基于协同过滤技术推荐系统的探究

江 水
(中国电子科技集团公司第三十二研究所,上海 201808)

摘 要: 基于数据挖掘技术的推荐系统在互联网和大数据时代无处不在,它能够根据用户的兴趣和行为特征,为其推荐感兴趣的项目,具有深远的商业价值。协同过滤技术综合考虑用户的历史特征和用户之间的相似关系,为用户进行预测推荐,已经广泛应用于现有的推荐系统中。文中阐述了协同过滤技术的概念内容、前提假设、数据采集与过滤模式、数据结构、评价标准和算法分类,并着重分析对比了基于存储、基于模型和关联规则等三种推荐系统算法的不同实现方式及其优缺点。在此基础上,编程实现了向量相似性算法、个性诊断算法、奇异值分解算法和关联算法等典型常用算法,并在 Movielens 数据集和稀疏成绩单数据库上进行了验证与评估,实验结果可以为设计人员提供指导依据。

关键词: 推荐系统; 数据挖掘; 协同过滤; 大数据; 个性诊断
中图分类号: TP302 **文献标识码:** A **文章编号:** 1673-629X(2021)11-0001-07
doi: 10.3969/j.issn.1673-629X.2021.11.001

Research on Recommender Systems Based on Collaborative Filtering

JIANG Shui
(The 32nd Research Institute of China Electronics Technology Group Corporation, Shanghai 201808, China)

Abstract: Recommender systems based on data mining technology have been widely used in the era of Internet and big data. They can recommend relevant items to users according to their interests and behaviors, which contributes a deep commercial value. Collaborative filtering belongs to the most widely used recommender systems, which can implement prediction and recommendation using the history pattern of users and their similarity. We systematically introduce the concept, assumption, data collection and filtering mode, data structure, evaluation criteria and classification of collaborative filtering, analyze and compare different implementation and advantage and disadvantage of memory-based, model-based and association rules. On the basis, we have programmed several typical algorithms, such as vector similarity algorithm, personality diagnosis algorithm, singular value decomposition algorithm and correlation analysis algorithm, which are verified and evaluated on Movielens dataset and sparse report card database. The experimental results can provide instructions for system designers.

Key words: recommender system; data mining; collaborative filtering; big data; personality diagnosis

1 概 述

网上商城通常对产品进行分类,并提供符合用户需求的产品选项供用户点选查看,然而,用户浏览产品的随机性很大,他们可能会选择很多属于不同类别或风格的产品,使得网上商城无论提供多少预设选项都无法满足用户的需求。因此,商业站点普遍通过展示特价产品、新到产品或者商家推荐产品去试图弥补这方面的不足。在众多的营销手段中,从最畅销产品和用户关注产品的角度进行推荐往往最有效。作为其中的一种吸引人的技术实现方法,基于数据挖掘技术的推荐系统(Recommender Systems)已经广泛应用于各类网上商城,包括其他行业需要向用户推荐、推销产品的在线系统中[1]。

推荐系统主要依据用户自身的特征和过往消费情况向其推荐产品。有些推荐系统还会结合使用与产品信息相关的问题进行推荐,有些则仅仅依赖于用户自身的特征与相似用户的总体特征,后者一般被称为“协同过滤”(Collaborative Filtering)系统[2-3],不要求用户在搜索的过程中显示输入过滤条件,而是通过分析当前用户的过往消费或浏览行为和与其相似的用户的总体特征来给出推荐意见。现有的推荐系统算法大致可以分为基于存储(Memory-based)的推荐算法、基于模型(Model-based)的推荐算法和关联规则(Association Rules)算法[1-2,4]。其中,基于模型的推荐算法

收稿日期: 2020-12-09 **修回日期:** 2021-04-13
基金项目: 国家自然科学基金资助项目(61402497)
作者简介: 江 水(1970-),男,工程师,研究方向为分布式计算。

又包括基于项目(Item-based)的协同过滤算法、个性诊断(Personality Diagnosis)和奇异值分解(Singular Value Decomposition,SVD)等。

文中阐述了协同过滤技术的概念内容、数据结构、评价标准和分类,着重分析对比了协同过滤技术的不同实现方式及其优缺点,为设计人员提供指导依据。

2 协同过滤算法的基本概念

2.1 协同过滤的根本假设

协同过滤技术综合考虑用户的历史特征和用户之间的相似关系,为用户进行预测推荐[2,5],其基本假设是:在过去有相似偏好的用户,在未来也极有可能有相似的偏好。基于这个假设,可以根据用户的历史信息来预测将来该用户可能感兴趣的商品。需要注意的是,该假设并不是在任何时候都成立但对于推荐系统是足够的,在最差情况下,与盲目的猜测相比,应用协同过滤算法设计的推荐系统能够给出较优的结果。

2.2 显式与隐式的数据采集

为了得到用户的历史偏好信息,需要进行数据采集。数据采集的过程可以分成两种,一种是显式的采集方式:要求用户在现有评分体系下明确对某一对象进行打分(比如对一首流行歌曲从一颗星到五颗星的评分);另一种是通过记录用户在站点上的点击、浏览和停留时间长短等行为,隐式地进行数据采集。通过显示方法采集到的数据,用户给出的等级排名或者评分信息,能够被直接翻译为用户的偏好信息,使系统容易推断用户将来的选择或喜好,但是其过于依赖用户操作,在实际情况下,用户也许并不想去花费时间进行评分,导致采集到的数据不全或者不具有代表性。相比之下,隐式的数据采集不需要用户的干预,依托相应的算法,系统能够容易获得大量用户操作信息,然而系统无法像使用显式的评分信息那样,直接利用这些隐式信息进行预测推荐,需要使用精确的算法将用户行为转换成其偏好。由于难以辨别用户的行为能多大程度上反映用户的实际偏好,使得使用隐式信息进行预测和推荐的难度加大。当然,这两种数据采集方法不是互斥的,通过融合能够得到优化的整体结果,其中一个可行的方案是,对于用户明确给出评级或者评分的情况,系统使用其进行预测;对于评分缺失的情况,系统抽取隐式的用户行为进行推荐预测。

2.3 主动过滤和被动过滤

数据采集完成后需要使用协同过滤方法进行预测,协同过滤方法大致可分为被动和主动两种方式:被动过滤仅使用数据的聚合(Data Aggregate)关系进行预测(比如某一商品的平均投票);更加先进的方式是使用用户历史特征进行预测的主动过滤,比如使用与当前用户相似的用户的历史信息对当前用户进行预测。两种方法的根本区别在于,推荐系统是否针对特定的用户:对于被动过滤系统中的某一个商品,每位用户会得到相同的预测结果,系统没有针对特定的用户;而对于主动过滤,系统需要考虑每位用户的不同历史特征进行推荐,Amazon.com 就是其中之一。尽管被动过滤对于很多的应用来说也非常有用,对于以向用户推荐、推销商品为目的的大多数网上商城来说,往往只能使用主动过滤的方法,大家通常说的协同过滤,指的也都是主动过滤。

2.4 以用户为中心和以项目为中心

所有的推荐系统需要决定是要分析用户还是商品之间的特征进行预测。以用户为中心的系统寻找用户之间的相似点,然后使用相似用户的偏好对当前用户的选择进行预测;以商品为中心的系统寻找商品之间的关系,然后根据这些关系和一个用户的偏好预测其可能选择的其他商品。尽管可以将这两种方式组合实现,通常情况下,推荐系统会侧重于其中一种。

2.5 基于存储和基于模型的算法

推荐系统使用的预测算法也可以分为基于存储(Memory-based)和模型(Model-based)两类。基于存储的算法持续使用所有的数据信息进行预测;基于模型的算法使用数据进行学习、训练,利用得到的模型进行预测。这意味着基于存储的算法通常需要将所有的数据存储起来,而基于模型的算法在模型建立之后可以使用少于原始数据的量进行快速预测。

2.6 推荐系统评价标准

推荐系统或协同过滤算法的好坏主要有以下几点评价标准[1-2,5]:

(1)预测质量:能够做出较为准确或者符合实际用户需求的预测或推荐。

(2)预测速度:推荐系统广泛应用于商业站点中,其算法需要高效,能够快速做出预测。

(3)可扩展性:推荐系统的预测算法不会因为数据库规模的扩大而降低预测质量和速度。

(4)可维护性:在改变排名、评分或者量化标准时,已有算法能够迅速使用新的机制进行预测,而不需要花费太多的额外人力和时间设计新的算法模型。

除此之外,还有一些其他重要的评价指标,比如“冷启动”能力,能够对新用户进行准确的预测、推荐;对稀疏数据的处理能力等[4]。

3 数据结构

在开发推荐系统的早期,需要选择使用何种数据结构对庞大的数据集合进行存储和管理,保证能够简单、高效地对数据集进行访问和管理。这个问题在测

试 Netflix 的数据时变得更加明显,它拥有超过 10 亿次的投票数量,使得速度和存储效率成为非常重要的系统评价指标。常见的两种数据存储技术包括: 关系型数据库和基于 Hash 表的主存读取。

3.1　关系型数据库

存储数据最自然的方式是使用关系型数据库管理系统,它可以实现复杂的数据结构,保持不同数据之间的关联关系,使得用户能够使用结构化的查询语言访问各类相关数据。然而,尽管大多数的数据库系统已经高度优化,数据存储在硬盘上的组织方式带来较大的访存代价。另外,有些数据库比如 MySQL,没有实现针对大量数据的复杂查询算法,对于使用比如 Netflix 的大型推荐系统来说,效率偏低。

3.2　主存读取(MemReader)

使用数据库效率低的主要原因是它需要频繁访问磁盘,因此,通过将所有必要的数据存储在主存中可以解决这个问题。尽管 Netflix 的数据集非常庞大,使用主存读取(MemReader) 技术进行仔细的处理,能够将整个数据集限制在 2 GByte 内并存储于主存中,实现数据的高效访问[6]。如图 1 所示,为了实现数据的快速访问,将每项排名信息(Rating) 存储于两个 Hash 表中,分别使用 uid 和 mid 进行索引,注意,因为性能的原因,首先为每个电影和用户计算平均评分。这种情况下,uid 和 mid 都只需要不超过 24 bits 的位数来存储。在 Java 系统中,整型(int) 数字使用 32 位来表示,于是将 uid 或者 mid 存储于 32 位的高 24 位,然后将排名投票信息存储于剩余的低 8 位。所有的操作都可以转换成位运算,由于位运算执行效率高,这项技术能够降低主存的使用而不会减慢访问速度。

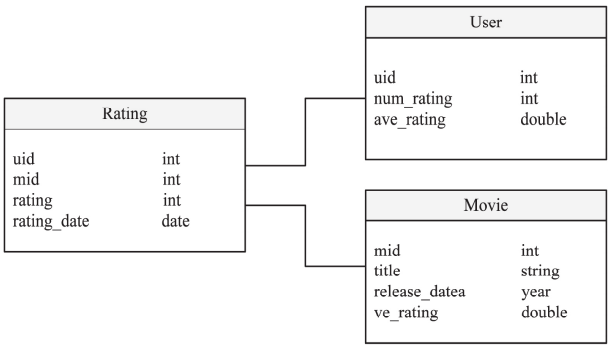

图 1　基于主存读取技术的 Hash 表组织结构

图 2 描绘了一个用户 - 电影的 Hash 表实现方式[6]。保存了一个哈希表,其中包含了被每个用户打过分的电影数目和所有评分的加和(类似的,还有每个电影的被评分的次数和总分) ,这使得一个 MemReader 项目具有高效的计算效率,并且能够高效地添加用户。主存读取技术在应对 Netflix 类似的大型数据集时非常高效。

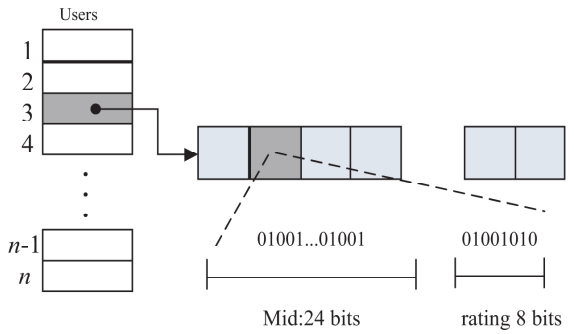

图 2　基于主存读取技术的用户-电影的 Hash 表实现

4　协同过滤算法分类

4.1　基于存储的推荐系统

基于存储的算法使用整个数据集合进行协同过滤,首先找到与目标用户相似的其他用户,使用他们的偏好为目标用户进行预测[5]。

4.1.1　算法描述

首先需要定义相似度的测量方法,以此确定用户之间的关联程度,相似度可以使用 [-1,1] 之间的值进行描述,1 表示两者的偏好完全相同,-1 表示偏好完全相反。

有两种方法进行相似度测量,一种是使用在样本相关性评估中广泛应用的皮尔逊相关系数(Pearson Correlation Coefficient) ,它通过分析用户各自的整体投票变化趋势评估其相关性,也就是说,以用户各自的平均投票情况为基准,分析每个商品的得票情况: 对于两个用户共同评价过的商品,如果投票趋势变化大致相同,两个用户正相关; 否则,两个用户负相关。另一种方法称为向量相似度,可以将两个用户对所有商品的投票情况看作是 n 维空间中的两个向量, n 是商品的数量,因此,可以计算两个向量的夹角,直观来说,如果两个向量基本的方向大致相同,它们是正相关的,如果指向相反,则负相关。通过对两个向量的夹角取余弦值,可以得到 [-1,1] 区间的值,以此对相似度进行量化。

为了预测当前用户对某个商品的评分,需要根据之前计算的用户相似度,得到其他所有用户对某一商品的打分情况相对于当前用户对该物品打分情况的权重关系,然后,根据已知的其他用户对该商品的投票情况和每个用户所占的权重,完成对当前用户对当前商品投票情况的预测。由此也可看出,权重高的用户对当前用户影响较大,这种影响可以是正影响或者负影响。

以上描述的算法在实际情况中,仍需做出一定的改进。比如说,(1) 在数据集相对稀疏的情况下,使用相关系数进行相似度测量会不准确,也就是说,当两个

用户几乎没有共同评价过的商品的时候,他们之前的权重关系就无法表示实际的情况,相反却能过度影响整体的投票结果,通过使用默认投票虚构一些他们共同评价过的商品能够使投票结果更加平滑;(2)凭借直觉可以发现,那些被公众都喜欢的物品,相对于不常见的物品来说,对于权重的影响较小。比如,如果大家每个人都喜欢看星际迷航的电影,据此得到的相似度无法用来决定两个人喜欢一个不常见电影的相对权值。因此,可以使用翻转投票率的方法解决这个问题;(3)还可以通过按照指数增加或者减小权重,增加权重的差别,这种方法有时会对推荐结果产生微小的改进。

4.1.2 算法优缺点

基于存储的算法预测质量相对较好,实现算法相对简单,由于它每次均使用整个数据集合进行预测,使得数据库每次更新后,预测也会发生相应变化,提高预测的准确度。然而,由于需要将所有的数据存储在主存中,大量的访存操作限制了预测速度;在数据集稀疏的情况下,其预测表现欠佳;由于该方法使用用户对商品的所有投票进行分析,包含所有潜在的随机变量,而不对数据进行外加的概括处理,可能会造成数据过度拟合或过度预测,造成预测结果不准确。

4.2 基于模型的推荐系统

基于存储的推荐系统在面对超大数据集合时,往往受限于预测速度和存储容量的瓶颈。因此,使用基于模型的推荐系统便应运而生。

4.2.1 算法描述

基于模型的推荐系统[5,7-8]从用户及其投票情况构成的数据集中提取特征,得到算法模型,然后使用该模型进行预测推荐,而不需要每次使用整个数据集合,在速度和支持大规模数据方面优势明显。基于存储的推荐系统主要是计算用户或者商品之间的相似度,使用其作为权重来对投票进行预测。与之类似,在基于模型的推荐系统中,用户或商品之间的相似度关系以模型的方式进行存储,然后使用模型进行预测推荐。

基于模型的推荐系统可以使用剪枝的方法进一步支持大规模数据集合。可以选择 k 个最相似的实体进行相似度分析和建模,以此进行预测推荐,研究者已经发现,限制相似实体的数目对于预测的准确性几乎没有影响。基于模型的推荐系统可以通过多种不同的方法来实现,可以将一个(用户,商品)对的投票预测看作是投票数是某个值的概率,以此将其转化为一个概率问题,Bayesian 网络和聚类算法通常根据这个思想。作为示例,文中实现了基于项目的协同过滤算法(见 5.1 节)。推荐系统有时候可以看作是对已有用户和其对商品的投票构成的矩阵,进行线性代数运算,使其转化为线性代数问题,以此实现推荐。作为示例,文中实现了奇异值分解算法(见 5.3 节)。

4.2.2 算法优缺点

大多数使用基于模型算法导出的计算模型比实际使用整个数据集合要小,因此,对于较大的数据集,模型可以变得足够小以提高预测效率,具有较好的可扩展性;与基于存储的推荐系统相比,基于模型的系统通常具有更快的预测速度,这主要是因为询问模型的时间一般情况下要远远短于询问整个数据集合的时间;如果建立的模型已经能够准确概括真实情况下投票情况,使用该模型就能够避免过度预测的情况。

然而,由于建立模型的过程通常会消耗大量的时间和资源,使得难以实时增加新的数据集到系统中,导致该算法灵活性变差。由于不使用整个数据集合而是选用部分数据进行预测,这可能会导致预测准确度的降低。然而,预测质量主要依赖于模型的构建方式,事实上,一个基于模型的推荐系统通常能够获得较好的预测表现。

4.3 关联规则

4.3.1 算法实现

关联规则尝试关联项目之间的因果关系[9-10]。关联规则的形式为 $A_1,A_2,A_3\cdots\Rightarrow B_1,B_2,B_3\cdots$,表示根据项目 $A_1,A_2,A_3\cdots$ 可以推出 $B_1,B_2,B_3\cdots$ 等;如果有 $A\Rightarrow B,C$,表示 A 的成立蕴含 B 和 C 的成立。可信度是描述规则的大小在 0 和 1 之间的另一个因子。如果可信度为 1,表示该关联规则永远适用;如果可信度为 0,就表示该关联规则永远不会成立。

对于一个包含 n 个项目的数据库,需要建立基于单项目关系的关联可信度矩阵,单项目关系是指满足 $A\Rightarrow B$ 形式的关联规则,这意味着需要查看所有的单项目关系,即对于评价了项目 B 的当前用户,其评价项目 A 的可能性。然后使用 n 维空间的一个向量表示每一个用户。将表示一个用户的 n 维向量与关联可信度矩阵相乘,就可以得到其推荐向量,即给定当前用户的历史评价信息,得到其将来最可能评价的项目。该推荐向量也可以被用来获得用户的偏好。

4.3.2 算法优缺点

与其他方法相比,关联规则主要对矩阵进行操作,执行效率高,预测结果准确,使用多级关联规则能够适用稀疏的数据集。缺点是无法预测评分,只能反映偏好,如果需要具体的预测评分值,则需要使用其他的算法。

5 协同过滤算法的实现方式

5.1 基于项目的协同过滤算法

基于项目的协同过滤是一种基于模型的算法[7],

数据集中不同项目的相似度通过使用某种测量的方法来获得,然后使用这些相似值对用户对商品的评分进行预测。

5.1.1　算法描述

通过观测所有对两个物品都打过分的用户信息可以分析得出两个项目之间的相似度,如图 3 所示,两个项目 i 和 j 的相似度依赖于对两个项目都打过分的用户(1, u 和 m) 的评分。

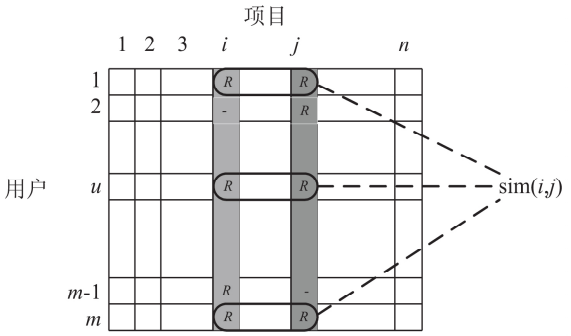

图 3　基于项目的协同过滤示例

下面列举几类典型的计算两个项目(i 和 j) 相似度(sim(i,j)) 的数学方法[1-2,5],也称之为向量相似性算法(U 表示对两个项目都评价过的所有用户集合) 。

• 余弦相似性(Cosine-based Similarity) ,也称之为基于向量的相似度(Vector-based Similarity) ,将两个项目 i 和 j 的所有评分分别记作两个向量,其相似度可以使用向量的夹角余弦值来表示,见公式(1) 。

$$\mathrm{sim}(i,j)=\cos(\vec{i},\vec{j})=\frac{\vec{i}\cdot\vec{j}}{\|\vec{i}\|\cdot\|\vec{j}\|} \tag{1}$$

• 皮尔森相关相似性(Pearson Correlation-based Similarity) ,基于所有共同用户对两个项目的评分相对于其平均得分的偏离程度来衡量,见公式(2) 。

$$\mathrm{sim}(i,j)=\frac{\sum_{u\in U}(R_{u,i}-\bar{R}_i)(R_{u,j}-\bar{R}_j)}{\sqrt{\sum_{u\in U}(R_{u,i}-\bar{R}_i)^2}\sqrt{\sum_{u\in U}(R_{u,j}-\bar{R}_j)^2}} \tag{2}$$

• 调整的余弦相似度(Adjusted cosine similarity) 考虑了不同用户可能使用不同评分标准的情况。换句话说,一些用户可能评分整体较高,另一些用户偏好打低分,为此,将用户对项目的评分 $R_{u,i}$ 分别减去各自的整体平均打分 $\bar{R}_u$ 能够得到较好的相似度结果(公式(3)) 。

$$\mathrm{sim}(i,j)=\frac{\sum_{u\in U}(R_{u,i}-\bar{R}_u)(R_{u,j}-\bar{R}_u)}{\sqrt{\sum_{u\in U}(R_{u,i}-\bar{R}_u)^2}\sqrt{\sum_{u\in U}(R_{u,j}-\bar{R}_u)^2}} \tag{3}$$

完成相似度建模之后,可以使用加权和(Weighted Sum) 的方法为所有的(用户,项目) 对的评分进行预测,见公式(4) 。首先,可以选取所有类似于目标项目,且当前用户已经打过分的项目; 然后对用户的评分情况设置权值; 最后,通过累加相似度,并按比例调整,就能够得到一个合理的预测评分值($P_{u,i}$) 。

$$P_{u,i}=\frac{\sum_{\text{all similar items},N}(s_{i,N}*R_{u,N})}{\sum_{\text{all similar items},N}(|S_{i,N}|)} \tag{4}$$

5.1.2　算法优缺点

使用 Movielens 数据库对该方法进行测试(见第 6 节) ,但测试结果并不理想,其中的原因主要是由于数据的稀疏性导致的[7-8] 。

第一个问题发生在调整的余弦相似度计算中,当某部电影只有一个观众打分的情况。由于需要将观众对电影的平分减去用户评分的平均值,对用户减去了平均值,对于一个物品,在有一个共同的用户情况下,其调整的余弦相似度为 1,表示最高的可能性,使得最相似的物品实际上成为那些拥有一个共同用户的。解决这个问题的方法是,可以规定两个电影之间共同的用户数目必须不小于一个特定的值。

第二个挑战就是当使用加权和来为用户-电影对计算评分值。因为对每个电影只存储 50 个相似的电影,并且对于每个目标电影,只考虑当前用户已经看过的相似的电影。这就使得在大多数的情况下,对于 Movielens 的数据库,对于大多数的用户,并没有太多符合这些条件的电影。因此对于较大的测试集,具有较差的预测能力。这是由数据集稀疏性导致的。

5.2　个性诊断模型

5.2.1　基本算法

个性诊断(Personality diagnosis,PD) 是一个基于概率的,模型和存储混合算法[11] ,它假设用户具有“真实个性(True Personality) ”的隐藏属性,能够被用来进行准确预测。对于数据库中的每个用户 a ,给定其评分向量(R_a) ,根据评分向量的相似度计算当前用户(a) 是某一个用户(i) 的概率,即当前用户的真实个性与其他用户个性的相似度。另一方面,在其他用户已经对某一物品(j) 评过分的基础上,当前用户(a) 也可能对该物品(j) 进行评分,将这个概率与前面的概率相乘。然后针对所有的用户,将所有的概率乘积进行累加,拥有最高概率的打分是预测的当前用户对某个物品的打分,如公式(5) 所示:

$$\max_{h=1,2,\cdots,5}\sum_{i=1}^{n}\Pr(r_a(j)=h\mid r_i(j)=y)\cdot\Pr(a^{\text{true}}=i\mid R_a,R_i) \tag{5}$$

其中, h 是一个可能的评分, n 是用户的数目, $r_a(j)$ 是当前的用户 a 对项目 j 的评分。在实验中,该公式可以

转换为公式(6),其中,两个用户共同评价过的电影使用 l 到 m 进行编号,参数 σ 的值根据实际数据集进行选择(实验中其值为 2)。

$$\max_{h=1,2,\cdots,5}\frac{1}{n}\sum_{i=1}^{n}\left(e^{-(h-r_i(j))^2/2\sigma^2}\right)\prod_{l=1}^{m}\left(e^{-(r_a(l)-r_i(l))^2/2\sigma^2}\right) \tag{6}$$

5.2.2 算法优化

上面的方法需要循环访问所有其他用户的信息来为当前用户进行预测,具有基于存储的协同过滤算法的性能缺点。因此,可以根据基于模型的算法,选择只迭代其他一部分用户,比如选取 50 个最相似的用户,并使用调整的余弦相似度测量方法。如表 1 所示,与基于存储算法实现的 PD 相比,基于模型的算法具有明显的加速,而几乎没有丢失精度[9]。

5.3 维度缩减和奇异值分解

5.3.1 维度缩减(Dimensionality Reduction)

表 1 个性诊断方法不同实现方式的比较

实验	基于存储算法的 PD (所有用户)	基于模型算法的 PD (50 个最相似用户)
实验 1	均方根误差: 1.112 4, 时间: 388 秒	均方根误差: 1.199 6, 时间: 27 秒
实验 2	均方根误差: 1.127 1, 时间: 367 秒	均方根误差: 1.202 3, 时间: 28 秒

可以使用特征空间对数据库进行表示,每一个特征向量可以表示一个项目,每一个向量点表示一个用户对项目的评分。该方法可以扩展到任意数目的维度。这里需要解决维度过高的问题,进行维度缩减[12]。例如,如果每个喜欢物品 A 的用户也都喜欢 B,就可以将 A 和 B 归为一组,形成特征集。然后通过查看两个用户对每个特征集而不是每个物品的打分,来比较这两个用户。

如果有一个具有 30 000 部电影的数据库,每个用户就是一个长度为 30 000 的向量,其存储和比较操作速度相对较慢,且消耗大量内存。使用较小的维度不仅可以提高预测效率,还可以提高预测的准确度。比如说,假设有两个用户都喜欢科幻电影,如果一个用户对《Star Wars》评价较高,而另一个用户对《Empire Strikes Back》评价较高,这两部电影同属于一个电影集,据此可以推断,这两个用户非常相似。相比而言,如果基于单个电影对两个用户进行比较,由于其喜欢的电影不同,使得他们并不是那么相似,而只有那些共同评价过的电影才会影响到他们的相似度,影响预测的准确性。

5.3.2 奇异值分解(Singular Value Decomposition)

基于信息检索的背景,Deerwester 等人曾经对维度缩减的问题进行了检验[13]。通常情况下,文档的比较是通过对词组的内容进行比较实现的,Deerwester 提出创建能够代表多个词组的特征集,然后对其特征进行比较,他们使用了奇异值分解的数学方法。之后,Sarwar 等人将这一方法应用到推荐系统中[14]。

奇异值分解[15]是将一个 $m\times n$ 的矩阵 X 分解成三个矩阵 U、S 和 V,其中 S 是对角矩阵,包含矩阵 X 的 r 个奇异值,线性无关的概念正好可以用在之前的例子中,如果每个喜欢《Star Wars》的用户都喜欢《The Matrix》,这两个电影向量就会线性相关。

$$\underset{m\times n}{\boldsymbol{X}}=\underset{m\times r}{\boldsymbol{U}}\underset{r\times r}{\boldsymbol{S}}\underset{r\times n}{\boldsymbol{V}^{\mathrm{T}}}=\begin{pmatrix}x_{11}&\cdots&x_{1n}\\ \vdots&\ddots&\vdots\\ x_{m1}&\cdots&x_{mn}\end{pmatrix}=\begin{pmatrix}u_{11}&\cdots&u_{1r}\\ \vdots&\ddots&\vdots\\ u_{m1}&\cdots&u_{mr}\end{pmatrix}\begin{pmatrix}s_{11}&0&\cdots\\ 0&\ddots&\vdots\\ \vdots&\cdots&s_{rr}\end{pmatrix}\begin{pmatrix}v_{11}&\cdots&v_{1n}\\ \vdots&\ddots&\vdots\\ v_{r1}&\cdots&v_{rn}\end{pmatrix} \tag{7}$$

如果大多数喜欢电影 A 的用户也喜欢 B,为了比较 A 和 B,可以简单地在矩阵 S 中,保持前 k 个奇异值($k<r$),从而有效降低原始空间的维度。

在实际应用中,使用基于 SVD 的推荐系统进行预测评分,第一步将数据集表示为矩阵,其中用户为行,项目为列,矩阵中每一个值为评分值。为了提供一个基准,对于原来的空值,使用电影的平均得分替代,然后进行奇异值的分解[13-14]。

6 不同算法实现与性能比较

针对 Movielens 数据库(10 万数据项),使用 Python 语言对向量相似性算法(Vec. Sim.)、个性诊断算法(PD)、奇异值分解算法(SVD)和关联算法(Corr.)进行了实现,并与求全局平均值方法(Avg.)得到的预测结果进行比较,通过计算相对用户实际打分的均方根误差(Root Mean Squared Error,RMSE)对预测结果的准确性进行评估。

实验结果如图 4 所示。

可以看出,关联算法的 RMSE 值最低,表示其对于小的稀疏数据集预测效果最好,向量个性诊断由于不适合小的稀疏数据集,预测结果最差。当将算法应用于较大的 Netflix 数据集时,奇异值分解算法的预测效果最好(RMSE = 0.92),优于向量相似度算法(RMSE=0.99)。

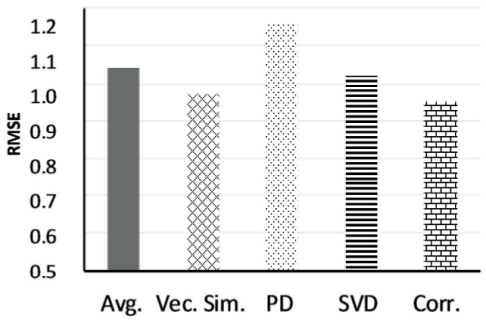

图 4　基于 Movielens 库实现的协同过滤算法的比较

同时使用拥有 13 万数据项的稀疏成绩单数据库对几种典型的算法进行了测试,如图 5 所示,稀疏数据是通过分别移除 1 门、5 门、10 门课程的成绩实现的。通过比较可以发现,向量相似性算法与关联算法取得了较好的性能,相比之下,随着数据稀疏性的不断增加,平均算法的表现不断下滑。

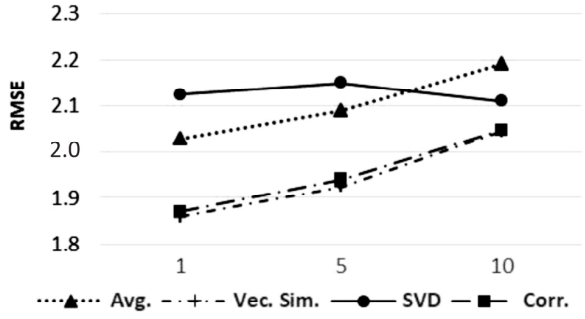

图 5　基于成绩单库实现的协同过滤算法的比较

7　结束语

推荐系统能够根据用户的兴趣和行为特征,为其推荐感兴趣的项目,已经广泛应用于互联网和大数据系统。协同过滤技术综合考虑用户的历史特征和用户之间的相似关系,为用户进行预测推荐。文中阐述了协同过滤技术的概念内容、数据结构、评价标准和分类,包括基于存储、基于模型和关联规则的推荐系统,对基于项目的协同过滤、个性诊断以及维数缩减和奇异值分解等技术实现进行了详细的描述、对比,并通过实验进行介绍,分析优缺点,为设计人员提供依据。

参考文献:

[1] 许海玲,吴　潇,李晓东,等.互联网推荐系统比较研究[J].软件学报,2009,20(2):350-362.

[2] 刘青文.基于协同过滤的推荐算法研究[D].合肥:中国科学技术大学,2013.

[3] 程　曦,陈　军.基于 MapReduce 与项目分类的协同过滤算法[J].计算机工程,2016,42(7):194-198.

[4] 杨怀洲,张留美.一种物联网协同过滤 QoS 预测与服务推荐方法[J].计算机工程,2016,42(11):1-7.

[5] BREESE J S,HECKERMAN D,KADIE C.Empirical analysis of predictive algorithms for collaborative filtering[C]//The 14th conference on uncertainty in artificial intelligence.San Francisco,CA,USA:[s.n.],1998.

[6] LEW D,SOWELL B,STEINBERG L E,et al.Recommender systems[EB/OL].[2016-10-16].http://www.cs.carleton.edu/cs_comps/0607/recommend/recommender/index.html.

[7] DESHPANDE M,KARYPIS G.Item-based top-n recommendation algorithms[J].ACM Transactions on Information Systems,2004,22(1):143-177.

[8] SARWAR B M,KARYPIS G,KONSTAN J A,et al.Item-based collaborative filtering recommendation algorithms[C]//The 10th international world wide web conference.New York,NY,USA:[s.n.],2001:285-295.

[9] KIM C, KIM J. A recommendation algorithm using multi-level association rules[C]//IEEE/WIC international conference on web intelligence.Washington,DC,USA:IEEE,2003.

[10] 王英博,马　菁,柴佳佳,等.基于 Hadoop 平台的改进关联规则挖掘算法[J].计算机工程,2016,42(10):69-74.

[11] PENNOCK D,HORVITZ E,LAWRENCE S,et al.Collaborative filtering by personality diagnosis: A hybrid memory- and model-based approach[C]//The 16th conference on uncertainty in artificial intelligence.Standford,CA,USA:[s.n.],2000.

[12] SARWAR B M,KARYPIS G,KONSTAN J A,et al.Application of dimensionality reduction in recommender system - a case study[C]//ACM WebKDD 2000 web mining for e-commerce workshop.Boston,MA,USA:ACM,2000.

[13] DEERWESTER S,DUMAIS S T,FURNAS G W,et al.Indexing by latent semantic analysis[J].Journal of the American Society for Information Science,1990,41(6):391-407.

[14] SARWAR B M,KARYPIS G,KONSTAN J A,et al.Incremental singular value deocmposition algorithms for highly scalable recommender systems[C]//The 5th international conference on computer and information technology (ICCIT).Chicago,IL,USA:[s.n.],2002.

[15] BRAND M.Fast online SVD revisions for lightweight recommender systems[C]//The 3rd SIAM international conference on data mining.San Francisco,CA,USA:[s.n.],2003.