

推荐系统中显式评分输入的用户聚类方法研究

崔春生

(河南财经政法大学 计算机学院, 郑州 450002)

摘要: 从推荐系统的矩阵稀疏性问题出发,在保证推荐的覆盖率基础上,借助社区密度的思想并通过矩阵约简来提高推荐的准确率。运用归一化处理的用户评分矩阵对一致性的评分矩阵进行了改造,并借助社区的概念通过矩阵调整得到了一种基于用户相似评分偏好的聚类分析结果;通过构造社区网络结构图得到社区密度,并选择其中的最大值作为聚类中心,运用紧致与分离性效果函数验证了聚类结果的可靠性。该方法的提出,在不损害已有信息的基础上降低了计算复杂度,通过 Movielens 的数据检验了该方法在一定程度上的正确性和有效性。

关键词: 推荐系统; 聚类分析; 社区网络; 稀疏矩阵

中图分类号: TP311

文献标志码: A

文章编号: 1001-3695(2011)08-2856-03

doi:10.3969/j.issn.1001-3695.2011.08.015

Aggregation method of recommender systems with way of sparse matrix reduction by sociality structure

CUI Chun-sheng

(School of Computer Science, Henan University of Economics & Law, Zhengzhou 450002, China)

Abstract: Started in the sparse matrix of recommender systems, and ensured the recommended coverage rate. Based on the idea of sociality density, improved the accuracy by matrix reduction. With the user ratings matrix which had been normalized, transformed the consistency score matrix. Meanwhile, with the concept of sociality, got an aggregation analysis result based on user similarity score preferences through adjusting the matrix. Computed the sociality densities by constructing a sociality network diagram. Selected the maximum as the cluster centers, and used the compact and separation effect functions to validate the reliability of aggregation results. Compared to other method, the method reduces the computational complexity. Meanwhile, with the test of Movielens, the method is accuracy and effectiveness in a certain degree.

Key words: recommender systems; aggregation analysis; sociality network; sparse matrix

在推荐系统质量中,查全率和查准率的大小一方面取决于推荐并出现在评估集中的产品数量,另一方面取决于评估集、推荐集的大小。评估集是客观存在的,属于事后结果,无法借助技术手段得到提高。因此,提高推荐质量的一种思路是寻求高质量的推荐集,在不损害评估集的基础上提高产品推荐的精确率^[1-3]。

进一步考虑,在基于内容的推荐算法中,如果能找到相似的用户,并在相似用户(具有类似的购物需求、偏爱特征和消费能力)^[4]之间根据产品的相似度进行推荐,推荐的计算复杂度将会大幅度降低。同时,随着用户间相似程度的提高,用户间的偏好得以更加准确地描绘,推荐覆盖率自然相应提高。

由此可见,问题的核心在于用户间相似程度的提高。社区思想的本质是一种聚类的思想,社区密度大小恰好是用户相似程度的一种有效衡量手段,因此该思想的引入将会对这一问题的解决搭建一座可行的平台。所谓社区密度(sociality density),可以理解为在一个熟人圈子里每个人都互相认识的程度,在实际的社会网络中,如熟人网络中,社区密度反映了某人的朋友相互之间也是朋友的平均可能程度。协同过滤推荐系统依赖的是在熟人之间进行产品的推荐,所以电子商务虚拟社区的紧密性程度的提高有助于协同过滤推荐的质量;基于内容推荐算法依

赖的是产品间的相似性,但是若能在具有相似兴趣的电子商务虚拟社区内进行产品推荐,将会进一步提高推荐质量。

1 社区思想聚类方法

本文的研究立足于计算复杂度的降低,通过社区思想的引入,一方面改善矩阵稀疏程度,另一方面通过新的社区划分得到更加精确的用户聚类结果,从而在保证推荐系统覆盖率的基础上提高推荐准确率。

问题的解决经由以下八个步骤完成:

a) 获得评分矩阵。

推荐系统的形式化定义,设 $C = \{c_1, c_2, \dots, c_u, c_v, \dots, c_n\}$ 是所有用户(user)的集合, $S = \{s_1, s_2, \dots, s_i, s_j, \dots, s_m\}$ 是系统中所有产品(product)的集合。实际中, C 和 S 集合的规模通常很大,如上百万的用户以及上亿种产品等; $R_{ui} (u = 1, 2, \dots, n; i = 1, 2, \dots, m)$ 表示用户 c_u 对产品 s_i 的评分^[5]。

b) 评分值归一化处理。

用户的评分值是用户对待产品的一种主观态度,基于用户间性格、偏好、态度等的极大的差别,评分结果会有一定的差异。例如有的用户在对产品的评分过程中觉得 7 分(假设评分等级为 9 级)已经很高了,而有的用户觉得 7 分比较低^[6]。

评分值之间的差异将会影响到产品的推荐结果,因此需要对评分结果进行归一化处理,即把评分值变换到 $[0,1]$ 上,以便在同一基准上进行探讨。

采用半三角模糊函数,可以将用户 c_u 对产品 s_i 的评分结果 R_{ui} 表示为

$$R_{ui}' = \frac{R_{ui} - \min R_u}{\max R_u - \min R_u} \quad (1)$$

其中: $\max R_u$ 和 $\min R_u$ 分别表示用户 c_u 对所有产品的评分值中的最大值和最小值,即

$$\begin{aligned} \max R_u &= \max_{i=1,2,\dots,m} \{R_{ui}\} \quad u=1,2,\dots,n \\ \min R_u &= \min_{i=1,2,\dots,m} \{R_{ui}\} \quad u=1,2,\dots,n \end{aligned}$$

特殊情况下,如果 $\max R_u = \min R_u$,则定义 $R_{ui}' = 0$ 。

因此可以定义用户 c_u 喜欢的产品的集合为

$$\left\{ s_i : R_{ui}' \geq \varepsilon, \text{ i. e. } R_{ui} \geq \frac{\max R_u + \min R_u}{2} \right\}$$

其中: ε 称做喜欢度阈值。 ε 可以依据用户在产品群中的喜好程度确定,一般情况下,定义产品中一半是用户喜欢的产品,一半是用户不喜欢的产品,因此可以取 $\varepsilon = 0.5$ 。

c) 定义新矩阵。

鉴于评分矩阵 R_{ui} 稀疏性的客观存在,经过归一化处理后的评分矩阵 R_{ui}' 仍为稀疏矩阵,需要对 R_{ui}' 矩阵进行进一步的处理,以降低矩阵稀疏性。

(a) 定义 R_{ui} 矩阵的有数格和无数格。若 $R_{ui}' \neq 0$,则此位置称为有数格,否则称为无数格。

(b) 根据有数格的不同评分值修正数据。若 $R_{ui}' = 1$,即每一行的最大值,该值不变;若存在有数格, $R_{ui}' \geq \varepsilon$ 且 $R_{ui}' \neq 1$,则定义 $R_{ui}' = 0$;若存在有数格, $R_{ui}' < \varepsilon$,则定义 $R_{ui}' = -1$ 。

d) 调整稀疏矩阵。

通过c)的定义,新矩阵中的有数格只有 $-1, 0, 1$ 三类数。下面对新矩阵进行行调整和列调整,调整的原则为:先行调整,后列调整; $R_{ui}' = 1$ 的有数格尽量靠近矩阵主对角线的位置,若某一行某一列存在多个 $R_{ui}' = 1$,可以任取一个; $R_{ui}' = 0$ 的有数格尽量位于主对角线的两侧;不考虑 $R_{ui}' = -1$ 的有数格, $R_{ui}' = -1$ 可以暂时看做无数格。

e) 划分社区。

沿着主对角线的方向,将 $R_{ui}' = 1$ 和 $R_{ui}' = 0$ 最密集的地方框起来就构成了一个社区,可以认为同一社区内的用户具有类似的偏好。

显然,社区划分过程中每个人的观点不同,得到的社区划分结果就会有所不同。因此,需要运用社区密度评价社区划分结果的可靠性,保证社区内用户偏好的相似性程度。

f) 建立社区网络图。

定义节点 u 表示社区内用户 c_u ,社区内的其他用户称为节点 u 的邻居节点,社区外的其他用户称为节点 u 的一般节点。

对于任意 R_{ui}' 为有数格:

若 $R_{ui}' \geq 0$ 且 $R_{vi}' \geq 0$,则在节点 u 和 v 之间填一条直线;

若 $R_{ui}' < 0$ 且 $R_{vi}' < 0$,则在节点 u 和 v 之间填一条直线;

若 $R_{ui}' \geq 0$ 且 $R_{vi}' < 0$,则在节点 u 和 v 之间去一条直线;

若 $R_{ui}' < 0$ 且 $R_{vi}' \geq 0$,则在节点 u 和 v 之间去一条直线。

进而得到节点 u 的社区网络图。

节点 u 的社区网络图反映了所有系统用户与用户 c_u 的紧密/松散关系。

g) 计算社区密度。

社区密度用 D 来表示,在实际的社会网络中,如熟人网络中, D 反映了某人的朋友互相之间也是朋友的平均可能程度。在电子商务系统中, D 反映了某用户喜欢的产品也被其他用户欣赏的平均可能程度。

对选定的节点 u ,假设节点 u 所在的社区 $q(q=1,2,\dots,K)$ 共有 $t-1$ 个邻居节点,其中 t 为社区人数(或者社区规模)。整个网络中有 $k_u(k_u \leq n-t)$ 个其他节点与节点 u 相连,如果在社区内 $t-1$ 个邻居节点间存在着 e_u 条边,则节点 u 的社区密度 D_u 可以通过下式计算:

$$D_u = \frac{e_u}{k_u(k_u - 1)/2} \quad (2)$$

其等价定义为

$$D_u = \frac{\text{社区内与节点 } u \text{ 相连的三角形的个数}}{\text{网络中以节点 } u \text{ 为中心的三连组个数}}$$

其中:分母可以看做这个 k_u 节点间最多可能存在的边数。整个社区的社区密度 D 为社区内所有单个节点的社区密度 D_u 的平均值,即

$$D = \frac{1}{t} \sum_{u=1}^t D_u$$

h) 聚类质量评价。

上文得到的社区 q 就是一个基于用户的兴趣类,同一社区中的用户具有相似或相近的偏好。社区划分的质量实质上就是聚类质量,聚类质量也称为聚类的有效性^[7]。聚类的质量决定于初始值的设定、聚类数的选择、聚类方法等诸多因素。评价的标准是“类内组合度高,类间耦合度低”,即各个聚类中心的间距尽可能地大,而样本与其中中心间距尽可能地小。

一种客观的聚类质量评价方法是紧致与分离性效果函数(该方法评价效果较好)。定义 $S(U, K)$ 是输入样本与它们相应的聚类中心间距的平均值与聚类中心最小间距的比值,即

$$S(U, K) = \frac{\frac{1}{n} \sum_{q=1}^K \sum_{u=1}^n u_{qu}^2 |x_u - y_q|^2}{\min_{i,j} |y_i - y_j|^2} \quad (3)$$

其中, K 的最优选择为 $\max_K \left\{ \max_{\Omega} S(U, K) \right\}, K=2,3,\dots,n-1, \Omega$ 为所有聚类结果; $y_q = \max_u D_u, u=1,2,\dots,t$ 代表第 $q(q=1,2,\dots,K)$ 个社区(聚类中心), $x_u(u=1,2,\dots,n)$ 代表用户 c_u (样本数据), $u_{qu} \in \{0,1\}$ 代表用户 c_u 属于第 q 个社区的隶属度。

2 数据实验及计算

2.1 数据背景

MovieLens 是一个大型的电影推荐系统网站(www.movielens.org),是历史最悠久的推荐系统。它由美国 Minnesota 大学计算机科学与工程学院的 GroupLens 项目组创办,是一个非商业性质的、以研究为目的的实验性站点。GroupLens 项目组是全世界推荐系统领域最具影响力的学术研究团体,他们为推荐系统的发展做出了许多卓越的贡献。MovieLens 主要使用 collaborative filtering 和 association rules 相结合的技术,向用户推荐他们感兴趣的电影。

MovieLens 成立于 1997 年,作为一个基于 Web 的研究型电影推荐系统,每周都有几百个用户登录该网站对影片进行评分,并得到相应的影片推荐列表。MovieLens 公布的数据集包括 6 040 个用户针对 3 900 部电影所进行的将近 1 000 000 条打分数据,是使用最为广泛的实验资料,并且已经成为评价推荐算法的基准数据集。

MovieLens 使用打分的方法获取用户对电影的偏好数据, 打分时分别用 must see、will enjoy、it’s ok、fairly bad、awful 表示用户对产品的态度, 并依次用 5 ~ 1 分来标记。用户在正式使用其服务前, 需要对至少 15 部电影进行打分。当然, 用户打分的电影越多, 系统给出的推荐结果就会更趋于准确。MovieLens 的电影数据比较陈旧, 最开始让用户打分的电影基本上是 20 世纪八九十年代的, 因此, 用户需要有足够的耐心找出自己看过的, 或者上 IMDB 了解剧情以便真实反映自己的偏好。

MovieLens 数据源包括三个数据文件, 其中 ratings. dat 列表收集了 userID、movieID、rating、timestamp 四项内容, users. dat 列表收集了 userID、gender、age、occupation、zip-code 五项内容, movies. dat 列表收集了 movieID、title、genres 三项内容。表 1 ~ 3 给出了三个数据列表中的部分数据以供参考。

2.2 案例应用

这里随机抽取 MovieLens 中的 28 个用户和 18 部电影, 分别用 $c_1, c_2, \dots, c_{28}$ 和 $s_1, s_2, \dots, s_{18}$ 表示。结合 a) ~ e) 得到表 1 的用户/产品聚类组合表, 并得到七个聚类小组。

表 1 用户/产品聚类组合表

	s_{16}	s_{11}	s_7	s_{12}	s_3	s_6	s_4	s_{10}	s_{18}	s_9	s_2	s_5	s_{15}	s_{13}	s_1	s_8	s_{17}	s_{14}
c_{18}	1	0							-1					0			0	0
c_2	1	0								-1								
c_{27}	1		0								-1							
c_4	1			0					-1	0								0
c_5	0	1																
c_{24}		1	1	0												-1		
c_{28}	0		1											-1	0			
c_8		1	1	0							-1	-1	0					
c_{19}		0	1	1		0											-1	
c_{10}			0	0	1								-1	-1				
c_{21}			0	1	1	1					-1							
c_{20}	-1			0	0		1	1										
c_{13}			0	0	-1			1	1				0					
c_{26}			0	0	-1	0			0	1						-1		
c_{12}					0	0			0		1	0	-1			-1		
c_{16}			0		-1	0					1					-1		
c_{23}									0	-1	0		1					
c_1	-1		0											1		0		
c_{11}			0										-1	0	1	0		
c_{25}	-1													1	1		0	
c_7			0		-1								-1		1			
c_9			0					0							1			
c_{15}	-1	0				0							0			1	0	
c_6						0							-1		-1	1		
c_{14}	0	0							0				-1			1	0	
c_{22}		0									-1						1	
c_{17}	-1												-1			1		
c_3		0											-1				1	

结合步骤 f) ~ g) 计算社区密度, 如表 2 所示。

表 2 用户社区密度表

	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
D_i	0	1.167	0.690	0.467	0.933	0.741	0.75	0.31	0.529	0.25
	c_{11}	c_{12}	c_{13}	c_{14}	c_{15}	c_{16}	c_{17}	c_{18}	c_{19}	c_{20}
D_i	1.256	0.667	0.5	0.645	0.588	0.571	0.870	0.467	0.346	0.578
	c_{21}	c_{22}	c_{23}	c_{24}	c_{25}	c_{26}	c_{27}	c_{28}		
D_i	0.188	0.8	0	0.333	1.125	0.381	0.467	0.529		

在每一个社区内选择 $y_q = \max D_u, u = 1, 2, \dots, t$, 且 $y_q \leq 1$, 则各个聚类中心 y_q 依次为: $y_1 = 0.933, y_2 = 0.529, y_3 = 0.25, y_4 = 0.667, y_5 = 0.75, y_6 = 0.87$ 。

由 h) 得到 $S(U, K) = 399.6846$ 。

若调整社区的结构, 可以得到不同的 $S(U, K)$ 值, 如表 3 所示。

表 3 各种社区结构的紧致与分离性效果函数

社区划分结果	$S(U, K)$
$\{c_{18}, c_2, c_{27}, c_4, c_5\} \{c_{24}, c_{28}, c_8, c_{19}\}$ $\{c_{10}, c_{21}\} \{c_{20}, c_{13}, c_{26}, c_{12}, c_{16}\}$ $\{c_{23}, c_1, c_{11}, c_{25}, c_7, c_9\}$ $\{c_{15}, c_6, c_{14}, c_{22}, c_{17}, c_3\}$	399.6846
$\{c_{18}, c_2, c_{27}, c_4, c_5\} \{c_{24}, c_{28}, c_8, c_{19}, c_{10}\}$ $\{c_{21}, c_{20}, c_{13}, c_{26}, c_{12}, c_{16}\}$ $\{c_{23}, c_1, c_{11}, c_{25}, c_7, c_9\}$ $\{c_{15}, c_6, c_{14}, c_{22}, c_{17}, c_3\}$	397.7656
$\{c_{10}, c_2, c_{27}, c_7, c_5\} \{c_{24}, c_{20}, c_8, c_{19}, c_{13}\}$ $\{c_{21}, c_{15}, c_{10}, c_{26}, c_{11}, c_{16}\}$ $\{c_{23}, c_1, c_{17}, c_{25}, c_3, c_9\} \{c_{28}, c_6, c_{14}, c_{22}\}$ $\{c_{12}, c_4\}$	189.7743
$\{c_{16}, c_5, c_{15}, c_{22}, c_{18}\} \{c_8, c_2, c_{27}, c_4, c_9\}$ $\{c_{25}, c_{14}, c_{26}, c_{12}, c_{17}\}$ $\{c_{24}, c_{28}, c_7, c_{13}, c_{10}\} \{c_{23}, c_1, c_{19}, c_6, c_{21}\}$ $\{c_3, c_{20}, c_{11}\}$	301.5098

通过比较可以看到, 文中介绍的方法得到的 $S(U, K)$ 值最大, 说明该方法得到的社区划分为最优聚类结果, 该方法在一定程度上有效。

3 结束语

稀疏矩阵一直是困扰推荐研究者的一个难题, 本文借助社区网络的思想, 通过矩阵调整有效地约简了稀疏矩阵, 在保证覆盖率和精确率的情况下得到了一种新的用户聚类分析方法。社区结构思想的引入不仅可以准确地描述问题, 而且对实际问题有一定的简化, 降低了计算复杂度。

采用 MovieLens 的数据进行验证, 不仅具有一定的代表性, 同时也增强了问题说服力, 但是数据采集量偏少, 因此存在一定的局限性。

本文采用紧致与分离性效果函数进行聚类质量的评价, 实质上还可以采用分离系数 $F(U, K) = \frac{1}{n} \sum_{i=1}^K \sum_{j=1}^n (u_{ij})^2$ 、分离熵 $H(U, K) = -\frac{1}{n} \sum_{i=1}^K \sum_{j=1}^n u_{ij} \log(u_{ij})$ 等方法进行聚类质量评价。虽然紧致与分离性效果函数的评价效果相对较好, 但是在推荐系统中各种评价函数的适用性是一个值得探讨的问题。

文中定义 $\left\{s_i: R'_{ui} \geq \varepsilon, \text{i. e. } R_{ui} \geq \frac{\max R_u + \min R_u}{2}\right\}$ 为用户喜欢的产品集合, 并以特殊情况 $\varepsilon = 0.5$ 进行案例分析。实质上, 这里 ε 的取值方法是一个值得研究的问题。

参考文献:

[1] 刘建国, 周涛, 郭强, 等. 个性化推荐系统评价方法综述[J]. 复杂系统与复杂性科学, 2009, 6(3): 1-12.
[2] 李宇澄. 协同过滤算法研究[D]. 上海: 复旦大学, 2005.
[3] PAZZANI M. A framework for collaborative, content-based, and demographic filtering[J]. Artificial Intelligence Review, 1999, 13(5-6): 393-408.
[4] 冀俊忠. 贝叶斯网及其在网上智能中的应用研究[D]. 北京: 北京工业大学, 2004.
[5] 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.
[6] 崔春生, 吴新宗. 基于 Vague 集的内容推荐算法研究[J]. 计算机应用研究, 2010, 27(6): 2108-2110.
[7] 史忠植. 知识发现[M]. 北京: 清华大学, 2005.