

[技术杂谈] 《机器学习算法实践:推荐系统的协同过滤理论及其应用》 评分:7.8

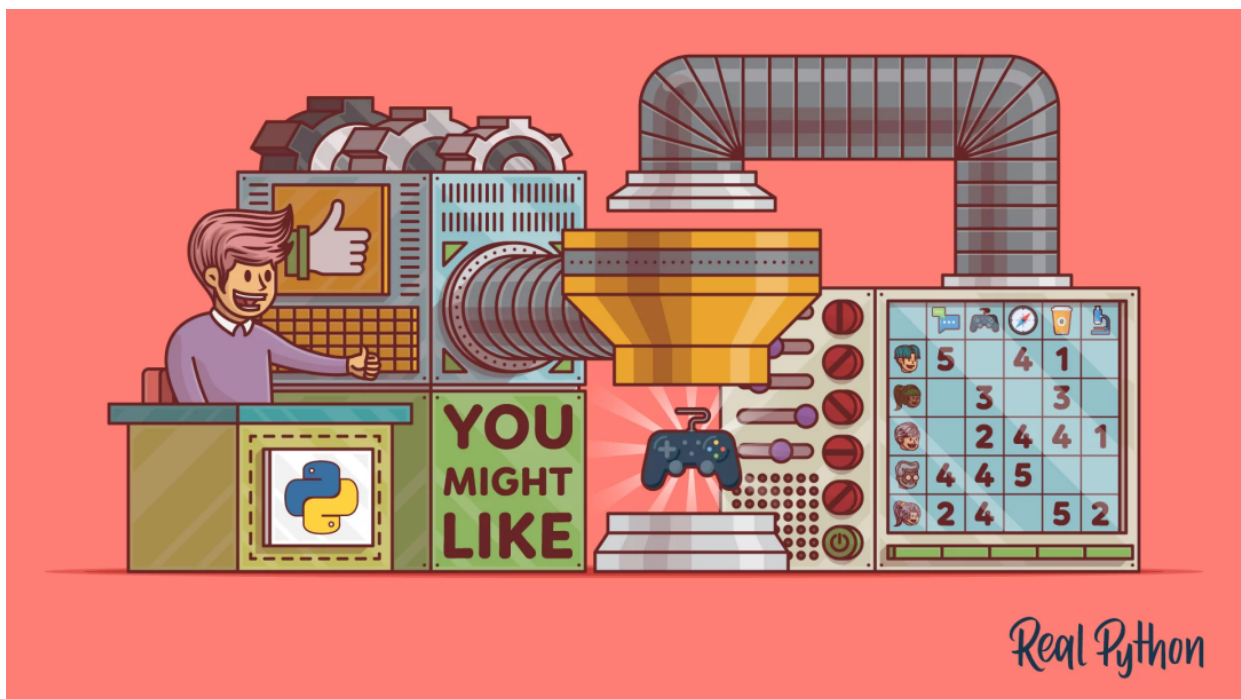
原创 erixhao 未来阅读空间 2020-04-04

收录于话题

#技术博文

22个

作者: 王建芳 评分: 7.8 阅读时间: 2020年3.25- 4.05日

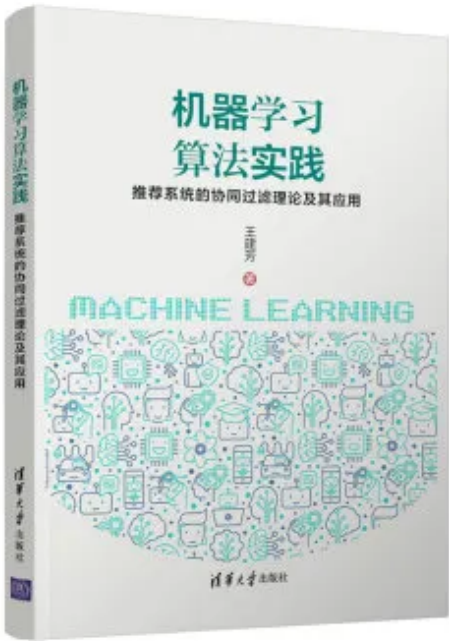


“可惜,

世界上最聪明的大脑, 想的都是如何推荐,
怎么样让用户去点击广告。”

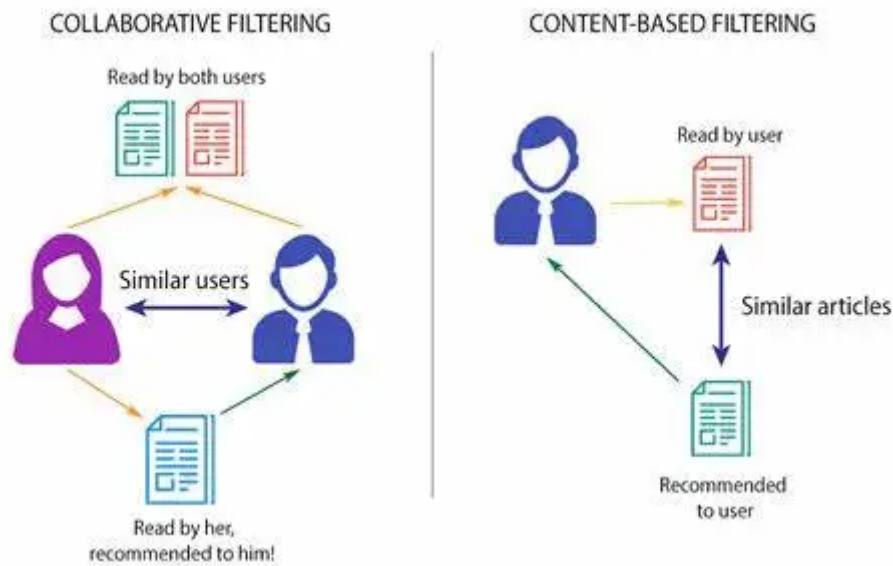
— [X] Anonymous

简评



个性化推荐能够根据用户的历史行为**显式**或者**隐式**地挖掘用户潜在的兴趣和需求, 是业界近几年(10+几年)较为流行的算法, 尤其是在**电商平台, 广告推荐投放, 影视书籍推荐, 如Netflix**, 本书《机器学习算法实践:推荐系统的协同过滤理论及其应用》较全面地介绍了**基于协同过滤的推荐系统**存在的问题、解决方法和评估策略, 主要内容涉及协同过滤推荐算法中的时序技术、矩阵分解技术和社交网络信任技术等知识。

目前推荐系统常采用的方法主要有基于内容的推荐、基于网格的推荐、基于上下文情景的推荐和**协同过滤推荐**。



协同过滤（Collaborative Filtering, CF） 推荐技术是推荐系统中最为常用且有效的方法，它的任务是利用用户与项目评分矩阵中的已知元素来预测未知元素的评分值并将预测评分高的项目推荐给用户。

- 基于近邻的协同过滤推荐算法
- 基于用户兴趣的推荐算法
- 基于模型的协同过滤推荐算法
 - a. 矩阵分解模型
 - b. 交替最小二乘
 - c. 概率矩阵分解
- SVD和信任因子相结合的协同过滤推荐算法

2

作者: 王建芳

王建芳，男，博士，河南理工大学副教授，硕士研究生导师。研究方向包括推荐系统、深度学习、人工智能及智能计算算法。

3

推荐系统

基础概念

推荐系统的传统定义可以理解为“采集用户历史行为信息，结合具体推荐模型帮助用户选择商品或提供建议的过程”。

现阶段完整的个性化推荐模型主要由**数据收集及预处理**、**推荐算法**和**产生推荐**三部分组成。

协同过滤 (Collaborative Filtering, CF)

协同过滤 (Collaborative Filtering, CF) 推荐技术是推荐系统中最为常用且有效的方法，可分为基于内存的协同过滤和基于模型的协同过滤，前者根据**用户**或者**项目**的相似度选出与目标用户最相似的若干用户的评分来对未评分的项目进行评分预测；后者通过分析用户和项目的**内部规律模型**，预测用户对项目的偏好，其中**概率矩阵分解**技术是其典型代表。

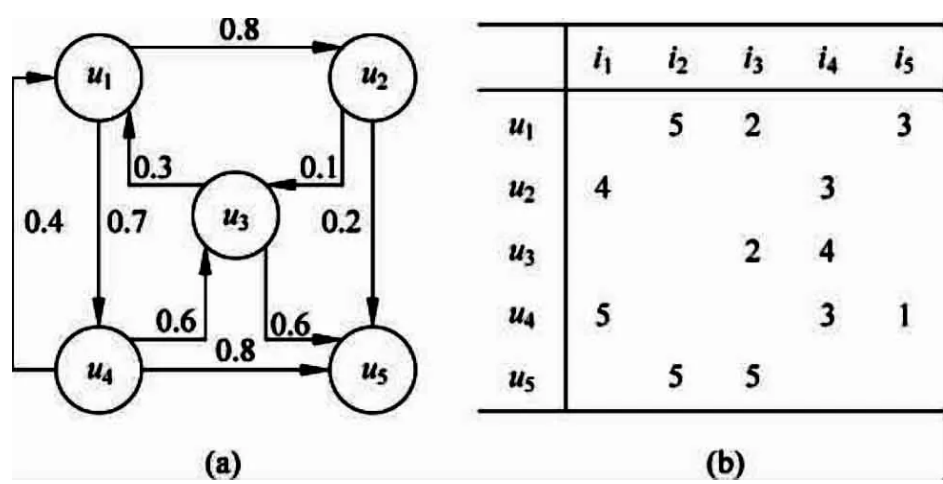
目前概率矩阵分解技术还存在数据的高维稀疏性和海量数据环境下的扩展性等制约其进一步发展的瓶颈问题。

一个典型的电影推荐系统一般包括含有N个用户的用户集合 $U= \{u_1, u_2, u_3, ..., u_N\}$ 和含有M个项目的项目集合 $I= \{i_1, i_2, i_3, ..., i_M\}$ ，每个用户 $u_i \in U$ 评价了I中的一部分项目，评价过的项目用 $I_{ui} \subseteq I$ 表示，用户的打分记录往往表示成RNM

$$R_{NM} = \begin{bmatrix} r_{11} & \cdots & r_{1k} & \cdots & r_{1M} \\ \vdots & & \vdots & & \vdots \\ r_{21} & \cdots & r_{2k} & \cdots & r_{2M} \\ \vdots & & \vdots & & \vdots \\ r_{N1} & \cdots & r_{Nk} & \cdots & r_{NM} \end{bmatrix}$$

(1-1)

每一行 r_i ——用户 i 评价过的电影集合，所有用户集合用 U 表示；每一列 r_j ——评价电影 j 的用户集合，所有电影集合用 V 表示；每一个元素 r_{ij} ——用户 i 对电影 j 的评分，通常 r_{ij} 的取值为1~5的整数，数据越大表示用户对该项目越满意。实际中**RNM非常稀疏**，因此传统推荐算法的质量才会特别差。



推荐算法的形式化定义如式:

$$\forall p \in P, \quad \forall q \in Q, \quad q'_p = \arg \max_{q \in Q} u(p, q)$$

未来阅读空间

式中， P ——用户集合； Q ——能够推荐给用户的物品集合； u ——一个用来计算用户 p 对物品 q 偏好程度的效用函数，计算过程可以表示为 $u: P \times Q \rightarrow R$ ，其中 R 为排序后的项目集合。


算法的目标是对于每个用户 p 都找到能够最大化效用函数 u 的物品子集 $Qq'p \in Q$ 。

基于近邻的协同过滤推荐算法

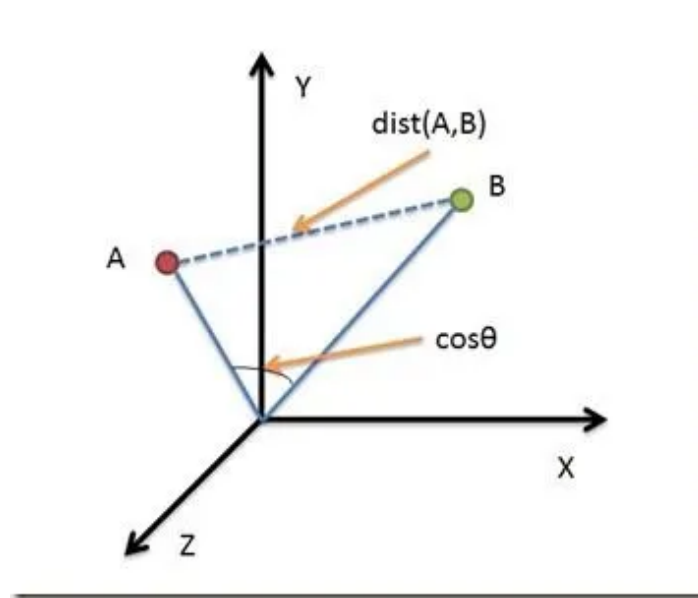
基于近邻的协同过滤推荐算法是一种非常流行的建立推荐系统的方式，仅仅通过收集相似用户的行为。

余弦相似度定义向量 a 和向量 b 为RNM中的第 u 行和第 v 行，将两个向量的夹角余弦值定义为用户 u 和用户 v 的相似度，如式（1-3）所示。

$$\text{sim}(u, v) = \cos(a, b) = \frac{a \cdot b}{|a| |b|}$$

 未来阅读空间

$\text{sim}(u, v)$ 的值越接近1, 说明用户 u 与用户 v 的相似度越高。



欧氏距离衡量的是空间各点的绝对距离, 跟各个点所在的位置坐标直接相关; 而余弦距离衡量的是空间向量的夹角, 更加体现在方向上的差异, 而不是位置。

修正的余弦相似度鉴于传统的余弦相似度考虑了用户的评分偏好。也就是说, 有的用户倾向于评高分, 有的用户倾向于评低分。

$$\text{sim}(u, v) = \frac{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u) \times (r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_u} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_v} (r_{vi} - \bar{r}_v)^2}} \quad (1-4)$$

Pearson相似度

Pearson相似度和修正余弦相似度不同的是分母为用户的共同评分项目。

$$\text{sim}(u,v) = \frac{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u) \times (r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_{uv}} (r_{vi} - \bar{r}_v)^2}} \tag{1-5}$$

基于模型的协同过滤推荐算法

1. 矩阵分解模型

基于模型的协同过滤一般分为聚类模型、分类模型和**矩阵分解模型**等。本书主要研究其中的矩阵分解模型及其与社交网络的信任相结合的算法。

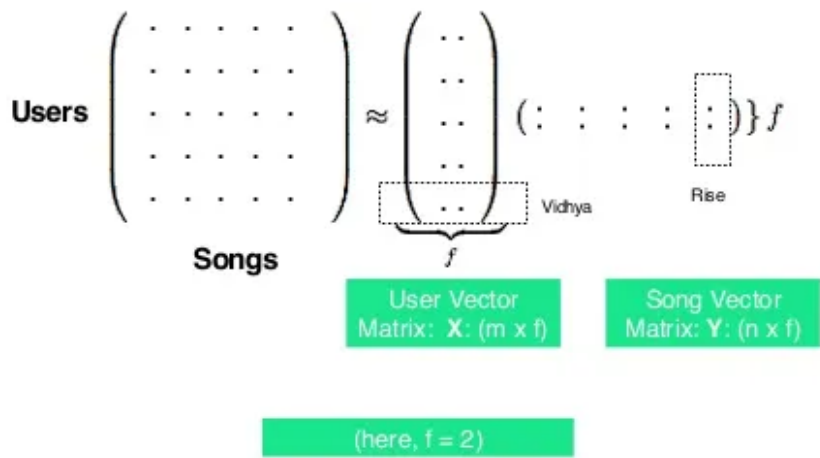
传统的矩阵分解模型有**奇异值分解 (Singular Value Decomposition, SVD)**、**概率矩阵分解 (Probabilistic Matrix Factorization, PMF)** 和非负矩阵分解 (Non-negative Matrix Factorization, NMF) 等。

概率矩阵分解算法，该算法从概率的角度来预测用户的评分，假设用户潜在因子矩阵和项目潜在因子矩阵均服从均值为0的球形高斯先验分布，在此假设的基础上，结合概率论和矩阵论的相关理论来预测用户对项目的偏好。

矩阵分解模型假设用户对项目的评分受到若干**潜在因子**的影响，将用户和项目映射到一个共同的**潜在因子空间**。该类算法到底受哪种因素的影响却很难确定，正是囿于此种缺陷，一般又将矩阵分解模型称为隐语义模型。

Latent Factor Models

“Compact” representation for each user and items(songs): f-dimensional vectors



传统的矩阵分解模型往往将固有的用户项目评分矩阵 R_{NM} 分解为两个低秩矩阵的乘积，以达到对矩阵中缺失值的预测目的：

$$R_{NM} \approx U_{kN}^T V_{kM}$$

 未来模型空间

其中， $k \ll \min (M, N)$ ，指的是潜在因子的数量； U_{kM} 和 V_{kN} 为由分解得到的两个低秩矩阵，可以看作是用户潜在因子矩阵和项目潜在因子矩阵，往往通过迭代训练来使得 U_{kM} 和 V_{kN} 的内积不断逼近原始的用户项目评分矩阵，同时得到 U_{kM} 和 V_{kN} 后还可以对用户没有评分的项目进行评分预测。

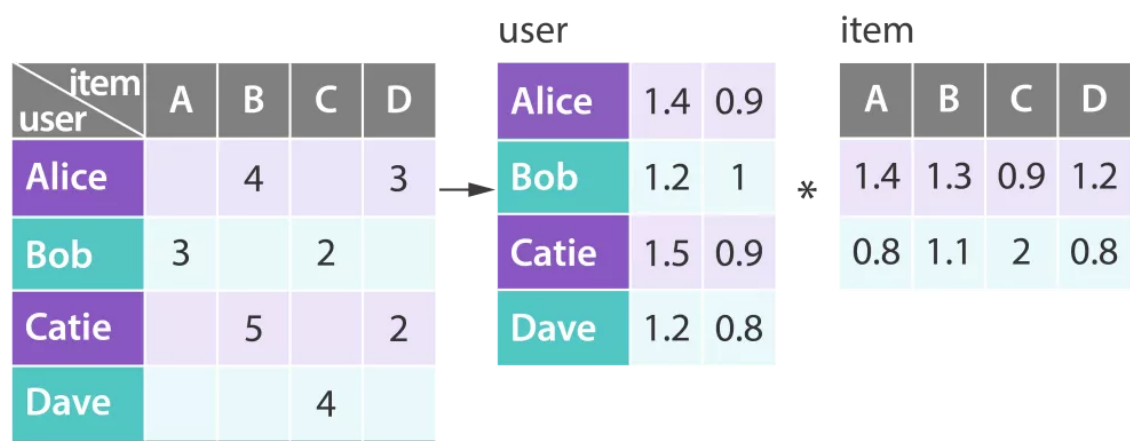
基于矩阵分解的算法是一种学习型算法，实际中往往采用**随机梯度下降（Stochastic Gradient Descent, SGD）**来优化预先设定的目标函数从而得到全局最优解，而且由于潜在因子的数量 $k \ll \min (M, N)$ ，算法的离线计算的空间复杂度低，这在当今大数据的环境下具有很强的实用价值；同时，由于该算法有一个全局的目标函数，使得算法的预测准确率高。

2. 交替最小二乘

理论研究表明交替最小二乘（Alternating Least Squares, ALS）随着迭代的进行误差会逐步降低直至收敛，ALS完全不能保证将会收敛至全局最优解，而且在实际应用中，ALS对初始点选取较为敏感，不恰当的选择会导致数据振荡地收敛到局部最优解。


<https://stanford.edu/~rezab/classes/cme323/S15/notes/lec14.pdf>

stanford




首先按高斯分布初始化用户和项目的潜在因子向量U和V。然后固定V，将损失函数对V求偏导，并令导数等于0，得到新的用户潜在因子向量U，如式（1-9）所示：

$$U \leftarrow (V^T V + \lambda I)^{-1} V^T R$$

 未来网络空间

其次固定U，将损失函数对U求偏导，并令导数等于0

$$V \leftarrow (U^T U + \lambda I)^{-1} U^T R$$

 未来网络空间

式中，λ——正则化系数，需要实验确定。最后便可利用得到的用户项目潜在因子空间U和V进行评分预测。

推荐系统现存问题

矩阵分解应用在推荐系统中目前存在四类问题，即冷启动问题、数据稀疏性问题、可扩展性问题和易受攻击性问题。冷启动主要是为了解决新用户和新项目的推荐问题，易受攻击主要是为了缓解用户的恶意评分，从而提升相关用户的知名度或者提升相关项目被推荐的次数。

Bedi等人利用Facebook社交网络上用户之间的互动，试图处理冷启动问题。Facebook是一个很受欢迎的社交网站，朋友或熟人的选择往往会影响用户的意见或选择，可以利用这个思想来为用户提供推荐；提出一个**IBSP算法，利用社会交往因子克服冷启动问题**；使用Java开发的一个图书原型系统，用Facebook的API图形从用户的社交图中提取信息。于洪等人利用用户注册信息（年龄、性别、职业、民族、居住地等）和项目内容信息（项目的详细描述）分别进行建模，提供推荐服务。Le等人提出一种新的相似度度量方法——NHSM来解决用户冷启动问题。

在传统推荐算法的研究过程中，往往具有海量的用户和项目信息。也就是说，用户和项目的**潜在因子矩阵是高维稀疏的**，由此导致任意两个向量之间近似正交，计算得到的相似度往往为零，传统的基于相似度计算的模型将得不到理想的结果。因此，评价数据集的稀疏度显得十分必要，实际应用中往往采用用户项目评分矩阵中未评分数据量占评分总量的比例作为稀疏度的衡量指标，**稀疏度越大，传统算法的精度越低，也就越难处理。**

测评指标

推荐算法的评测指标很多，针对推荐系统的侧重点不同，其评价标准也不尽相同。衡量算法的评分预测准确度时多采用**平均绝对误差（Mean Absolute Error, MAE）**和**均方根误差（Root Mean Square Error, RMSE）**指标；

针对Top-N推荐的预测准确率时一般通过**准确率（Precision）**、召回率（Recall）度量F值（F-Measure）和P（u）@N指标。

$$\text{Precision} = \frac{\sum_{u \in U} |R_u \cap T_u|}{\sum_{u \in U} |R_u|} \quad (1-19)$$

为了衡量推荐系统发掘长尾的能力，使用覆盖率（Coverage）来计算系统所推荐的项目占项目集合的比例，这是商家最关心的指标，其定义如式（1-22）所示。

$$\text{Coverage} = \frac{|U_{u \in U} R_u|}{|n|}$$

未来阅读空间

基于时序的协同过滤推荐算法

由于传统的协同过滤算法忽略了随着时间变化而用户的兴趣也在不断发生变化这一问题，即存在用户兴趣漂移现象。传统的协同过滤推荐算法只是单一地通过评分来分析用户的兴趣爱好，统一地将评分用1~5分代表用户的喜爱程度，其时效性不足。

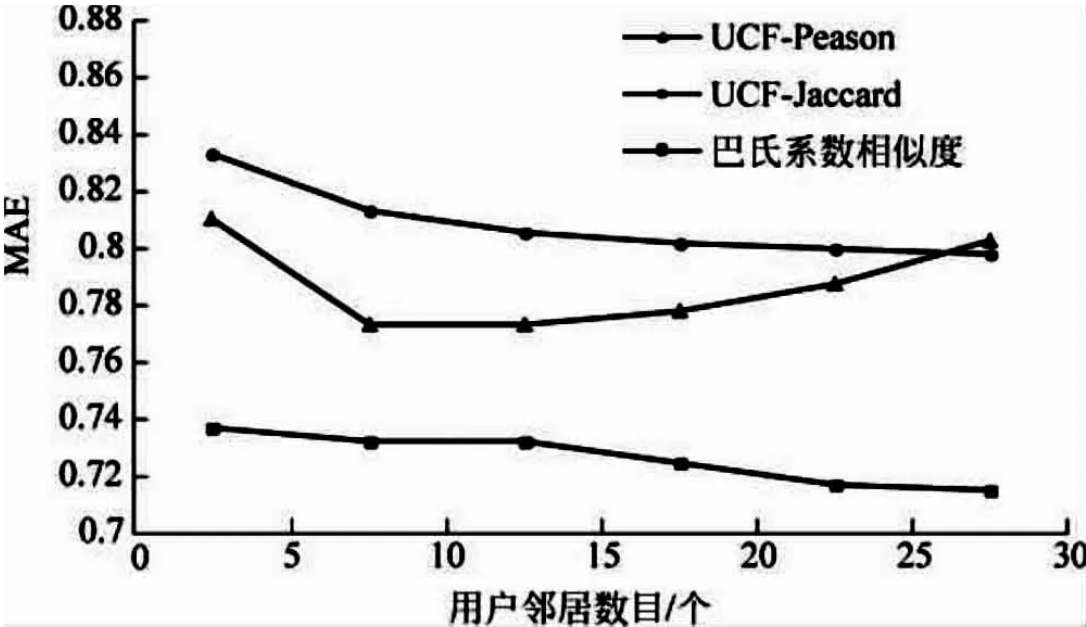
随着时间推移，用户的关注点在不断变化，如何捕获这一动态的时间效应是一个难题。

基于巴氏系数改进相似度的协同过滤推荐算法

针对传统协同过滤推荐算法中评分数据稀疏性及所造成的推荐质量不高问题，提出一种**巴氏系数 (Bhattacharyya Coefficient)** 改进相似度的协同过滤推荐算法。在基于近邻协同过滤推荐算法基础上，利用巴氏系数改进相似度计算方法，在计算相似度时不仅依靠两个用户间的共同评分而是所有评分，首先利用Jaccard相似度来计算用户间的全局相似度；其次使用巴氏系数获得评分分布的整体规律，并结合Pearson相关系数来计算其局部相似度；最后融合全局相似度和局部相似度得到最终的相似度矩阵。

巴氏系数 (Bhattacharyya Coefficient, BC) 是对两个统计样本的重叠的近似计算，**可用来对两组样本的相关性进行测量**，已广泛应用于信号处理、模式识别研究领域。在统计学中巴氏系数用于测量两种离散概率分布的可离性，衡量两个概率分布之间的相似度。

随着用户邻居数目 k 的增大，BCCF算法其MAE值总体趋于减少，当邻居数达到30时MAE值基本稳定；相对于传统的Jaccard和Pearson相关系数计算方法，BCCF算法的MAE值最低且相对稳定。



基于矩阵分解的协同过滤推荐算法

矩阵分解模型最早由Yehuda Koren于2008年提出，主要目的是找到**两个低维的矩阵**，它们相乘之后得到的矩阵的近似值，与评分矩阵中原有值的位置中的值尽可能接近。

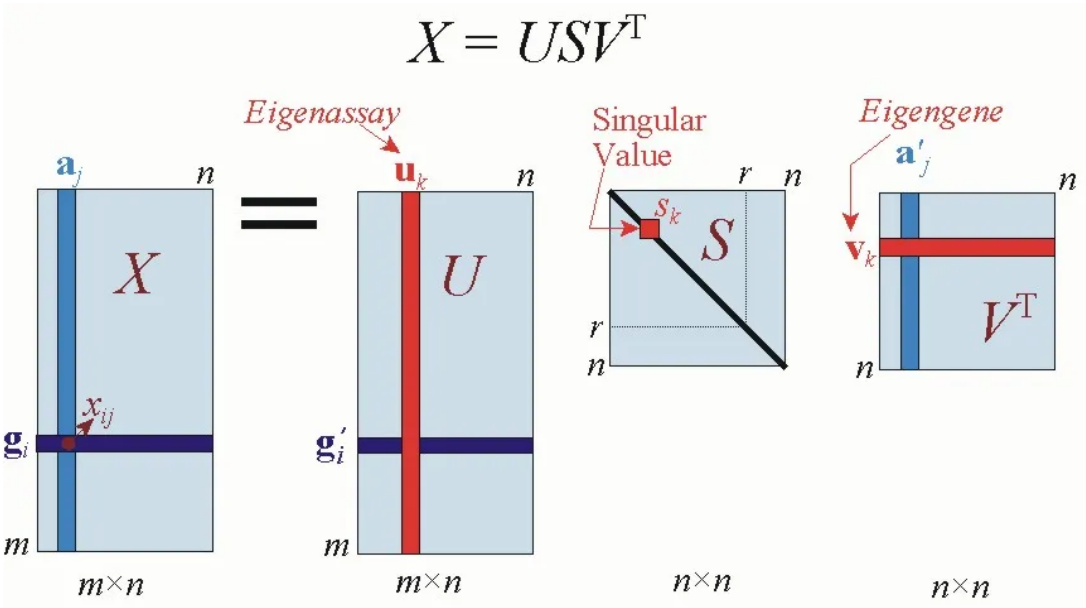
近年来，学术界的研究和工业界的应用结果表明，在处理高维数据方面，个性化推荐中使用**矩阵分解模型**要明显优于传统的**基于邻域的协同过滤**（又称基于内存或者记忆的协同过滤）方法，如**User-CF、Item-CF**等，这也使得矩阵分解成为目前个性化推荐研究领域中的主流模型。

国内外学者尝试采用各种方法来进行对**稀疏矩阵进行降维**，常用的**降维技术**有**SVD (Singular Value Decomposition)**、**主成分分析 (Principal Component Analysis, PCA)**，**矩阵分解**、**概率矩阵分解**、**非负矩阵分解**以及其改进模型等。

SVD (Singular Value Decomposition)

SVD是一种有效的数据降维技术，并有着自动抽取矩阵重要特征的能力。针对推荐系统在数据稀疏情况下推荐质量不高的问题，提出将奇异值分解（Singular Value Decomposition, SVD）技术和信任模型相融合的协同过滤推荐算法。首先运用SVD降维技术得到项目的隐式特

征空间。然后用改进的余弦相似度计算项目间相似度，根据**k近邻 (k-Nearest Neighbor, k-NN)** 算法得到第一阶段最近邻居集。接着引入项目信任因子，建立信任模型并融入相似度空间中，进行第二阶段k近邻选择，从而完成预测推荐。



SVD作为一种矩阵分解技术，通过生成一个低秩矩阵近似逼近原始矩阵。给定一个矩阵，则A的奇异值分解定义如式：


$$A = U \times S \times V^T$$

 未来学习空间

式中： $U \in R^{m \times m}$ ； $V \in R^{n \times n}$ ； $S \in R^{m \times n}$ 。矩阵U和V为**正交矩阵**，且矩阵的列分别为AAT和ATA的特征向量。 r 为矩阵R的秩； σ_r 为R的奇异值，值为AAT或ATA的特征值的平方根。因此，这三个矩阵的**有效维数分别是 $m \times r$ 、 $r \times r$ 、 $n \times r$** 。

SVD可以提供三个矩阵相乘对原始矩阵A进行最优的近似。通过将矩阵S保留第k个最大的奇异值进行简化得到新的对角矩阵，其中 $k < r$ 。然后通过删除U和V的相应列，简化后的矩阵U和V表示为 U_k 和 V_k ，

$$A_{red} = U_k \times S_k \times V_k^T$$

 未来学习空间

这是最接近的k位近似原矩阵的酉不变范数。


基于交替最小二乘的改进概率矩阵分解算法

概率矩阵分解算法是解决信息过载的有效手段，但在研究中经常面临高维稀疏性带来的推荐精度不高的问题。

交替最小二乘 (Alternating Least Squares, ALS) 随着迭代的进行误差会逐步降低直至收敛，ALS完全不能保证将会收敛至全局最优解，而且在实际应用中，**ALS对初始点选取较为敏感，不恰当的选择会导致数据振荡地收敛到局部最优解。**


首先按高斯分布初始化用户和项目的潜在因子向量U和V。然后固定V，将损失函数对V求偏导，并令导数等于0，得到新的用户潜在因子向量U

$$U \leftarrow (V^T V + \lambda I)^{-1} V^T R$$

 未来网络空间

其次固定U，将损失函数对U求偏导，并令导数等于0，

$$V \leftarrow (U^T U + \lambda I)^{-1} U^T R$$

 未来网络空间

式中： λ ——正则化系数，需要实验确定。最后便可利用得到的用户项目潜在因子空间U和V进行评分预测。

4

Implicit

Python - implicit库, 它是**基于隐式偏好**数据的python快速协同过滤推荐算法,它主要通过**矩阵分解**的方法来预测用户对物品的评分, 从而进行个性化的推荐。

<https://github.com/benfred/implicit/>

<https://implicit.readthedocs.io/en/latest/quickstart.html>

Implicit协同框架实现了多种流行的隐式推荐策略:

- ALS 交替最小二乘
- Bayesian Personalized Ranking
- Logistic Matrix Factorization
- Item-Item Nearest Neighbour Model

Implicit使用起来非常简单:

```
import implicit

# initialize a model
model = implicit.als.AlternatingLeastSquares(factors=50)

# train the model on a sparse matrix of item/user/confidence weights
model.fit(item_user_data)

# recommend items for a user
user_items = item_user_data.T.tocsr()
recommendations = model.recommend(userid, user_items)

# find related items
related = model.similar_items(itemid)
```



未来阅读空间

5

总结

工作项目需要, 学习了解中。

无特别总结, 除了:

“可惜,