推荐系统实践

2019年3月23日 10:55

推荐系统基本概念

- 推荐系统与搜索引擎比较
  - 推荐系统和搜索引擎都是用来解决信息过载问题
  - 搜索引擎满足用户有明确目的时主动查找的寻求
  - 推荐系统在用户没有明确的目的是帮助他们发现感兴趣的新内容
- 推荐系统实验方法
  - 离线实验
  - 用户调查
  - 在线实验
- 评测指标
  - 用户满意度
  - 预测准确度
  - 覆盖率
  - 多样性
  - 新颖度
  - 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，那么就可以说推荐结果的惊喜度很高，而推荐的新颖性仅仅取决于用户是否听说过这个推荐结果
  - 信任度
  - 实时性
  - 健壮性

基于用户行为数据的推荐算法

用户行为数据

- 无上下文信息的隐性反馈数据集
- 无上下文信息的显性反馈数据集
- 有上下文信息的隐性反馈数据集
- 有上下文信息的显性反馈数据集

评测指标(TopN推荐)

- 召回率：$\mathrm{Recall} = \frac{\sum_u |R(u) \cap T(u)|}{\sum_u |T(u)|}$
- 准确率：$\mathrm{Precision} = \frac{\sum_u |R(u) \cap T(u)|}{\sum_u |R(u)|}$
- 覆盖率：$\mathrm{Coverage} = \frac{|\cup_u R(u)|}{|I|}$
  - R(u)表示对用户推荐的N个物品
  - T(u)表示测试集用户喜欢的物品集合
  - I表示所有的物品

基于邻域的算法

- 基于用户的协同过滤
  - 找到和目标用户兴趣相似的用户集合
    - Jaccard用户兴趣相似度

$$w_{uv} = \frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|}$$

- 余弦相似度

$$w_{uv} = \frac{|N(u) \cap N(v)|}{\sqrt{|N(u)||N(v)|}}$$

- 用户兴趣相似度改进
- $$w_{uv} = \frac{\sum_{i \in N(u) \cap N(v)} \frac{1}{\log(1+|N(i)|)}}{\sqrt{|N(u)||N(v)|}}$$

  - 找到这个集合中的用户喜欢的，但是目标用户没有听说过的物品推荐给目标用户
    - 用户u对物品i的感兴趣程度

$$p_{ui} = \sum_{v \in S(u,K) \cap N(i)} w_{uv} r_{vi}$$

N(u)表示用户u有过正反馈的物品集合

r_{vi}表示用户v对物品i的兴趣

- 基于物品的协同过滤
  - 计算物品之间的相似度
    - 物品的相似度

$$w_{ij} = \frac{|N(i) \cap N(j)|}{|N(i)|}$$

- 余弦相似度

$$w_{ij} = \frac{|N(i) \cap N(j)|}{\sqrt{|N(i)||N(j)|}}$$

- 用户活跃度对物品相似度的影响

$$w_{ij} = \frac{\sum_{u \in N(i) \cap N(j)} \frac{1}{\log(1+|N(u)|)}}{\sqrt{|N(i)||N(j)|}}$$

  - 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
    - 用户u对一个物品j的兴趣

$$p_{uj} = \sum_{i \in N(u) \cap S(j,K)} w_{ji} r_{ui}$$

N(i)表示喜欢物品i的用户集合

r_{ui}表示用户u对物品i的兴趣

隐语义模型

评分预测问题

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

平均值

- 全局平均值
  - $\mu = \frac{\sum_{(u,i) \in train} r_{ui}}{\sum_{(u,i) \in train} 1}$
  - $\widehat{r_{ui}} = \mu$
- 用户评分平均值
  - $\bar{r_u} = \frac{\sum_{i \in N(u)} r_{ui}}{\sum_{i \in N(u)} 1}$
  - $\widehat{r_{ui}} = \bar{r_u}$
- 物品评分平均值
  - $\bar{r_i} = \frac{\sum_{u \in N(i)} r_{ui}}{\sum_{u \in N(i)} 1}$

- $\widehat{r_{ui}} = \bar{r_i}$
- 用户分类对物品分类的平均值
  - $\widehat{r_{ui}} = \frac{\sum_{(v,j)\in train, \phi(u)=\phi(v), \varphi(i)=\varphi(j)} r_{vj}}{\sum_{(v,j)\in train, \phi(u)=\phi(v), \varphi(i)=\varphi(j)} 1}$

基于邻域的方法

- 基于用户的邻域算法
  - 通过皮尔逊系数计算用户之间的相似度
    - $w_{uv} = \frac{\sum_{i\in I}(r_{ui} - \bar{r_u})(r_{vi} - \bar{r_v})}{\sqrt{\sum_{i\in I}(r_{ui} - \bar{r_u})^2 \sum_{i\in I}(r_{vi} - \bar{r_v})^2}}$
  - 用户对该物品的评分
    - $\widehat{r_{ui}} = \bar{r_u} + \frac{\sum_{v\in S(u,K)\cap N(i)} w_{uv}(r_{vj} - \bar{r_v})}{\sum_{v\in S(u,K)\cap N(i)} |w_{uv}|}$
- 基于物品的邻域算法
  - 余弦相似度
    - $w_{ij} = \frac{\sum_{u\in U} r_{ui} r_{uj}}{\sqrt{\sum_{u\in U} r_{ui}^2 \sum_{u\in U} r_{uj}^2}}$
  - 皮尔逊相似度
    - $w_{ij} = \frac{\sum_{u\in U}(r_{ui} - \bar{r_i})(r_{uj} - \bar{r_j})}{\sqrt{\sum_{u\in U}(r_{ui} - \bar{r_i})^2 \sum_{u\in U}(r_{uj} - \bar{r_j})^2}}$
  - 修正的余弦相似度
    - $w_{ij} = \frac{\sum_{u\in U}(r_{ui} - \bar{r_u})(r_{uj} - \bar{r_u})}{\sqrt{\sum_{u\in U}(r_{ui} - \bar{r_u})^2 \sum_{u\in U}(r_{uj} - \bar{r_u})^2}}$
  - 用户对物品的评分
    - $\widehat{r_{ui}} = \bar{r_i} + \frac{\sum_{j\in S(u,K)\cap N(u)} w_{ij}(r_{uj} - \bar{r_j})}{\sum_{j\in S(i,K)\cap N(u)} |w_{ij}|}$

隐语义模型与矩阵分解模型

用户的评分行为可以表示成一个评分矩阵R，其中$R[u][i]$就是用户u对物品i的评分。但是，用户不会对所有的物品评分，所以这个矩阵里有很多元素都是空的，这些空的元素称为缺失值（missing value）

- 传统的SVD分解
  - 首先对评分矩阵的缺失值进行简单的补全，比如全局平均值，或者用户/物品平均值补全，得到补全之后的矩阵R'，利用SVD进行分解：

    $$R' = U^T S V$$

    其中$U \in R^{k*m}$，$V \in R^{k*n}$，$S \in R^{k*k}$对角线上每个元素都是矩阵的奇异值
  - 为了对R'进行降维，可以取最大的f个奇异值组成对角矩阵S_f，并且找到这f个奇异值中每个值在U、V矩阵中对应的行和列，得到U_f、V_f，从而可以得到一个降维后的评分矩阵：

    $$R'_f = U_f^T S_f V_f$$

    $R'_f(u,i)$ 就是用户u对物品i评分的预测值
- 隐语义模型

  从矩阵分解角度来说，LFM就是将评分矩阵R分解成为两个低维矩阵相乘：
  - $\hat{R} = P^T Q$

  则用户u对物品i的评分的预测值为：
  - $\widehat{r_{ui}} = \sum_f p_{uf} q_{if}$

  LFM通过训练集中的观察值来最小化RMSE学习P、Q矩阵
  - 损失函数

- $C(p,q) = \sum_{(u,i)\in train} (r_{ui} - \widehat{r_{ui}})^2 = \sum_{(u,i)\in train} \left(r_{ui} - \sum_{f=1}^{F} p_{uf} q_{if} \right)^2$

○ 损失函数正则化

- $C(p,q) = \sum_{(u,i)\in train} \left(r_{ui} - \sum_{f=1}^{F} p_{uf} q_{if} \right)^2 + \lambda(\|p_u\| + \|q_i\|)^2$

○ 随机梯度下降法

- $\frac{\partial C}{\partial p_{uf}} = -2q_{if} + 2\lambda p_{uf}$
- $\frac{\partial C}{\partial q_{if}} = -2p_{uf} + 2\lambda q_{if}$
- $p_{uf} = p_{uf} + \alpha(q_{ik} - \lambda p_{uk})$
- $q_{if} = q_{if} + \alpha(p_{uk} - \lambda q_{ik})$