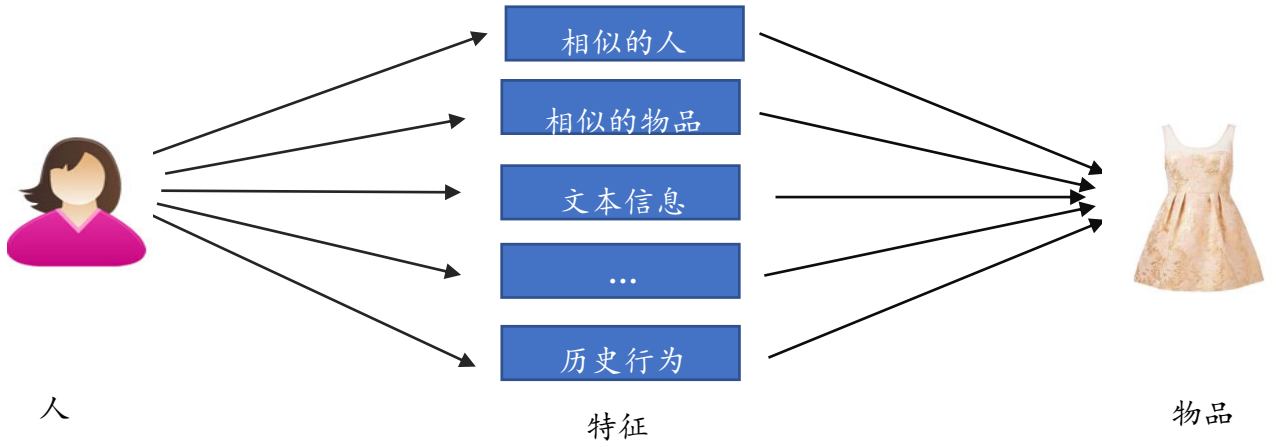
如何从海量信息中过滤出有用的信息？

信息过滤的方式

- 类目导航：用户按照类目查找
  - 代表：雅虎、新浪、搜狐、网易等
- 搜索：用户提出意图明确的查询请求
  - 代表：Google、百度等
- 推荐：系统提供给用户选择
  - 代表：亚马逊、今日头条、淘宝等
  - 推荐的优势：
    - 用户大多数的情况下并没有明确的意图
    - 推荐可以帮助用户发现，给用户带来惊喜

推荐系统

- 什么是推荐系统？
  - 通过建立用户（user）与物品（item）之间的二元关系，利用已有的选择过程或者相似关系挖掘每个用户潜在的兴趣对象，进而实现个性化的推荐，其本质就是信息过滤。
- 推荐系统的核心问题
  - 如何评估一个用户对一个物品的喜欢程度（评分）？



推荐系统的发展历史

- 1992年，Goldberg提出了第一个个性化邮件推荐系统Tapestry，第一次提出了协同过滤的思想
- 1994年明尼苏达大学推出第一个自动化推荐系统 GroupLens
- 1997 年 Resnick 等首次提出推荐系统一词（recommender system），自此推荐系统开始成为一个重要的研究领域
- 1998年亚马逊（Amazon.com）上线了基于物品的协同过滤算法，将推荐系统推向服务千万级用户和处理百万级商品的规模，并能产生质量良好的推荐
- 2003年亚马逊的Linden等人发表论文，公布了基于物品的协同过滤算法
- 2006年，Netflix举办的百万美元推荐系统算法竞赛
- 2007年，J.A. Konstan 等人组织了第一届ACM推荐系统大会(RecSys)
- 2016年，YouTube将深度神经网络应用推荐系统中，实现了从大规模可选的推荐内容中找到最有可能的推荐结果。

推荐系统的应用和价值

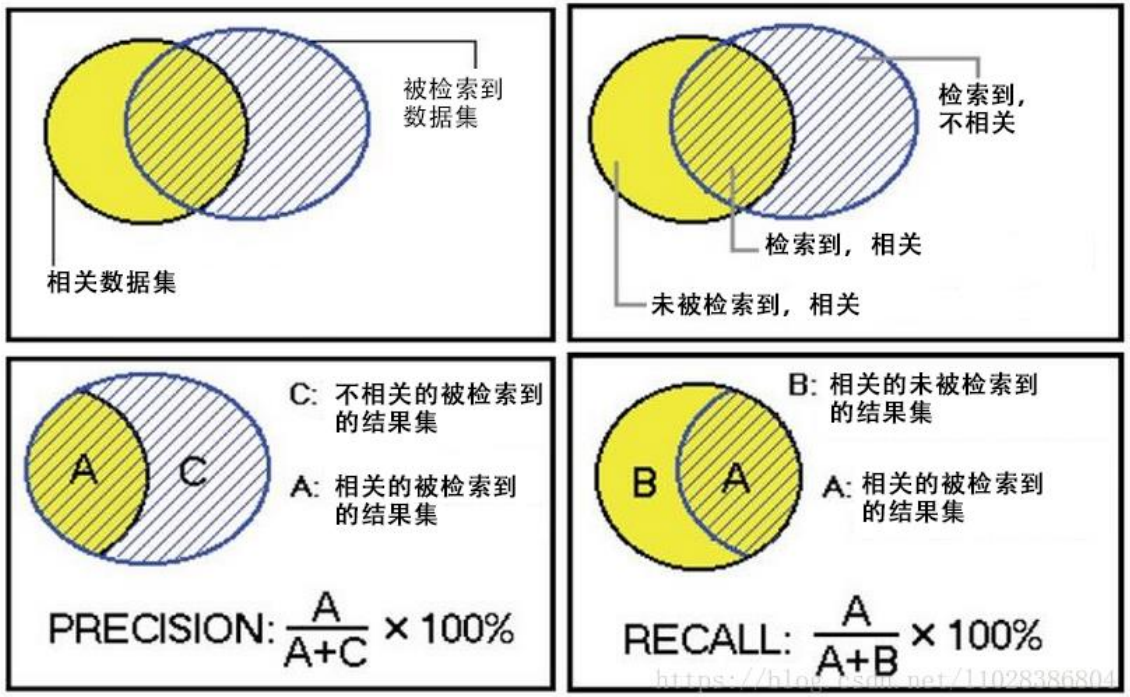
- 推荐系统的应用
  - 音乐、电影的推荐
  - 电子商务中的商品推荐
  - 个性化的阅读推荐（新闻等）
  - 社交网络好友推荐
  - 基于地理位置的服务推荐
- 推荐系统的价值
  - Netflix: 2/3 的电影是因为被推荐而观看
  - Google News: 推荐提升了38%的点击
  - Amazon: 销售中推荐占比高达35%

推荐系统的评价标准

- 用户满意度 (User Satisfaction)：调研或用户反馈；点击率、转化率等
- 准确性 (Accuracy)：precision/recall/F-score
- 覆盖率(Coverage)：照顾到尾部物品和用户
- 多样性(Diversity)：两两之间不相似
- 新颖性(Novelty)：没听过、没见过的物品
- 惊喜性(Serendipity)：如何评价？
- 用户信任度(Trust) /可解释性（explanation）：推荐理由
- 鲁棒性/健壮性(Robustness)：抗攻击、反作弊
- 实时性(Real-time/online)：新加入的物品；新的用户行为(实时意图)
- 商业目标(business target)：一个用户带来多少盈利

推荐系统的评价标准

- 准确性（Accuracy）
  - 准确率（precision）和召回率（recall）



准确率也称作查准率，召回率也称作查全率

推荐算法的分类

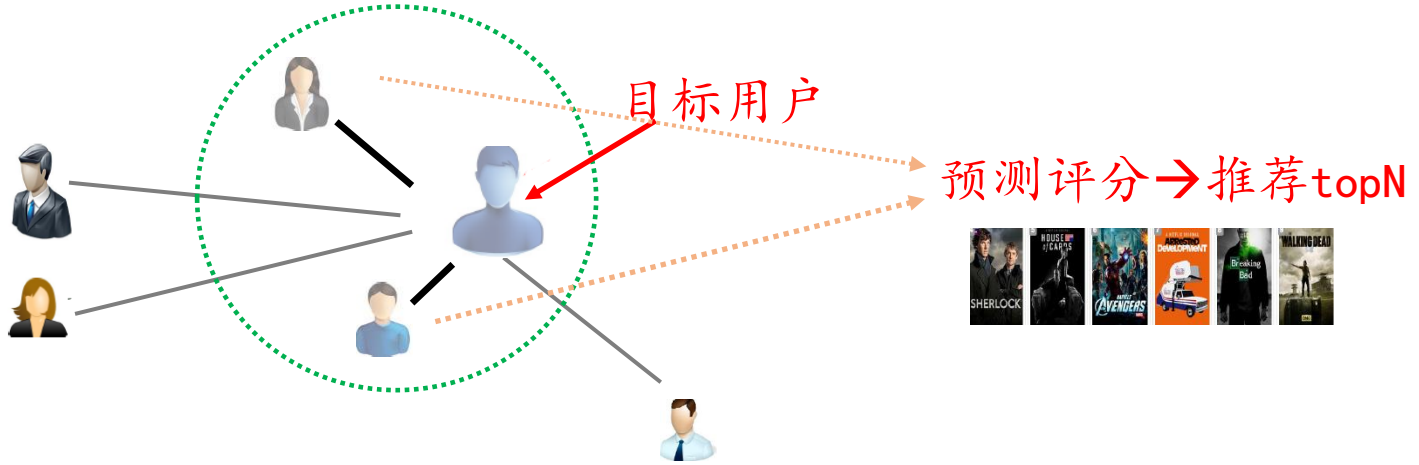
- 传统的推荐算法
  - 非个性化推荐：热度排行（Popularity）
  - 协同过滤（Collaborative Filtering）
  - 基于内容/知识的推荐（Content/Knowledge-Based）
  - 混合算法（Hybrid）
- 新的推荐算法
  - 学习排序（Learning to Rank）
  - 情景推荐：分解机（Factorization Machine）
  - 深度学习（Deep Learning）

非个性化推荐：热度排行

- 不同的排行榜算法
  - 单一或者多维度的评分
  - 考虑时间因素：按时间排行、引入衰减权重
  - 考虑反馈信息：例如用户投票的排名算法Reddit
  - 考虑置信度：例如威尔逊区间
  - 防止马太效应
- 评价
  - 优点：容易实现，可以解决新用户的冷启动问题
  - 缺点：更新很慢，很难推出新内容，不够个性化

协同过滤

- 协同过滤的基本思想
  - 思想来源：现实生活中朋友相互推荐自己喜欢的物品
  - 1. 寻找相似的用户集合；
  - 2. 寻找集合中用户喜欢的且目标用户没有的进行推荐。



协同过滤

- 基本组成元素
  - 目标用户
  - N个用户和M个物品
  - 评分矩阵
    - 显式：用户对物品的评分
    - 隐式：用户对物品的行为（例如购买记录等）
  - 相似度度量
    - 计算用户与用户（或者物品与物品）之间的相似性
    - 度量方法：Jaccard系数、皮尔逊相关系数、欧几里德距离、余弦距离等

协同过滤的分类

- 个性化or非个性化？
  - 个性化的协同过滤：基于相似用户进行预测
  - 非个性化的协同过滤：根据所有的用户进行平均预测
- 分类
  - 基于记忆的（memory-based/neighbor-based）的协同过滤
    - 基于用户的（User-based）
    - 基于物品的（Item-based）
  - 基于模型的（model-based）协同过滤
    - 聚类、分类、回归、矩阵分解、RBM、图模型

基于用户的协同过滤

- 基本思想
  - 基于用户对物品的偏好找到相邻邻居用户，然后将邻居用户喜欢的推荐给当前用户
- 算法步骤

协同

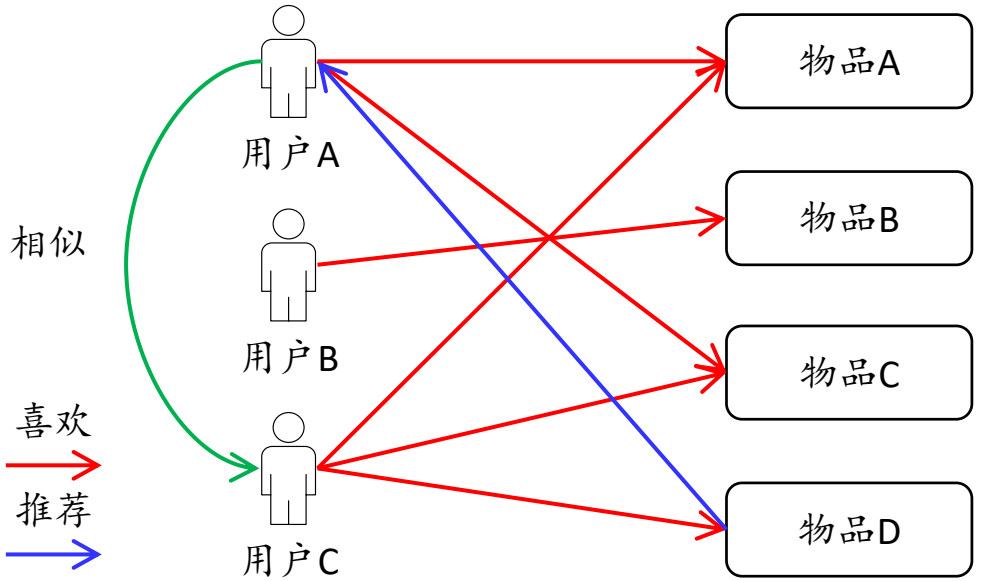
  - 1. 计算目标用户的(前k个)相似用户
    - ✓相似性度量： Pearson相关系数，Jaccard距离，cosine相似性
  - 2. 找出相似用户喜欢的物品，并预测目标用户对这些物品的评分
    - ✓预测模型：kNN , regression

过滤

  - 3. 过滤掉目标用户已经消费过的物品
    - ✓消费：购买商品，浏览新闻等
  - 4. 将剩余物品按照预测评分排序，并返回前N个物品

基于用户的协同过滤

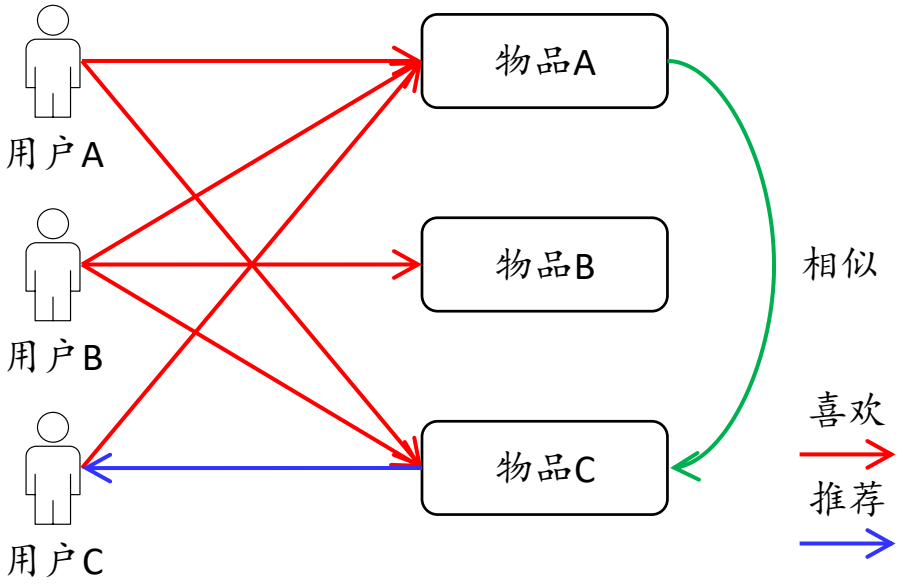
- 一个小例子



用户/物品	物品A	物品B	物品C	物品D
用户A	√		√	推荐
用户B		√		
用户C	√		√	√

基于物品的协同过滤

- 基本思想
  - 计算邻居时将用户替换成物品
  - 基于用户对物品的偏好找到相似的物品，然后根据用户的历史偏好，推荐相似的物品给他



用户/物品	物品A	物品B	物品C
用户A	√		√
用户B	√	√	√
用户C	√		推荐

User-based vs Item-based

- 基于用户的（User-based）协同过滤
  - 可以帮助用户发现新的商品，但是需要复杂的在线计算，而且需要解决新用户的冷启动问题
- 基于物品的（Item-based）协同过滤
  - 准确性好，表现稳定可控，便于离线计算，但是推荐结果的多样性会差，无法给用户带来惊喜
- 相似性的差异
  - Item之间的相似性比较单纯，是静态的
  - User之间的相似性比较复杂，是动态的

协同过滤的优缺点

- 优点
  - 模型的通用性强，需要很少的领域知识
  - 工程上实现简单，效果很好
- 缺点
  - 冷启动问题（新用户或者物品）
  - 数据的稀疏性问题
  - 假定“过去的行为决定现在”，没有考虑具体情境的差异
  - 热门倾向性（Popularity Bias）：很难推荐出小众的偏好

基于模型的协同过滤

- 关联规则（Associate Rule）
- 聚类（Clustering）
- 分类/回归
- 矩阵分解：SVD、SVD++
- 受限玻尔兹曼机（RBM）
- 图模型，如SimRank，Markov Decision Process

关联规则（Associate Rule）

- 基本思想：基于物品之间的共现性挖掘频繁项
- 一般应用场景：看了又看、买了又买、商品搭配
- 算法：A-priori、FP-Growth

支持度 (support) $$s(X,Y)=\frac{包含(X,Y)的记录数}{总记录数}$$

置信度(confidence) $$c(X,Y)=\frac{包含(X,Y)的记录数}{包含X的记录数}$$

- 评价
  - 实现简单，通用性强，适合“推荐跟已购买商品搭配的商品”
  - 相似商品的推荐效果往往不如协同过滤好
  - 关联关系可能受一些隐含因素的影响

聚类（Clustering）

- 用户分类：按照喜好对用户进行分类
  - 有时可以给用户带来惊喜，但个性化稍差（群体 vs 个体）
- 物品聚类：相似商品
  - 精准性较高，但推荐的商品对用户来说无新鲜感
- 常用算法：K-means，层次聚类等
- 评价
  - 聚类可以一定程度上解决数据稀疏性问题
  - 聚类的精准度没有协同过滤算法好

分类/回归

- 基本思想
  - 把评分预测看做一个多分类（回归）问题
- 常用的分类器
  - 逻辑回归(Logistic Regression, LR)
  - 朴素贝叶斯(Naive Bayes)
- 输入：通常是物品的特征向量
- 评价
  - 比较通用，可以跟其他方法组合，提高预测的准确性
  - 需要大量的训练数据，防止过拟合现象

分类/回归

- 逻辑回归的三个步骤
  - 提取特征值
  - 通过用户的偏好矩阵，不断地拟合得到每个特征的权重
  - 预测新用户对物品的喜好程度
- 举个例子

姓名	个性开朗程度	颜值	喜爱程度
小红	1	9	45%
小绿	2	8	40%
小黄	9	5	30%
……	……	……	……
翠花	3	8	42%

基于内容的推荐

- 基本思想
  - 根据推荐物品的元数据，发现物品的相关性，然后基于用户以往的喜好记录，推荐给用户相似的物品
- 内容：物品的元数据
  - 文本描述：通常用NLP技术挖掘关键词
  - 物品的属性：电影的主题、衣服的材质等
  - 物品的特征：例如语音的信号表示、图像的向量表示等
- 基本组成
  - 物品的特征向量
  - 用户的profile向量：通过用户偏好的物品提取
  - 匹配分数：计算cosine，分类/回归模型等

基于内容的推荐

- 优点
  - 能够推荐出用户独有的小众偏好
  - 可以在一定程度上解决数据稀疏和物品的冷启动问题
  - 通常具有较好的可解释性
- 缺点
  - 对于很多推荐问题来说，提取有意义的特征并不容易
  - 很难将不同物品的特征组合在一起（领域思想）
  - 很难带给用户惊喜
  - 如果用户的profile挖掘不准，推荐的效果往往很差

混合方法（Hybrid approaches）

- 基本思想：不同的推荐算法相互融合在一起使用
- 融合方法
  - 加权 (weighted)组合
  - 切换 (switching) : 确定一个合理的切换跳进
  - 混合(Mixed)
  - 特征组合(feature combination)：不同模型特征组合到一起
  - 特征扩展 (feature augmentation)：一个模型的输出作为另一个的特征
  - 级联(Cascade)
- 评价
  - 通常比单个算法的性能好，但是需要在不同的算法之间进行权衡

最新的推荐算法研究

- 学习排序Learning to Rank
  - 推荐结果大多数是一个列表
  - L2R将排序看做一个机器学习问题
- 上下文感知（context-aware）的推荐
  - 订餐：餐厅的距离
  - 网购：用户的心情
  - 母婴：孩子的年龄
- 基于深度学习的推荐算法
  - 卷积神经网络（CNN）、循环神经网络（RNN）

Apache Mahout

- 单机算法
  - GenericUserBasedRecommender: 基于用户的推荐算法
  - GenericItemBasedRecommender: 基于物品的推荐算法
  - KnnItemBasedRecommender: 基于物品的KNN推荐算法
  - SlopeOneRecommender: Slope推荐算法
  - SVDRecommender: SVD推荐算法
  - TreeClusteringRecommender： TreeCluster推荐算法
- 基于Hadoop的分布式算法
  - 基于物品的协同过滤推荐算法

任务

- 基于Mahout实现基于物品的协同过滤推荐算法
  - 基于自己搭建的Hadoop平台
  - 此任务必须完成
- 基于Mahout实现图书电商系统中的个性化推荐算法
  - 可以有多种不同的推荐方式
  - 可以采用混合式的推荐算法

Thank you!

Q & A