协同过滤推荐系统中数据稀疏问题的解决*

吴 颜[1]，沈 洁[1]，顾天竺[1]，陈晓红[1]，李 慧[1,2]，张 舒[1]

(1. 扬州大学 信息工程学院 计算机科学系，江苏 扬州 225009；2. 淮海工学院 计算机科学系，江苏 连云港 222005)

摘 要：介绍了现有协同过滤推荐的几种主要算法。它们对数据稀疏性问题都有一定的缓和作用。通过在数据集 MovieLens 上的实验，分析了各个算法在不同稀疏度下的推荐质量，为针对不同数据稀疏度的系统实现提供了可靠依据。

关键词：电子商务；推荐系统；协同过滤；数据稀疏；相似性

中图分类号：TP391 **文献标志码**：A **文章编号**：1001-3695(2007)06-0094-04

Algorithm for Sparse Problem in Collaborative Filtering

WU Yan[1], SHEN Jie[1], GU Tian-zhu[1], CHEN Xiao-hong[1], LI Hui[1,2], ZHANG Shu[1]

(1. *Dept. of Computer Science, Institute of Information Technology, Yangzhou University, Yangzhou Jiangsu* 225009, *China*; 2. *Dept. of Computer Science, Huaihai Institute of Technology, Lianyungang Jiangsu* 222005, *China*)

Abstract: This paper summarized several primary algorithms, and experimented on MovieLens data. And analyzed different algorithm with the experimental results.

Key words: e-commerce; recommender system; collaborative filtering; data sparse; similarity

随着信息技术的发展和信息资源的膨胀，对于用户来说，寻找自己感兴趣的商品信息已经成为一件困难且昂贵的事情。因此，电子商务的推荐系统应运而生。推荐系统是电子商务个性化服务的重要组成部分。个性化推荐系统包括热销商品推荐、新品推荐、相关产品推荐和与用户群同兴趣推荐等。目前，许多电子商务网站都已经使用了推荐系统，如 Amazon、CDNow、Drugstore 和 Moviefinder 等。推荐系统利用数据挖掘技术在电子商务网站中帮助顾客访问感兴趣的产品信息并产生推荐。这些系统对扩大销售量，增加交叉销售额，提高顾客忠诚度等方面都有较大贡献。本文主要讨论与用户群同兴趣推荐。在该领域最成功的技术是协同过滤推荐技术。

协同过滤技术主要是利用目标用户对产品的历史评价与其他用户相匹配，以预测用户对未评价产品的评价，从而产生高效的推荐。在预测目标用户对某产品的评价前，首先需要根据该用户的历史评分和反馈信息，选择目标用户的最近邻居集，即与目标用户兴趣相似的用户集合(图 1)。在选定最近邻居之后，根据邻居对某些产品的评分对目标用户的评分进行预测，产生推荐。

虽然协同过滤技术得到了广泛应用，但是它仍然存在一些弊端。主要有数据稀疏性问题、系统可扩展性问题和同义词问题。本文主要讨论数据稀疏性问题。

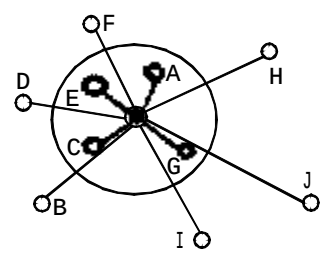

图 1 最近邻居的选择

1 协同过滤推荐技术与数据稀疏性

协同过滤推荐技术是目前推荐系统中最成功的技术；它基于一个假设。如果用户对一些产品的评分比较相似，则对其他产品的评分也比较相似[1]。

传统的协同过滤推荐技术，主要是通过寻找与目标用户兴趣爱好相似的用户，并根据其喜爱的产品，对目标用户喜爱的产品进行预测，产生推荐。例如，对一个电影网站的某个用户推荐电影，首先寻找与目标用户喜爱相同电影的用户群，即最近邻居集；然后，将邻居用户喜爱且目标用户未知的电影按特定的方法推荐给目标用户。

下面以向用户推荐电影为例，具体介绍推荐过程。假设若两个用户喜爱相同电影的数目越多，这两个用户就越相似。首先要寻找目标用户的最近邻居集。目前，大多数的协同过滤推荐系统是通过相似性计算来得到最近邻居集的。因此，为了衡量用户之间的相似性，首先将用户和电影信息进行整理，得到

收稿日期：2006-04-11；修返日期：2006-06-02 基金项目：江苏省自然科学基金资助项目(BK2005046)

作者简介：吴颜(1981-)，陕西西安人，硕士研究生，研究方向为数据挖掘(wodename_wu@hotmail.com)；沈洁(1955-)，男，江苏姜堰人，教授，硕导，研究方向为数据挖掘、信息管理；顾天竺(1981-)，男，江苏无锡人，硕士研究生，研究方向为数据挖掘；陈晓红(1981-)，硕士研究生，研究方向为数据挖掘；李慧(1979-)，江苏连云港人，硕士研究生，研究方向为数据挖掘；张舒(1979-)，江苏连云港人，硕士研究生，研究方向为数据挖掘.

用户—电影矩阵 $A(i, j)$ (表 1)。

表 1　用户—电影矩阵

Customer	$Prod_1$	$Prod_2$	…	$Prod_m$
$Cust_1$	R_{11}	R_{12}	…	R_{1m}
$Cust_2$	R_{21}	R_{22}	…	R_{2m}
…	…	…	…	…
$Cust_n$	R_{n1}	R_{n2}	…	R_{nm}

其中, n 行代表 n 个用户; m 列代表 m 个电影; 元素 R_{ij} 代表用户 i 对电影 j 的评分。评分代表了用户对电影的喜爱程度。衡量两个用户 i 与 j 之间相似性时, 首先得到用户 i 与 j 的所有评分电影和评分数据; 然后用不同的方法计算得到两个用户之间的相似性。下面介绍几种相似性的计算方法。

计算相似性比较成熟的算法主要有:

(1) 相关相似性: 通过皮而森相关系数进行衡量, 设经用户 i 与 j 共同评分的产品集合用 I_{ij} 表示, 则用户 i 与 j 之间的相似性:

$$sim(i, j) = \sum_{p\in I_{ij}} (R_{i,p} - \overline{R_i})(R_{j,p} - \overline{R_j}) / \left[\sqrt{\sum_{p\in I_{ij}} (R_{i,p} - \overline{R_i})^2} \times \sqrt{\sum_{p\in I_{ij}} (R_{j,p} - \overline{R_j})^2} \right]$$

其中, $R_{i,p}$ 表示用户 i 对产品 p 的评分; $\overline{R_i}$ 和 $\overline{R_j}$ 分别表示用户 i 与 j 对项目的平均评分。

(2) 余弦相似性: 将用户评分看做是 n 维空间上的向量, 通过计算向量之间的夹角来衡量相似性。若用户对某产品没有评分, 则将相应分量值设为 0。设用户 i 和 j 在 n 维产品空间上的评分分别表示为向量 i, j 则用户 i 和 j 之间的相似性:

$$sim(i, j) = \cos(i, j) = i \times j / (\| i \| \| j \|)$$

其中, 分子为两个用户评分向量的内积; 分母为两个用户向量模的乘积。

(3) 修正的余弦相似性[1]: 由于不同用户对产品评分尺度不同, 会影响预测的准确性, 而传统余弦相似性没有考虑到这个问题。对其进行修正, 减去用户对产品的平均评分, 以改善准确性。设经用户 i 和 j 共同评分的产品集合用 I_{ij} 表示, I_i 和 I_j 分别表示经用户 i 和 j 评分的产品集合, 则用户 i 和 j 之间的相似性:

$$sim(i, j) = \sum_{p\in I_{ij}} (R_{i,p} - \overline{R_i})(R_{j,p} - \overline{R_j}) / \left[\sqrt{\sum_{p\in I_i} (R_{i,p} - \overline{R_i})^2} \times \sqrt{\sum_{p\in I_j} (R_{j,p} - \overline{R_j})^2} \right]$$

其中, $R_{i,p}$ 表示用户 i 对产品 p 的评分; $\overline{R_i}$ 和 $\overline{R_j}$ 分别表示用户 i 和 j 对产品的平均评分。

通过上述相似性计算方法, 得到用户之间的相似性。将相似性数值按大小排序, 取前 N 个作为目标用户的最近邻居集 M_p。根据邻居用户对电影的评分, 预测目标用户 i 对电影 p 的评分:

$$p_{i,p} = \overline{R_u} + \sum_{n\in M_p} sim_{p,n} \times R_{i,n} / \sum_{n\in M_p} (|sim_{p,n}|)$$

其中, $\overline{R_u}$ 是用户 u 的平均评分。

最后, 根据预测评分的大小, 对电影进行推荐。

由以上推荐过程容易看出, 用户的评分信息对相似性的确定至关重要。因此, 用户评分数据稀疏, 造成用户相似性计算误差大, 用户评分预测的准确性降低。例如, 一个出售书籍的商务网站, 拥有两百万书籍, 然而每个用户评价的书籍大多不超过 100 本。这样, 得到的用户—书籍矩阵则是一个稀疏矩阵。显然, 基于这样的稀疏矩阵计算得来的用户相似性是不准确的。因此, 随着用户和产品数量的增加, 数据的稀疏程度增加, 系统推荐质量则急剧下降。

数据稀疏问题严重制约着协同过滤推荐系统的发展。对于大型商务网站来说, 由于产品和用户数量都很庞大, 用户评分产品一般不超过产品总数的 1%[2], 两个用户共同评分的产品更是少之又少, 解决数据稀疏问题是提高推荐质量的关键。

2　几种主要的协同过滤算法

为了提高推荐质量, 许多研究人员都试图缓和数据稀疏问题。他们从不同的角度对用户和产品信息进行分析、处理, 降低数据的稀疏程度。这些算法各有利弊。

2.1　基于项目的协同过滤推荐算法

传统的协同过滤推荐算法是通过计算用户之间的相似性, 寻找与目标用户兴趣相似的一组用户, 作为目标用户的最近邻居。然而, 由于数据的极端稀疏性, 两个用户共同评分的产品非常少, 得到用户之间的相似性很有可能为 0。因此, 出现了基于项目的协同过滤推荐技术。

基于项目的协同过滤推荐算法[1,3,4], 从产品角度进行分析, 寻找与目标产品相似的产品集合, 然后进行预测和推荐。它基于一个假设, 即用户对与其感兴趣产品相似的产品也感兴趣。由于项目间的相似性相对稳定, 而通常项目的数量比用户数量少, 这样可以减少计算量, 降低数据稀疏性。

算法步骤:

(1) 通过相似性算法, 计算列向量的余弦相似性, 即产品向量的相似性。

(2) 选择相似性最高且没有被目标用户评价过的前 M 个产品, 作为产品的邻居集合 M_p。

(3) 对邻居集合中产品的评分进行加权求和, 得到目标用户对目标产品的预测评分。

$$p_{i,p} = \sum_{n\in M_p} sim_{p,n} \times R_{i,n} / \sum_{n\in M_p} (|sim_{p,n}|)$$

其中, M_p 为邻居产品集合。

(4) 按照预测值的高低选择前 N 项推荐给用户。

2.2　降低矩阵维数的技术

降低矩阵维数的技术可对原始稀疏数据直接进行数据处理, 降低数据稀疏性。主要算法有单值分解、聚类等。

2.2.1　单值分解

单值分解算法利用矩阵的单值分解原理, 对用户—产品矩阵进行分解, 从而降低矩阵的维数, 抽取出主要信息[2,5]。

算法步骤:

(1) 使用每个产品的平均评分——列平均值——填充矩阵中的未评分项。

(2) 利用用户的平均评分——行平均值——进行标准化, 产生矩阵 R。

(3) 对 R 进行单值分解,得到

$$R = U \times S \times V$$

其中,U 为 $m \times r$ 阶矩阵,V 为 $r \times n$ 阶矩阵,S 为 $r \times r$ 阶方阵。

(4) 根据矩阵的单值分解原理,S 可以被降维到 k 阶($k < r$),得到 S_k;同理可以得到 U_k、V_k。于是得到与矩阵 R 相似的低维矩阵 R_k。即有

$$R_k = U_k \times S_k \times V_k$$

由上式可以看出,$U_k S_k^{1/2}$ 是一个 $m \times k$ 阶矩阵,它包含了所有 m 个用户信息,称其为用户矩阵。

(5) $U_k S_k^{1/2}$ 矩阵上进行用户相似性计算,得到最近邻居集。

(6) 根据同一产品被邻居评分的频繁程度产生推荐。

从算法步骤可以看出,通过单值分解得到的较低维的 $U_k S_k^{1/2}$ 矩阵比原始用户—产品稠密,并且抽取出了所有用户信息。在这个矩阵上进行相似性计算,可以减少计算量,提高在线推荐速度,并且提高推荐质量。实验表明,该算法在分解矩阵过程中,不可避免数据遗失。当原始矩阵极度稀疏时,单值分解算法试验结果并不理想。算法需要较大的计算量,较少的存储空间。

2.2.2 聚类

单值分解通过矩阵运算降低数据稀疏性,聚类[6~9]则是通过一些聚类算法将产品或用户聚成若干具有共同性质的类;然后在小的聚类数据中产生推荐。聚类算法有很多,本文主要讨论用 K-means 对用户的聚类算法。K-means 算法是目前最通用的快速聚类算法[10]。数字 K 是算法的一个输入,它代表聚类的个数。

算法步骤:

(1) 取前 K 个用户作为 K 个独立的聚类质心;剩余的每个用户与其最近质心进行比较。

(2) 在形成聚类质心的基础上,重新计算聚类的质心。

(3) 聚类内部的成员关系被重新估算。重复(1) ~(3),直到产生的 K 个聚类不再变化。

(4) 在目标用户所在的类中进行用户相似性计算,主要运用第一部分中的相关相似性方法计算,得到最近邻居集合。

(5) 对最近邻居的评分数据进行加权处理:

$$P_{i,j} = \overline{R_i} + \sum_{i=1}^{K} w_{i,j} \times \mathrm{Sim}_{i,j} \times (R_{i,j} - \overline{R_i}) / (\sum_{i=1}^{K} w_{i,j} \times \mathrm{sim}_{i,j})$$

其中,$P_{i,j}$ 是用户 i 对产品 j 的预测评分;$W_{i,j}$ 是用户 i 对产品 j 的权值。

(6) 预测产生后,根据预测评分的高低对目标用户进行推荐。

2.3 基于内容的协同过滤算法

2.1、2.2 节中介绍的算法都是建立在用户对产品评分的基础上,在一定程度上都缓和了数据稀疏带来的问题。基于产品内容的协同过滤算法[11]与前面介绍的几种算法的不同之处,是考虑到了产品本身的信息。由于增加了信息量,可以有效提高推荐质量。

单纯的基于产品内容的算法,根据单个用户已评价产品的内容信息,如电影的导演、演员、类型等,建立用户兴趣模型,进而产生推荐。这样的算法存在一些弊端:①由于通常获得的只是产品的部分信息,其他一些未知信息很有可能影响用户的行为,这就造成了推荐的不准确。②用户的喜好通常是多样的,而单个用户评分的产品数量却非常少。这样就使推荐局限于特定类型的产品上。

基于内容的协同过滤技术则不同。目前这方面的算法主要有线性结合型和连续结合型两个类型,如图 2、3 所示。

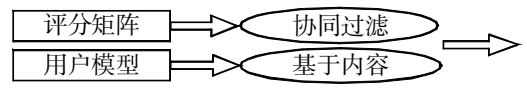

图 2 线性结合模型

图 3 连续结合模型

线性结合型算法步骤:

(1) 通过历史评价的产品内容信息,建立用户兴趣模型。

(2) 比较用户模型,得到基于内容的用户相似性。

(3) 根据用户—产品评分矩阵,得到基于评分的用户相似性。

(4) 将两个相似性进行线性组合,得到用户相似性。

(5) 根据相似性,得到目标用户的最近邻居集。

(6) 通过邻居集对目标用户进行评分预测和推荐。这个过程可以用 2.1、2.2 节介绍的算法进行。

连续结合型算法步骤:

(1) 通过历史评价的产品内容信息,建立用户兴趣模型。

(2) 根据每个用户兴趣模型的相互比较,获得目标用户的最近邻居集。

(3) 将符合目标用户或其邻居兴趣模型的产品推荐给目标用户。

以上两种算法混合了协同过滤算法和基于内容的推荐算法,缓和了传统协同过滤算法没有考虑产品本身信息的缺陷,也解决了基于内容的推荐算法中单个用户信息稀少的问题。然而,产品信息的获取和存储,是一个困难且昂贵的问题。

3 实验结果及其分析

3.1 数据集

本文采用 MovieLens 站点提供的数据集(http://movielens.umn.edu/)。MovieLens 是一个基于 Web 的研究型推荐系统,用于接收用户对电影的评分,并提供相应的电影推荐列表。该站点现在已经拥有 43 000 个用户和 35 000 部电影。随机选择了 100 000 个评分。其中每个用户至少对 20 部电影进行了评分。

3.2 度量标准

本文采用平均绝对偏差(Mean Absolute Error, MAE)作为评价推荐系统预测质量的评价标准。MAE 可以直观地对预测质量进行度量,是最常用的一种方法。MAE 通过计算预测的用户评分与实际评分之间的偏差度量预测的准确性;MAE 越小,预测质量越高。设预测的评分集合为 $\{p_1, p_2, \cdots, p_N\}$,对应的实际用户评分集合为 $\{q_1, q_1, \cdots, q_N\}$,则平均绝对偏差 MAE

为[1,3]

$$MAE = \sum_{i=1}^{N} |p_i - q_i| / N$$

3.3　实验参数

为了能更好地评估算法性能, 在实验中根据各个算法的特点, 对各个算法的参数进行了一定的设置。

在基于项目(Item-based) 的算法中, 运用余弦相似性进行相似性计算, 邻居选择为 30 个。SVD 中, 同样运用余弦相似性进行相似性计算, 并统一将数据降维至 100 维。在聚类算法中, 选择聚类数 $K = 50$。U-content 和 I-content 都是基于内容的线性结合模型算法。其中, I-content 是与基于项目的协同过滤技术进行线性结合; U-content 是与基于用户的协同过滤技术进行线性结合。

3.4　实验结果及分析

通过对上述算法在 MovieLens 数据集上的实验, 得到每个算法在不同数据稀疏程度下的 MAE 值, 如图 4 所示。

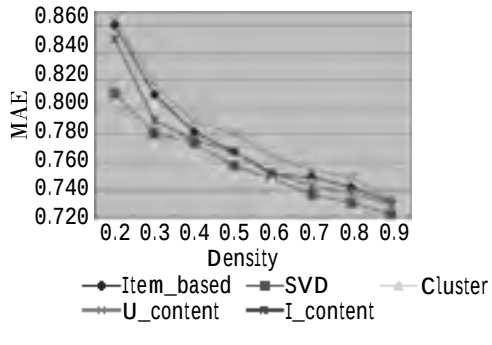

图 4　不同数据稀疏度下算法的性能

实验中对基于项目的协同过滤算法(Item-based CF) 、基于单值分解的协同过滤算法(SVD) 、基于聚类的协同过滤算法(Cluster CF) 和基于内容的协同过滤算法(Content-based) 进行了研究。通过数据稠密程度为 0. 2 ～0. 9 的变化, 研究数据稀疏程度对算法性能的影响。

从图 4 中可以看出, 随着数据稀疏程度的减小, 即数据稠密程度的增加, 各算法的 MAE 值都有所降低。这充分说明数据稀疏对算法性能的影响十分严重。

在实验的五个算法中, 当数据集稀疏程度小于 0. 4 或大于 0. 7 时, SVD 的 MAE 值最小, 性能也最好。也就是说, SVD 能有效减少数据稀疏对系统性能的影响。当数据稀疏程度在 0. 4 ～0. 7 时, 聚类算法的优势则较明显。

4　结束语

协同过滤推荐系统已经被广泛应用于电子商务、电子图书馆等众多领域, 随着用户和产品数量的不断增加, 传统算法的许多缺点逐渐暴露了出来。本文通过对几种主要的缓解数据稀疏性影响的算法的研究和实验, 对各种算法在不同数据稀疏程度下的算法性能进行了评估, 针对不同稀疏程度数据的系统实现提供了可靠的依据。

参考文献:

[1] 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[C]. 软件学报, 2003, 14(9): 1621-1628.

[2] SARWAR B, KARYPIS G, KONSTAN J, *et al*. Application of dimensionality reduction in recommender systems: a case study: proc. of the WebKDD Workshop at the ACM SIGKKD[C]. New York: ACM, 2006.

[3] SARWAR B, KARYPIS G, KONSTAN J, *et al*. Item-based collaborative filtering recommendation algorithms: proc. of the 10th International World Wide Web Conference[C]. [S. l.]: [s. n.], 2001: 285-295.

[4] KARYPIS G. Evaluation of item-based top-n recommendation algorithms: proc. of the 10th International Conference on Information and Knowledge Management (CIKM) [C]. Atlanta: [s. n.], 2001.

[5] BILLSUS D, PAZZANI M J. Learning collaborative information filters: proc. of the 15th International Conference on Machine Learning [C]. [S. l.]: [s. n.], 1998: 46-54.

[6] CONNER M O, HERLOCKER J. Clustering items for collaborative filtering: proc. of the ACM SIGIR Workshop on Recommender Systems[C]. Berkeley: [s. n.], 1999.

[7] KOHRS A, MERIALDO B. Clustering for collaborative filtering applications: proc. of CIMCA'99[C]. [S. l.]: IOS Press, 1999.

[8] UNGAR L H, FOSTER D P. Clustering methods for collaborative filtering: proc. of Workshop on Recommendation Systems at the 15th National Conf. on Artificial Intelligence[C]. Menlo Park: AAAI Press, 1998.

[9] XI Wensi, CHEN Zheng, XUE Guirong, *et al*. Scalable collaborative filtering using cluster-based smoothing: proc. of the 28th Annual International ACM SIGIR[C]. [S. l.]: [s. n.], 2005.

[10] LI Qing, KIM B M. Clustering approach for hybrid recommender system: proc. of the IEEE/WIC International Conference on Web Intelligence (WI'03) [C]. [S. l.]: [s. n.], 2003.

[11] BALABANOVIC M, SHOHAM Y. FAB: content-based, collaborative recommendation[J]. Commun. ACM, 1997, 40(3): 66-72.

[12] AGGARWAL C C, WOLF J L, WU K L, *et al*. Horting hatches an egg: a new graphtheoretic approach to collaborative filtering: proc. of the 5th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'99) [C]. San Diego: ACM, 1999: 201-212.

[13] BREESE J S, HECKERMAN D, KADIE C. Empirical analysis of predictive algorithms for collaborative filtering: proc. of the 14th Conference on Uncertainty in Artificial Intelligence[C]. Madison: Morgan-Kaufmann, 1998: 43-52.

[14] HUANG Z, CHEN H, ZENG D. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative[J]. ACM Transactions on Information Systems, 2004, 22(1): 116-142.

[15] MIRZA B J. Jumping connections: a graph-theoretic model for recommender systems[EB/OL]. http://scholar.lib.vt.edu/theses/available/etd-02282001-175040/unrestricted/etd.pdf.

[16] MIRZA B J, KELLER B J, RAMAKRISHNAN N. Studying recommendation algorithms by graph analysis[J]. J. Intel. Inf. Syst, 2003, 20(2): 131-160.

[17] HUANG Zan, CHUNG Wingyan, CHEN Hsinchun. A graph model for e-commerce recommender systems[J]. J. ASIST, 2003, 55(3): 259-274.