基于 SVD 与模糊聚类的协同过滤推荐算法①

林建辉, 严宣辉, 黄 波

(福建师范大学 数学与计算机科学学院, 福州 350007)

摘 要: 协同过滤为个性化推荐解决信息过载问题提供了方案, 然而也存在着数据的稀疏性、可扩展性等影响推荐质量的关键问题. 我们提出了一种基于奇异值分解(SVD)与模糊聚类的协同过滤推荐算法, 通过引用物理学上狭义相对论中能量守恒的方法以保留总体特征值的数目, 较为准确地确定降维维度, 实现对原始数据的降维及其数据填充. 另外, 再运用模糊聚类的方法将相似用户进行聚类, 从而达到减少邻居用户搜索范围的目的. 在 MovieLens 与 2013 年百度电影推荐系统比赛等不同数据集上的实验结果表明, 该算法能够提高推荐质量.

关键词: 个性化推荐; 协同过滤; SVD; 模糊聚类

Collaborative Filtering Recommendation Algorithm Based on SVD and Fuzzy Clustering

LIN Jian-Hui, YAN Xuan-Hui, HUANG Bo

(School of Mathematics and Computer Science, Fujian Normal University, Fuzhou 350007, China)

Abstract: Collaborative filtering provides a solution for the personalized recommendation to solve the problem of information overload. But the problems of data sparsity and scalability are the serious factors affecting the recommendation quality. To solve these problems, we propose a collaborative filtering algorithm based on singular value decomposition and fuzzy clustering. We retain the number of the total characteristic value through the theory of energy conservation in the special relativity in physics, so as to determine the dimension of dimension reduction. In addition, by using the fuzzy clustering, we also reduce the search range of the neighbors. Compared with traditional collaborative filtering recommendation algorithm in the different data sets of MovieLens and 2013 Baidu movie recommendation system, the proposed algorithm performs better in the recommendation quality.

Key words: personalized recommendation; collaborative filtering; SVD; fuzzy clustering

1 引言

个性化推荐技术[1]为互联网出现的信息过载以及在大部分用户没有特别明确的需求的情况提供了最好的解决方案. 运用此项技术, 在大量的信息中用户可以更容易找的自己喜欢的物品, 并且在用户没有明确的需求时, 可以通过该技术对其推荐可能感兴趣的物品. 而作为传统的个性化推荐方法, 协同过滤[2-3]拥有算法简单、容易实现、推荐的新颖性等特点. 目前, 协同过滤大致可以分为: 基于邻域的方法(neighborhood-based)[4-7] 、基于模型的算法(model-based)[8]等. 而基于邻域的方法又可以分为基于用户(user-based)[5,6]和基于物品(item-based)[4]的协同过滤算法. 然而, 传统的协同过滤对于目前出现的数据稀疏性、可扩展性、冷启动以及实时性等问题并没有很好的解决方案, 而这些问题也正是影响推荐质量的关键因素.

为了解决数据的稀疏性、可扩展性等问题, 文献[4]提出基于物品评分预测的协同过滤算法, 其采用空值填补的方法来弥补缺失值, 对于解决数据稀疏性问题有一定的效果, 但这种利用填充值来计算用户间相似性本身就是个需要解决的问题. 文献[5]提出基于用户的协同过滤算法, 对于数据稀疏性问题, 作者设定一个迭代次数阈值, 循环执行基于用户的协同过滤算法从而填补缺失值, 最后采用传统的协同过滤算法得

① 收稿时间:2016-03-03;收到修改稿时间:2016-04-19 [doi:10.15888/j.cnki.csa.005474]

到预测评分. 文献[7]使用 K-means 聚类算法聚类用户-项目评分矩阵, 选择目标用户所在聚类的用户作为最近邻集合, 通过减小最近邻搜寻空间的方式提高协同过滤的可扩展性. 文献[9]使用了一种改进的增量奇异值矩阵分解的方式, 通过随机梯度下降法来对用户—物品评分矩阵进行分解以应对数据稀疏性问题, 并结合用户档案信息来处理新用户问题, 但是作者并未能够处理好矩阵的降维问题及用户档案信息涉及隐私的问题.

本文针对数据稀疏性、可扩展性等问题提出了基于奇异值分解[9,10]与模糊聚类[11,12]的协同过滤推荐算法(collaborative filtering algorithm based on singular value decomposition and fuzzy clustering, SVDFC-CF). 首次引用了物理学上狭义相对论中能量守恒的方法, 确保了 SVD 对原始数据的降维既不会因过度降维而导致过多的信息损失但又能达到最好的推荐效果; 结合模糊聚类的方法将用户按相似度实行模糊归类, 可以大大缩小邻居的搜索范围; 而改进 Pearson 相关系数度量方法更加注重用户间的共同操作, 更能恰当地衡量了用户间的相似性. 这种基于奇异值分解与模糊聚类的协同过滤的方法对于解决数据稀疏性、可扩展性等问题有较好的效果. 实验表明, 本文算法能够提高推荐质量.

2 传统基于用户的协同过滤

2.1 数据初始化

将用户集 $U(u_1, u_2, \ldots, u_m)$及用户的评分项目集 $I(i_1, i_2, \ldots, i_n)$构造成为一个用户-项目矩阵 $R_{m\times n}$, $u_1, u_2, \ldots, u_m$ 表示有 m 个用户, $i_1, i_2, \ldots, i_n$ 表示有 n 个项目, $R_{i,j}$ 表示用户 u_i 对项目 i_j 的评分值. 得到用户-项目评分矩阵如下所示.

$$R_{m\times n}=\begin{bmatrix} R_{1,1} & \cdots & R_{1,j} & \cdots & R_{1,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ R_{i,1} & \cdots & R_{i,j} & \cdots & R_{i,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ R_{m,1} & \cdots & R_{m,j} & \cdots & R_{m,n} \end{bmatrix}$$

2.2 最近邻居的形成

基于用户的协同过滤在给目标用户推荐项目之前需先完成: 寻找目标用户的邻居, 根据对目标项目的操作有无来选取邻居; 计算邻居用户与目标用户的相似度, 将相似度最大的前 N 个用户作为目标用户的最终邻居. 而传统的基于用户的协同过滤其相似性度量方法[5]有三种: A、余弦相似性; B、相关相似性(也称 pearson 相关系数); C、修正的余弦相似性. 实验研究表明[13]相关相似性在相似性度量更为准确. 用户 u 和 v 之间的相似性用 $sim(u, v)$来表示, 其计算方法如式(1):

$$sim(u,v)=\frac{\sum_{j\in Iuv}\left(R_{u,j}-\overline{R_u}\right)\left(R_{v,j}-\overline{R_v}\right)}{\sqrt{\sum_{j\in Iuv}\left(R_{u,j}-\overline{R_u}\right)^2}\sqrt{\sum_{j\in Iuv}\left(R_{v,j}-\overline{R_v}\right)^2}} \quad (1)$$

其中, $R_{u,j}$ 表示用户 u 对项目 j 的评分值, $\overline{R_u}$ 表示用户 u 对其有过评分的项目的平均评分值, $R_{v,j}$ 表示用户 v 对项目 j 的评分值, $\overline{R_v}$ 表示用户 v 对其有过评分的项目的平均评分值, $I_{u,v}$ 表示用户 u 和 v 有共同评分的项目集.

2.3 评分预测

选择用户间相似度最大的前 N 个用户作为目标用户的最终邻居, 对目标用户未评分项目进行预测, 根据邻居用户的推荐可以预测得出评分值, 其计算公式如式(2):

$$p_{u,j}=\overline{R_u}+\frac{\sum_{v\in N}sim(u,v)\left(R_{v,j}-\overline{R_v}\right)}{\sum_{v\in N}sim(u,v)} \quad (2)$$

式中, $P_{u,j}$ 表示目标用户 u 对项目 j 的预测评分值, N 为目标用户 u 最终的邻居用户集.

3 基于SVD与模糊聚类的协同过滤

本节对本文提出的基于 SVD 与模糊聚类的协同过滤算法进行详细阐述, 算法的基本思路是: 先通过 SVD 对原始用户-项目矩阵 $R_{m\times n}$ 进行预处理, 构造得到用户相关矩阵; 其次, 对预处理的数据利用模糊聚类算法将用户按相似度实行归类; 然后, 采用改进 Pearson 相关系数方法计算用户与目标用户之间的相似性, 取相似性最大的前 N 个用户最为目标用户的邻居; 最后, 通过预测目标用户对未评分项目的预测, 将评分最高的前 N 个项目作为结果完成 *top-N* 推荐. 以下是对本文算法研究讨论的叙述过程:

3.1 构造用户矩阵

SVD 是一种矩阵分解技术, 其本质将一个矩阵分解为三个相关矩阵. 具体的表达式如式(3):

$$R_{m\times n}=U_{m\times m}\times S_{m\times n}\times I_{n\times n} \quad (3)$$

其中 U 的每一行代表一个用户的特性, I 的每一列代表

一个项目的特性, 而 S 代表对应的 U 和 I 之间关联程度, 是一个 $m\times n$ 的对角矩特征值矩阵.

$$U'_{m\times k}=R_{m\times n}\times I_{n\times k}\times S^{-1}{}_{k\times k} \tag{4}$$

$$S'=P\times\sum_{i\in m,j\in n}S_{ij}{}^2 \tag{5}$$

式(4)目的是将初始的数据矩阵映射到反映用户的相互关系当中去. 其中, $U'_{m\times k}$ 为构造的用户矩阵, 其每一行代表一个用户, $R_{m\times n}$ 为初始数据矩阵, $I_{n\times k}$ 为降维后的项目相关矩阵, $S^{-1}{}_{k\times k}$ 为用户与项目之间相关联的特征逆矩阵, k 为降维保留特征维数. 我们这样做的目的是: 一, 将高维的数据矩阵降维到较低维. 二, 简化了用户-项目之间的关系, 处理起来比初始数据矩阵更为方便. 三, 计算用户相似度时, 根据用户的特征矩阵可以更准确地计算用户之间的相似性.

式(5)的目的是为了得到保留的总体特征值能量, 由式(3)得到的特征值矩阵 $\boldsymbol{S}$, 是一个大小为 $m\times n$ 阶的对角矩阵, 并且其值由大到小在主对角上依次排列. 此时我们借鉴能量守恒公式(5), 式中 $\sum_{i\in m,j\in n}S_{ij}{}^2$ 表示的是要计算矩阵 $\boldsymbol{S}$ 的总特征值能量, 而 P 是保留总体特征值的百分比, 其值大小的确定在文中第 4 节实验部分有作描述, S' 为保留的总特征值能量, 而后我们根据此值的大小, 类似计算 $\sum_{i\in m,j\in n}S_{ij}{}^2$ 总体特征值能量的办法, 反过来计算 i, j 的值, 我们可以得到最终的降维 $K=i$. 式(5)是基于这个假设, 如果特征值的大小表示了原始数据矩阵的性质、结构及其原始信息, 那么在特征值总体的选取方面就应该遵循选择出来的特征值要能最大化地反映的原始信息. 也就是说, 如果选取的特征值总体过大(即为奇异矩阵维数过大), 那么就没有达到降维的目的. 如果选取的特征值总体太小(即为奇异矩阵维数过小), 导致失去原始信息过多. 因而根据物理学上狭义相对论中能量守恒定义, 奇异值的选取符合能量守恒的规则, 选择出来的奇异值的能量要能最大限度反映原始信息.

例如本实验用到的数据, 文中所用数据集中包含 943 个用户对 1682 部电影的 10000 条评分记录, 那么初始数据就为一个 943×1682 的矩阵, 而经过降维处理后得到的用户数据仅为一个 943×11 的矩阵, 这样需要处理的数据量就远远地小于了 943×1682, 并且在降维后的数据也尽可能地保留了数据原始信息.

3.2 模糊聚类分析

3.2.1 数据标准化处理

描述用户特征的方法纷繁复杂, 为了便于比较分析, 在计算过程首先有必要对数据矩阵进行标准化处理. 本文运用平移-极差变换进行标准化, 通过标准化处理后的用户矩阵为 $U''_{m\times k}$.

$$U''_{i,j}=\frac{U'_{i,j}-\min\{U'_{i,j}\,|\,1\le i\le m\}}{\max\{U'_{i,j}\,|\,1\le i\le m\}-\min\{U'_{i,j}\,|\,1\le i\le m\}} \tag{6}$$

式中, i=1, 2, …, m . j=1, 2, …, k.

3.2.2 建立模糊相似矩阵

针对上述的标准矩阵, 计算各分类对象的相似程度, 以此建立模糊相似矩阵 $U''_{m\times k}$, 这个过程又称为标定, 计算标定的方法大致可分为三类: (1)相似系数法; (2)距离法; (3)主观评价法. 综合考虑本论文中所需处理的数据, 选择距离法比较适合, 标定的计算公式为:

$$U^*{}_{i,j}=1-c\cdot d\left(U''_i,U''_j\right) \tag{7}$$

$$d\left(U''_i,U''_j\right)=\sqrt{\sum_{l}^{k}\left(U''_{i,\,l}-U''_{j,\,l}\right)^2} \tag{8}$$

$$c=\frac{1}{1+d\left(U''_i,\,U''_j\right)} \tag{9}$$

上述各式中, U''_i、U''_j 表示为用户 u_i 与 u_j, $U''_{i,l}$、$U''_{j,l}$ 表示为用户 u_i 与 u_j 的第 l 个相关值. i, j=1, 2, …, m. l=1, 2, …, k. d 为用户间的欧式距离, c 为根据距离所适当选取的参数.

3.2.3 构造模糊等价矩阵

利用传递闭包法将上述所得模糊相似矩阵转化为模糊等价矩阵, 传递闭包法即为对模糊相似矩阵 U^*, 求 U^{*2}, U^{*4}, $...U^{*k}$, 当 $U^{*k}\circ U^{*k}=U^{*k}$ 时, U^{*k} 即为所求的模糊等价矩阵. 其中运算符"∘"是模糊乘积, 相乘时取两个乘积值中的最小值, 相加时取相加各值中的最大值.

例如: 将一个相似矩阵 $U^*=\begin{bmatrix}1 & 0.1 & 0.2\\ 0.1 & 1 & 0.3\\ 0.2 & 0.3 & 1\end{bmatrix}$ 改造成为一个模糊等价矩阵. 按照上面所述进行处理:

$$U^*\circ U^*=\begin{bmatrix}1 & 0.1 & 0.2\\ 0.1 & 1 & 0.3\\ 0.2 & 0.3 & 1\end{bmatrix}\circ\begin{bmatrix}1 & 0.1 & 0.2\\ 0.1 & 1 & 0.3\\ 0.2 & 0.3 & 1\end{bmatrix}$$

$$=\begin{bmatrix}1 & 0.2 & 0.2\\ 0.2 & 1 & 0.3\\ 0.2 & 0.3 & 1\end{bmatrix}=U^{*^2}$$

$$U^{*^2}\circ U^{*^2}=\begin{bmatrix}1 & 0.2 & 0.2\\ 0.2 & 1 & 0.3\\ 0.2 & 0.3 & 1\end{bmatrix}\circ\begin{bmatrix}1 & 0.2 & 0.2\\ 0.2 & 1 & 0.3\\ 0.2 & 0.3 & 1\end{bmatrix}$$

$$=\begin{bmatrix}1 & 0.2 & 0.2\\ 0.2 & 1 & 0.3\\ 0.2 & 0.3 & 1\end{bmatrix}=U^{*4}=U^{*^2}$$

此时U^{*^2}就是我们所要构造的模糊等价矩阵.

3.2.4 模糊聚类

模糊聚类就是根据模糊等价矩阵将对象进行分类的方法. 这种聚类方法是通过一定的阈值来确定对象的相似类别, 使得属于同一类别的用户之间的相似性比较高. 即对于不同的聚类阈值水平$\lambda\in[0,1]$, 可以得到不同的分类结果, 将大于等于 λ 的值归为一类, 最终得到聚类结果. 根据聚类结果得到不同的用户分类, 在搜索邻居用户时, 只需要在同类中计算与目标用户的相似性大小就可以确定邻居用户, 从而减少了邻居的搜索范围.

例如, 对于一个如上述构造方法得到的模糊等价矩阵$\begin{bmatrix}1 & 0.6 & 0.5 & 0.7 & 0.5\\ 0.6 & 1 & 0.3 & 0.5 & 0.6\\ 0.5 & 0.3 & 1 & 0 & 0.1\\ 0.7 & 0.5 & 0 & 1 & 0.6\\ 0.5 & 0.6 & 0.1 & 0.6 & 1\end{bmatrix}$, 当 λ=0.5 时, 得到聚类矩阵$\begin{bmatrix}1 & 1 & 1 & 1 & 1\\ 1 & 1 & 0 & 1 & 1\\ 1 & 0 & 1 & 0 & 0\\ 1 & 1 & 0 & 1 & 1\\ 1 & 1 & 1 & 1 & 1\end{bmatrix}$, 由此便可以将用户 1 和 5 分为一类, 而用户 2 和 4 分为一类, 另外用户 3 单独归为一类.

3.3 改进相似性度量

传统协同过滤在用户的相似性度量方面有考虑用户项目的共同评分值, 实际上, 也只有对用户项目上有过共同评分操作才更能体现出用户之间的相似性. 举个例子: 如果用户 A 与 B 之间没有对任何项目的有过共同评分, 那么仅仅就因为用户 A 买了与 B 毫无相关的项目就要计算出 A 与 B 有相似度? 这显然不够合理. 因此, 我们做出了如下改进:

$$sim(u,v)=\frac{\sum_{u,v,j\in Iuv}\left(R_{u,j}-\overline{R_u}\right)\left(R_{v,j}-\overline{R_v}\right)}{\sqrt{\sum_{u,j\in Iuv}\left(R_{u,j}-\overline{R_u}\right)^2}\sqrt{\sum_{v,j\in Iuv}\left(R_{v,j}-\overline{R_v}\right)^2}} \quad (10)$$

上式更加重视用户之间的共同评分中, 式中, $\overline{R_u}$ 为用户 u 和 v 共同评分项目用户 u 对项目评分的平均值, $\overline{R_v}$ 为用户 u 和 v 共同评分项目用户 v 对项目评分的平均值.

3.4 算法描述

综合第二节对传统基于用户的协同过滤与第三节对 SVD 与模糊聚类的展开讨论, 可以得到以下对本文算法的简要描述:

Input: 用户-项目的评分矩阵 $R_{m\times n}$, 目标用户 u_i 及其未评分的目标项目 i_j, 聚类阈值 λ.

Output: 目标用户 u_i 对目标项目 i_j 的预测评分值, *Top-N* 推荐.

Step1: 对用户-项目矩阵根据(3)、(4)、(5)式计算得到用户矩阵.

Step2: 对用户矩阵进行模糊聚类, 根据聚类结果得到相似用户的聚类.

Step3: 根据(10)式计算用户与目标用户之间的相似性, 得到最终的 N 个最近邻居用户.

Step4: 结合得到的邻居用户与评分预测公式(2), 对目标用户的未评分目标项目进行预测评分.

Step5: 从得到的预测结果中选取评分最高的 N 个项目作为推荐结果, 产生 *Top-N* 推荐.

对于一个包含 m 个用户, n 个项目实验数据集($m<n$). 在 Step1 中的的奇异值分解模块, 总的需要 $m*n*m$ 次乘法运算, 其时间复杂度为 $O(n^3)$; 而在 Step2 的模糊聚类中, 对用户矩阵进行标准化处理的时间复杂度为 $O(m\cdot k)$, 计算模糊相似矩阵与模糊等价矩阵的时间复杂度都为 $O(m^2)$; Step3 中计算用户与目标用户之间的相似性的时间复杂度为 $O(m\cdot q)$, q 取决于聚类的用户数且 $q<<n$; 最后在计算评分预测矩阵的时间复杂度为 $O(m\cdot n)$. 综上所述, 算法的时间复杂度为 $O(n^3)$.

4 实验结果及分析

4.1 实验数据集

本文用的实验数据集包括: (1)美国明尼苏达大学 GroupLens 研究项目组所收集到的 MovieLens 数据集

(http://MovieLens.umn.edu), 根据用户对电影的评分向其提供推荐列表, 文中所用数据集中包含 943 个用户对 1682 部电影的 100000 条评分记录. 其中, 每个用户至少评价过 20 部电影. (2)百度在 2013 年举办的百度电影推荐系统比赛所用的数据集(http://openresearch.baidu.com/ark.jspx), 数据集包含了 15 万用户对 15000 部电影约一百万条的评分记录, 实验随机选择了 943 位用户对 1682 部电影共 62507 条的评分记录. 两个数据集的评分值均为 1～5 之间的整数, 评分越高代表用户越喜欢该电影. 实验将数据集多次随机分为训练集和测试集, 其数据比例为 4:1, 最终结果取平均值.

4.2 评测指标

作为推荐系统最常用的评测指标——平均绝对偏差 (Mean Absolute Error, MAE)已经被绝大多数人认可. 其计算预测用户的评分值与实际用户的评分值之间的偏差, MAE 的值越小, 说明推荐质量越好. 设预测用户的评分值集合为$\{p_1, p_2, ..., p_N\}$, 而实际用户的评分值集合为$\{q_1, q_2, ..., q_N\}$, 测试数为 T, 则 MAE 的计算公式如式(11):

$$MAE = \frac{\sum_{i=1}^{T}|p_i - q_i|}{T} \tag{11}$$

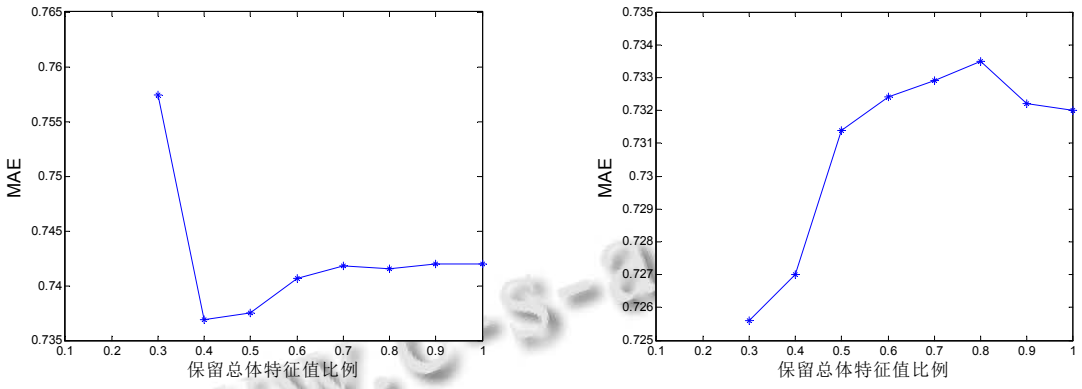
另外, 我们还用到准确度(Precision)指标作为衡量本文算法测评指标, Precision 指标是通过计算预测评分与实际评分相等的数量占整个测试集的比率来衡量推荐的准确度, Precision 指标值越大, 说明推荐准确性越好. 其表达式如(12)(13)所示.

$$N_i = \begin{cases} 1 & p_i=q_i \\ 0 & p_i \neq q_i \end{cases} \tag{12}$$

$$\Pr ecision = \frac{\sum_{i=1}^{T} N_i}{T} \tag{13}$$

4.3 实验结果及分析

首先需要确定降维的维数, 因为这会直接影响到实验结果. 实验中, 取横坐标代表保留总体特征值的比例, 其取值从 10%开始到 100%, 每次增加 10%, 终坐标代表 MAE, 观察 MAE 的变化, 最终确定保留比例. 实验结果如图 1 所示.



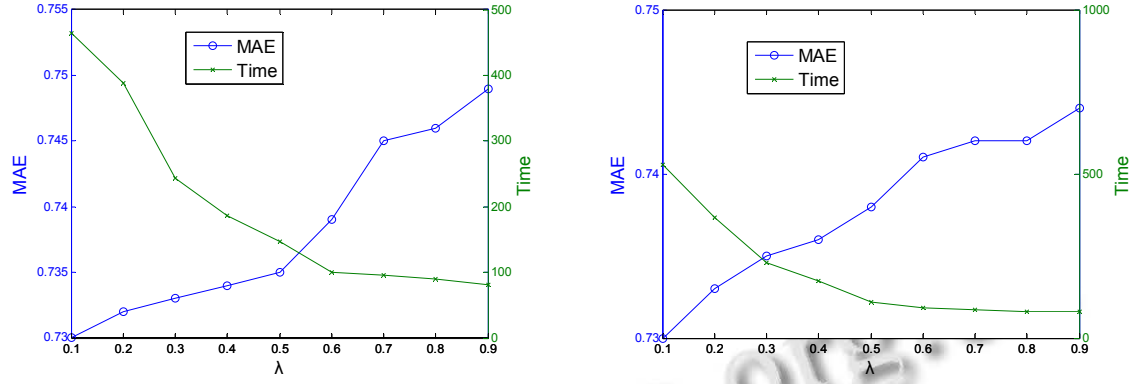
(a)MovieLens 数据集下保留总体特征值比例与 MAE 的关系　　(b)百度电影数据集下保留总体特征值比例与 MAE 的关系

图 1　不同数据集下保留总体特征值的比例对 MAE 的影响

由于在保留总体特征值的比例为 0.1、0.2 时的值为无穷大, 图 1(a)与(b)中无法描绘. 通过对比保留总体特征值比例对 MAE 的影响的实验结果, 在不同数据集下我们可以观察到 MAE 大致随着保留总体特征值比例增大先减小后逐渐增大, 从(a)可以观察到在 MovieLens 数据集下保留总体特征值的比例为 40%时 MAE 有最小值, 此时的推荐效果达到最佳, 因此在 MovieLens 数据集下的验证本文算法实验中将以保留总体特征值的 40%为基础. 通过(b)可知在百度电影数据集下保留总体特征值的比例为 30%时, MAE 取得最小值, 因此后续在百度电影据集下的验证本文算法实验将以保留总体特征值的 30%为前提. 两个数据集下的保留特征值比例不同, 表明本文算法能够根据不同数据集的特点准确选取相应的降维维数.

在模糊聚类的实验中, 我们选取横坐标代表聚类阈值 λ, 其取值从 0.1 开始到 0.9, 左侧纵坐标代表

MAE, 右侧纵坐标代表算法运行时间 Time(s), 综合观察 MAE 及 Time 的变化, 确定恰当的聚类阈值, 实验结果如图 2 所示.



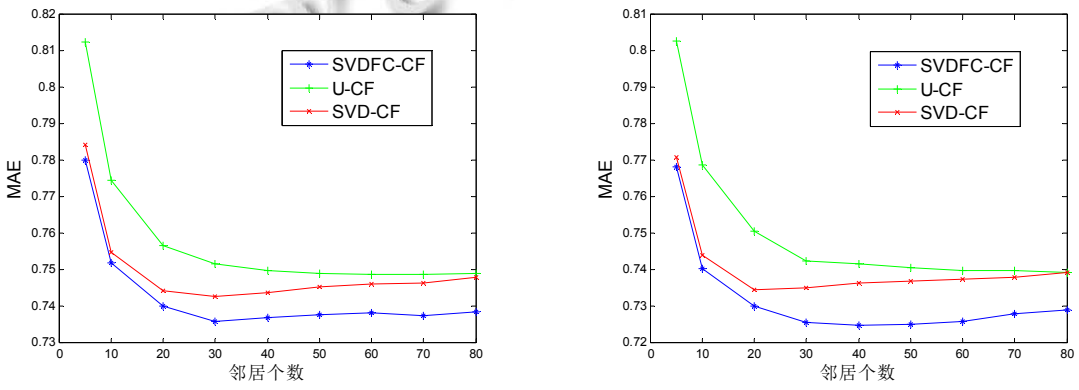
(a)MovieLens 数据集下阈值 λ 对 MAE Time 的关系

(b)百度电影数据集下聚类阈值 λ 对 MAE 与与 与 Time 的关系

图 2 不同数据集下聚类阈值 λ 对 MAE 与运行时间的影响

从实验结果观察可知, 在不同数据集下, 随着聚类阈值选取逐渐增大, MAE 都表现出增长的趋势, 而算法的运行时间在都呈现出减小的状态. 在实际中如果更注重 MAE, 那么就要牺牲算法运行时间为代价, 如果尽量想减少算法的运行时间, 那么就必须降低对 MAE 的要求. 权衡 MAE 及 Time 的实验结果, 在 MovieLens 数据集选取聚类阈值 λ 为 0.6 作为后续的实验基础, 在百度电影数据集下选取聚类阈值 λ 为 0.5 作为后续实验的基础. 不同的数据集选取不同聚类阈值时在考虑 MAE 及 Time 这两个指标的同时, 还应该尽量考虑数据集本身的特点等因素.

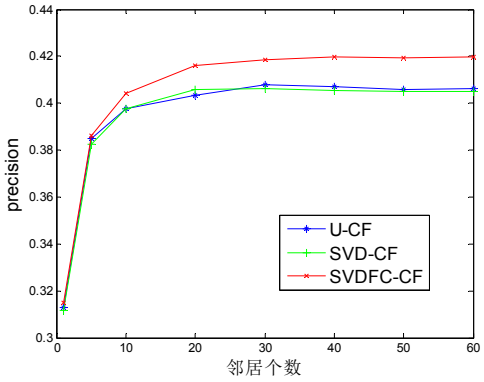
为了验证本文提出的基于 SVD 与模糊聚类的协同过滤推荐算法(SVDFC-CF)的推荐效果, 本文将与传统的基于用户的协同过滤算法(U-CF)以及文献[10]提出的基于 SVD 的协同过滤推荐算法(SVD-CF)在推荐质量与推荐准确度两项指标上进行对比. U_CF 首先是计算用户之间的相似性, 得到用户之间的相似矩阵; 其次是寻找目标用户的最近邻居, 将相似性最高的前 N 个用户作为目标用户的最近邻居; 最后是根据最近邻居的评分, 以此来实现对目标用户为评分的预测. 而 SVD_CF 的方法是对用户-项目的评分矩阵进行奇异值分解, 然后将分解得到的用户相关矩阵与特征值矩阵开根号的乘积, 最后在此基础上做协同过滤推荐. 实验结果如图 3、4 所示



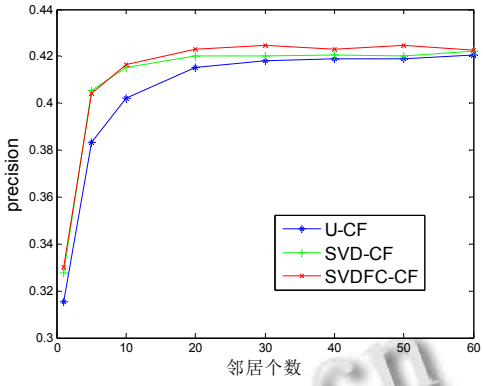
(a)Movielens 数据集下邻居用户个数与 MAE 的关系

(b)百度电影数据集下邻居用户个数与 MAE 的关系

图 3 不同数据集下各推荐算法推荐质量的对比



(c)Movielens 数据集下邻居用户个数与 Precision 的关系



(d)百度电影数据集下邻居用户个数与 Precision 的关系

图 4 不同数据集下各推荐算法推荐准确度的对比

从图 3 的(a)、(b)中我们可以发现, 无论是在 MovieLens 数据集还是在百度电影数据集下, 随着最近邻居用户个数的不断增加, 上述三种算法中的 MAE 都呈现出不同程度的下降趋势. 当邻居用户达到一定的值时, 算法 SVDFC-CF 与 U-CF 的 MAE 降低与增加逐渐开始趋于缓慢, 而 SVD-CF 的 MAE 略微上升. 我们通过 SVDFC-CF 与 U-CF 和 SVD-CF 在各邻居用户相同时的情况下对 MAE 值的对比中可以发现, 当邻居用户较少时, SVDFC-CF 的 MAE 比 SVD-CF 稍好, 且两者的 MAE 都比 U-CF 的好. 当邻居用户超过 20 个时, 本文提出的 SVDFC-CF 比其它两个算法有更大优势, 表现出更好更稳定的推荐质量. 而从图 4 的(c)、(d)中我们可以观察到, 无论是在 MovieLens 数据集还是在百度电影数据集下, 随着最近邻居用户个数的不断增加, 上述三种算法中的 Precision 指标值都呈现出不同上升趋势, 达到一定的值时开始趋于平缓. 我们通过 SVDFC-CF 与 U-CF 和 SVD-CF 在各邻居用户相同时的情况下对 Precision 值的对比中可以发现, 当邻居用户较少时, 三种算法的好坏差异不是很明显. 随着最近邻居用户个数的不断增加本文算法与 U-CF 和 SVD-CF 在 Precision 指标值的对比中有比较明显的优势, 表现出更好的推荐准确度.

5 总结

本文所提出的基于 SVD 与模糊聚类的协同过滤算法(SVDFC-CF)在一定程度上有自己的优势, 首先, SVD 技术可对矩阵进行降维与填充缺失值, 而模糊聚类在对缩减寻找邻居用户的时间上有比较大的帮助, 最后根据改进的相似性度量方法对上述所得用户矩阵进行度量也更准确反映了用户间的相似性. 实验结果表明提高了推荐的质量.

参考文献

1 Resnick P, Iacvou N, Suchak M. GroupLens: An open architecture for collaborative filtering of Netnews. Proc of the 1994 ACM Conf Computer Supported Cooperative Work. New York. ACM Press. 1994. 175–186.

2 Herlockerjl, Konstanja. Evaluating collaborative filtering recommender systems. ACM Trans. on Information Systems, 2004, 22(1): 5–53.

3 李聪.电子商务推荐系统中协同过滤瓶颈问题研究[博士学位论文].合肥:合肥工业大学,2009.

4 邓爱林,朱扬勇,施伯乐.基于项目评分预测的协同过滤推荐算法.软件学报,2003,14(9):1621−1628.

5 Yue S, Larson M, Hanjalic A. Exploiting user similarity based on rated-item pools for improved user-based collaborative. Proc. of the 3rd ACM Conf on Recommender System. New York. ACM Press. 2009. 125–132.

6 Zhang F, Chang HY. Employing BP neural networks to alleviate the sparsity issue in collaborative filtering recommendation algorithms. Journal of Computer Research and Development, 2006, 43(4): 667–672.

7 Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms. Proc. of the 10th International Conference on World Wide Web. New

York. ACM Press. 2001. 285–295.

8 张光卫,李德毅,李鹏.康建初,陈桂生.基于云模型的协同过滤推荐算法.软件学报,2007,18(10): 2403–2411.

9 杨阳,向阳,熊磊.基于矩阵分解与用户近邻模型的协同过滤推荐算法.计算机应用,2012,32(2):395–398.

10 Sarwar BM, Karypis G, Konstan JA, Riedl JT. Application of dimensionality reduction in recommender system--a case study. Proc. of the ACM WebKDD 2000 Web Mining for E-commerce Workshop. Boston USA. 2000. 82–90.

11 高新波.模糊聚类分析及其应用.西安:西安电子科技大学出版社,2004.

12 李华,张宇,孙俊华.基于用户模糊聚类的协同过滤推荐研究.计算机科学,2012,39(12).

13 Breese JS, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering. Proc. of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98). Sar Francisco. ACM Press. 1998. 43–52.