DOI:10.16208/j.issn1000-7024.2014.12.025

聚类与协同过滤相结合的隐式推荐系统

许 伟，段 富

（太原理工大学 计算机科学与技术学院，山西 太原 030024）

摘 要：针对目前大多推荐系统中使用的协同过滤算法都需要有显示的用户反馈的问题，提出一种在隐式反馈推荐系统中使用聚类与矩阵分解技术相结合的方法，为用户提供更好地推荐结果。其结果是由基于用户历史购买记录的隐式反馈产生的，不需任何显式反馈提供的数据。采用高维的、无参数的分裂层次聚类技术产生聚类结果，根据聚类的结果为每个用户提供高兴趣度的个性化推荐。实验结果表明，在隐式反馈的情况下该方法也能有效获得用户偏好，产生大量的高准确度推荐。

关键词：推荐系统；协同过滤；聚类；隐式反馈；分裂层次聚类算法

中图法分类号：TP301.6 **文献标识号**：A **文章编号**：1000-7024（2014）12-4181-05

Combining clustering and collaborative filtering for implicit recommender system

XU Wei, DUAN Fu

(College of Computer Science and Technology, Taiyuan University of Technology, Taiyuan 030024, China)

Abstract: Aiming at the problem that most collaborative filtering algorithms require explicit user feedback, a combination method of clustering and matrix decomposition in implicit feedback was proposed to provide users with better recommendation results, and the results were generated using only implicit feedback based on users' purchase history without requiring any parameters from explicit feedback. A high dimensional, parameter-free, divisive hierarchical clustering technique was used to produce clustering results and personalized recommendations were provided to users based on the clustering results. Finally, experimental results demonstrate effective user preference can be obtained and a high percentage of recommendation with high ratings can be generated while using only implicit feedback through this method.

Key words: recommender systems; collaborative filtering; clustering; implicit feedback; DHCC

0 引 言

目前，虽然已经有很多基于协同过滤的显式反馈推荐系统，但是开发推荐系统的非显式反馈技术仍不完善。尽管已经有一些推荐系统采用了隐式反馈推荐技术，如基于人体标签[1]的社交网络推荐系统和基于概率模型[2]与语义模型[3]的新闻推荐系统，但很少有不利用评价机制进行推荐的系统，而且这些隐式反馈技术通常都要求设置一些参数。

聚类技术可以有效挖掘出具有相似行为结构的用户，并按照其行为结构的相似程度对用户进行分组，即在同一组的用户必定有相似的兴趣偏好。因此，我们可以根据这种行为结构（即用户组的产品偏好）为组内的用户做推荐。但是，由于有大量的产品需要被推荐，而用户的行为结构仅仅定义了用户组对全部产品中一小部分产品的偏好，因此，挖掘用户组和每个组的行为结构是非常困难的。

本文提出了一种高维的、无参数的分裂层次聚类技术，它只需要通过基于用户历史购买记录的隐式反馈就可以挖掘出同一簇中用户之间的关系。使用矩阵分解的方法就可以挖掘出潜在的信息，以便找到用户之间的相似性。此推荐模型已经通过了著名的 Movielens 数据集的验证。

1 隐式反馈面临的问题

在使用隐式反馈时会遇到一些问题，这些问题已经在文献[4]中做了详细的说明，为便于理解本文也做了一些总结。

（1）缺乏用户反馈

在显式反馈推荐系统中，用户会对自己喜欢的产品进行评价，进而系统可以很明确的知道用户的喜好。而在隐式反馈推荐系统中，系统却几乎不能提供明确的与用户偏好有关

收稿日期：2013-12-24；修订日期：2014-03-06

基金项目：山西省科技基础条件平台计划基金项目（2012091003-0103）；山西省卫生厅科技攻关计划基金项目（2011119）

作者简介：许伟（1983-），男，山西太原人，硕士，研究方向为软件开发环境与工具、软件项目管理、软件设计及测试技术；段富（1958-），男，山西大同人，博士研究生，教授，研究方向为智能优化算法及应用、软件理论与算法、信息系统集成及安全、软件开发环境与工具、软件项目管理、软件设计及测试技术。E-mail：xuweiluzi@163.com

的信息，而且对于用户不购买产品的原因也一无所知。如用户可能对这个产品不感兴趣，因为产品的价格太高，或是产品的详细信息没有及时显示出来，用户没有看到等。

另一方面，通过查看用户的购买历史记录，我们可以挖掘出用户比较频繁的行为操作，如比较偏好某类产品。根据这些频繁的行为操作就可以确定用户对哪类产品比较感兴趣。比如，一个用户购买了很多动作电影和少量恐怖电影，即使没有任何显示反馈我们也可以确定该用户对动作电影比较感兴趣。

（2）数据的一致性

如果没有显示反馈，很难区分用户购买的产品是自己喜欢的还是为别人购买的（其实是别人喜欢的）。在很多情况下，用户可能给某人买一个礼物，也可能为家人租一部影碟等等。如果没有考虑到这个问题，就可能产生一些不准确地信息，给推荐带来了负作用，降低了推荐的准确性。

（3）置信度与偏好

在隐式反馈中，如果一个用户多次购买了某类产品，我们就可以获得用户对这类产品喜好的置信度信息。如果用户对某类产品只购买了一次，则可能有很多购买原因，但是如果用户进行了“多次”购买，我们就可以获得更多的用户偏好信息。所以，我们应该更加注重“多次”购买，而不是“一次”购买。在显式反馈中，只要用户显式评价了某类产品我们就可以知道用户对这类产品是否有兴趣和用户的偏好。就算是用户购买了“一次”某类产品，通过显式评价也可以为推荐提供有意义的用户偏好信息。

（4）需要重新定义评价机制

在显式反馈推荐系统中有很多我们熟知的评价机制。如推荐值与实际评价值的均方根误差（RMSE）。而在隐式推荐中，我们并不清楚用户没有购买产品的原因，事实上，用户购买产品的原因我们也不清楚。因此，我们需要一种方法来挖掘出用户间的相似性并且区分出用户对哪些产品更感兴趣和对哪些产品比较感兴趣。

本文将重点研究以下两方面的问题：①在没有显示反馈的情况下发现用户之间的相似关系；②推荐给用户他们感兴趣的产品。

2 相关工作

近年来，推荐系统日益增多，但是与基于显式反馈的推荐系统相比隐式反馈推荐系统相对较少。

（1）关联规则

基于关联规则的推荐系统是采用从购买历史中提取规则并发现重要关联的方式进行推荐的[5]。而利用这种方法常常会产生大量的规则，并且从这些规则中获取信息是非常低效的。此外，关联规则的方法还需要显式参数，如规则的数量和与规则相关联的置信水平。

有些基于关联规则的推荐系统是利用规则模板来建立的，虽然这样可以大大减轻了发现大量规则的工作，但是大量的规则降低了推荐的置信度这个问题依然存在[6]。

（2）潜在模型

在文献［4］中，作者建立了一个基于潜在模型的 TV 推荐系统来解决隐式反馈的问题。虽然利用这种方法产生了很好的效果，但是需要设置多个参数而且找出最佳参数也是非常耗时间的。在这个推荐系统中，作者是通过确定用户观看某电视节目的次数，然后根据这个次数与相应的权值建立偏好矩阵进行推荐的。而一般情况下，对于同一个产品用户只购买一次。

（3）协同过滤

协同过滤中运用最广泛的当属 K-最近邻居算法（KNN）。文献［7］中详细阐述了如何运用基于邻居的推荐算法进行推荐项预测和评价预测；而文献［8］中则实现了加权用户和加权推荐项的 KNN 算法。他们都在文章中利用余弦相似性算法来找出用户之间或推荐项之间的相似性。虽然 KNN 算法得到了很好的应用，但这个算法常常依赖于显式的关系信息，在潜在相似的情况下并不适用。

在文献［9］中，作者基于隐式反馈的协同过滤算法实现了一个销售家装产品的推荐系统。开发这个推荐系统主要是让专业人士（这里是销售员）使用的，系统可以为他们推荐几类适合的产品以供参考。由于这个推荐系统是给一些专业人士使用的或是为了提供几类产品做参考，所以不能把这个系统直接应用在一般的基于购买历史推荐的情况中。

3 聚类与协同过滤算法

为了发现用户之间偏好的潜在关系，本文提出了高维的、无参数的分裂层次聚类技术（DHCC）来解决分类数据的问题。聚类产品购买中的隐式反馈数据（分类数据）要比聚类数值数据复杂，因为找不到有效的方法来计算分类数据之间的相似性。DHCC 算法避开了计算一对一的相似性，提出了计算一个数据与一组数据之间的相似性的方法，并且通过分离优化的方法对这些数据进行聚类。

从用户的购买记录中我们可以挖掘出他们都比较感兴趣的产品，在这里称其为受欢迎的产品。那么假设受欢迎的产品往往可作为用户首选的产品。那么通过对有相似兴趣的用户分组我们就有可能为用户推荐他们较感兴趣的产品。应用到隐式反馈的购买数据上，DHCC 算法将通过构建二叉树的方法来发现相似的用户组。从这棵树中，我们可以提取出每个簇中的用户最可能首选的那些产品。通过这种技术可以很好的为每个簇中的用户进行个性化推荐。

3.1 聚 类

在推荐系统中运用聚类算法会有很多问题。首先，基于隐式反馈的购买数据属于分类数据，并且没有一个明确的方法可以用来计算分类数据之间的相似度。比如用户 A 可能与用户 B 相似（他们购买了相似的产品-动作片），用

户 B 又与用户 C 相似（他们购买了相似的产品-恐怖片），也不能认为用户 A 与用户 C 也是相似的，他们有可能不是相似的。因为根据例子我们只能推断出用户 A 可能喜欢看动作片，用户 B 可能既喜欢看动作片又喜欢看恐怖片，而用户 C 可能喜欢看恐怖片，并不能说用户 A 也喜欢看恐怖片或用户 C 也喜欢看动作片，即不能确定用户 A 与用户 C 是否相似。其次，一个网店中往往会有大量的产品，那么就需要一种聚类技术来处理高维数据。第三，由于我们希望创建一个通用的方法，因此，这个聚类技术应该能够自动识别簇的数目，而且不需要设置参数或者设置尽量少的参数。此外，也没有一个方法可以用来获取用户对产品的评价信息或是评分信息，而仅仅能获取一个只包含用户是否购买了产品这两种可能的矩阵。

DHCC 算法[10]在聚类分类数据方面无疑是一种有效的而且是高效的分裂层次算法，能够自动识别簇的数量，且能够很好的处理高维数据和相似度计算的问题。这种方法可以较准确的对用户进行分类，正如一个簇中的用户有相似的兴趣偏好，通过相似的兴趣偏好就可以向同簇的其他用户做出推荐。DHCC 算法更详细的内容请参考文献［10］。

为了应用 DHCC 算法，首先创建一个包含每个用户购买历史所有信息的模型，然后提取每个产品的特征值，进而就可以建立一个用户与产品的二维矩阵来表示用户对产品的购买情况，见表 1。在 DHCC 算法中，每个用户与簇之间的相似度可以用卡方距离来计算，计算公式如式（1）所示。

表 1　用户×产品矩阵

用户	产品 a		产品 b	
	买	没买	买	没买
用户 A	1	0	0	1
用户 B	0	1	1	0
用户 C	1	0	0	1

$$d_{Chi}(u_i, C_k) = \sum_j \frac{(u_{ij} - c_{kj})^2}{c_{kj}} \tag{1}$$

式中：u_i——用户向量；c_k——簇 k 的簇中心向量；C_k——簇中分类值频度的平方根，相当于簇 C_k 的行为结构。

下面将举例说明运用 DHCC 算法对用户进行聚类的过程。如图 1 所示，每个英文字母表示一个用户，每个叶子结点表示一个簇。如果完成了聚类就将形成一棵二叉树，而其中的每个叶子结点就是一个簇（此例中有 5 个簇）。与树中的其它结点相比，我们更加关心叶子结点，主要是因为叶子结点更容易反映出每个用户组（簇）之间的不同。

表 2 中列出了文章中用到的其它变量的定义。其中比较重要的是：P_c 表示簇 c 中的用户比较感兴趣（受欢迎）的产品列表；$P_{c,n}$ 中的 n 表示感兴趣的产品数量；U_c 表示执行 DHCC 算法后形成的簇 c 中的用户。

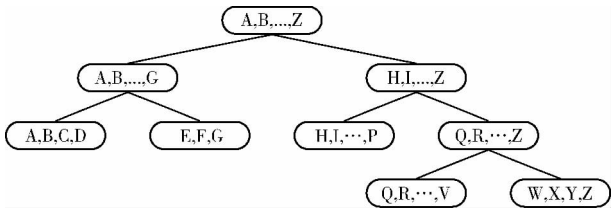

图 1　DHCC 算法的二叉树聚类模型

表 2　相关变量描述

变量	描述
n	受用户欢迎的产品列表中的产品数量
C	不同簇的集合
C_k	具体的簇 k
c_k	C_k 的中心向量
U_c	簇 c 中的用户集合
I_c	簇 c 中的产品集合
S_k	第 k 个用户子集
R_u	用户 u 的推荐集合
B_u	用户 u 购买的产品集合
P_c	受簇 c 中的用户欢迎的产品列表
$P_{c,n}$	受簇 c 中的用户欢迎的产品列表，但限制列表中只能有 n 个产品
$r_{u,i}$	用户 u 对产品 i 的评价值
$er_{u,i}$	用户 u 对产品 i 的估计评价值
$S_{u,i}$	根据 $r_{u,i}$ 是否存在来决定取值是 $r_{u,i}$ 还是 $er_{u,i}$
t	事先考虑好的阀值

3.2　推 荐

为了完成推荐，首先我们假设比较受用户欢迎的产品会有一个好的评分，事实上受到用户欢迎的产品先前可能通过一些外部手段对其进行了推广，比如朋友告知或是做了一些广告等等。在这个假设的基础上，我们首先对 P_c 中的 n 个产品进行降序排序。然后，对簇 C 中的每个用户 $u \in U_c$，建立推荐列表 R_u。这个推荐列表是基于 P_c 和 B_u 的，定义如式（2）所示。由于 P_c 已经排过序，我们就可以根据前 n 个产品元素创建 Top-N 列表[11]

$$R_u = P_c - B_u \tag{2}$$

我们用 Top-N 与邻居推荐相结合的混合推荐方法，解决了第一部分隐式反馈面临的问题所述的数据一致性问题。即使用户 A 为别人 B 购买了一个产品，也将 B 看成与 A 有相似偏好的邻居用户，并为其推荐兴趣度较高的产品。一般情况下，用户为了得到与自己偏好相符的推荐，大部分用户在购买产品时都是根据自己的兴趣偏好进行购买的。

簇的行为结构在推荐系统中起着重要的作用。根据用

户的购买行为而形成的行为结构，即受欢迎的产品，使用户被聚类在一个簇中。但是，值得注意的是受欢迎的产品是仅限于簇中的用户这个范围而不是所有的用户，即受欢迎的产品是受某个簇中的用户欢迎，而不是受所有用户（所有簇中的用户）的欢迎。尽管每个用户的购买历史只涵盖了一部分行为结构，我们也能为那些购买历史涵盖了尽量多行为结构的用户做出推荐。

4 实验结果

基于上述隐式反馈的假设在下面的实验中产生了很好的效果。以下将采用我们熟知的 MovieLens 数据集来验证这个假设和我们的推荐系统。MovieLens 数据集包含了 100KB 左右的评分（1-5）信息，包含了 943 个用户和 1682 个电影资源，共有 5 组随机划分的结果，每个用户至少有 20 条的评分记录。这些评分信息只用来验证我们的方法，不用来验证推荐。利用 MovieLens 数据集创建的 DHCC 算法所使用的用户×产品矩阵是一个 943×3364 的矩阵，型如表 1 所示。

4.1 假设验证

为了验证我们的假设：一个受欢迎的产品往往是值得推荐的产品，我们利用评分信息定义了一个验证标准，那就是用我们的不需要任何显示反馈信息的推荐系统产生的结果与用显示反馈信息的协同过滤推荐产生的结果做比较。事实上，使用我们的方法产生的推荐结果中很大一部分与协同过滤产生的高评分（4 或 5）推荐结果是一致的。

我们观察了 P_c 中 n 个最受欢迎产品的评分，发现这些评分可能是可用的初始数据，也可能是协同过滤算法计算出来的数据。准确的讲，如果用户给这个产品做了评分，则我们就直接用这个评分，如果没有，我们将使用此用户的邻居对这个产品的综合评分。由于协同过滤中使用的相似度取决于由先前 DHCC 算法创建的簇。因此，对于每个产品，我们将计算同一簇中的用户对该产品的平均评分。如式（3）所示，其中 u’是用户 u 所在簇的其他用户；$r_{u',i}$ 表示用户 u’对产品 i 的评分

$$er_{u,i}=\sum_{u\in U_c}\frac{r_{u,i}}{|U_c|} \tag{3}$$

不论是初始的评分还是后来计算的评分，都需要参考一个确定的阈值来进行推荐，就像在协同过滤算法中判断阈值 t 进行推荐一样。下面的式（5）说明了如何通过判断阈值 t 来决定是否把产品 i 推荐给用户 u。

$$s_{u,i}=\begin{cases}r_{u,i} & if \quad exist\\ er_{u,i} & otherwise\end{cases} \tag{4}$$

$$G_{u,i,t}=\begin{cases}1 & S_{u,i}\geqslant t\\ 0 & S_{u,i}<t\end{cases} \tag{5}$$

为了比较两个推荐系统所产生的推荐结果，我们利用式（5）得出的结果为推荐结果创建一个分值，即 $P_{c,n}$ 的分值。我们可以通过下面的式（6）计算出 $P_{c,n}$ 所得分值，它是根据 $P_{c,n}$ 中的每一项通过协同过滤算法为 U_c 中的每个用户做推荐所产生推荐结果的平均值。很明显，$P_{c,n}$ 的分值是随着影响协同过滤推荐结果的阈值 t 的变化而变化的

$$ratioH(P_{c,n,t})=\frac{\sum_{i\in P_{c,n}}\sum_{u\in U_c}G_{u,i,t}}{n\times|U_c|} \tag{6}$$

现在，我们可以为这个假设定义一个总分值。根据簇的个数对每一个簇的分值做了加权平均，得到了一个全局的比率，即总分值。如式（7）所示

$$GlobalRatioH(P_{c,n,t})=\frac{\sum_{c\in C}ratioH(P_{c,n})\times|U_c|}{\sum_{c\in C}|U_c|} \tag{7}$$

表 3 中显示了用 MovieLens 数据集验证的结果，通过设置不同的阈值 t 和 n 而得到相应的总分值。从表中我们可以看出，受欢迎产品列表中的 n 个项都获得了很高分值，可进行很好的推荐。当 $t=3$ 时，分值都在 93%以上，则表示这 n 个产品可以推荐给大多数用户。注意到当 $t=4$ 时，分值仍然很理想。由于 MovieLens 数据集只定义了 5 个评分级别，我们把 3 级定为平均分数，则如果评分级别高于这个平均分数，如 4 级或 5 级，就认为是非常好的。而且，从表 3 中可以看出推荐前 10 个产品的总分值就达到了 75%以上，推荐前 50 个产品的总分值也达到了 67%以上。因此，实验证实了仅仅在隐式反馈的情况下也能有效的获得用户的偏好，并能达到与显式反馈的协同过滤同水平的推荐效果。

表 3 不同阈值下的用户全局比率

	n=10	n=25	n=50
t=2	98.1548%	98.2821%	98.4687%
t=3	93.9873%	94.0318%	94.6257%
t=4	74.9417%	70.3033%	66.7805%

4.2 推荐验证

我们将基于用户采用 k 折交叉验证[12]的方法来验证我们的推荐系统，而且还为用户推荐他们以前没有见过的产品。与验证假设的方法类似，我们把用户的评分历史作为基础，根据用户对他们以前购买过的产品的评分，比较推荐系统能够推荐的产品与他们购买过的产品。如果发现在推荐的产品中有些产品用户没有购买过，那么对没有购买过的这些产品的评分就用协同过滤算法获得他的邻居用户的综合评分。

由于现有的验证技术基本都是用来验证显式反馈推荐的，并不能用于本文所述的推荐验证。因此，我们就需要找到另外一种验证技术，那就是 Nri/Nari 准确度：在所有的信息中能够发现多少相关的信息，是一个数量值。应用到本文中就是：在所有的推荐结果中，可以挖掘出多少更

值得推荐的结果，且可以通过 $G_{u,i,t}$ 获得。为了获得一个整体准确度，我们将对所有评分做平均来获得推荐的准确度。

为了验证推荐，首先把用户分割成 k 个子集 S_k，一个单独的子集 S_t 用来验证推荐，其余的 k-1 个子集 S_{k-1} 用来训练推荐。举个例子，如果 k=3，第 1 次推荐可以由第 3 个子集来验证，由第 1 个和第 2 个子集来训练；类似的，第 2 次推荐可以由第 1 个子集来验证，由第 2 个和第 3 个子集来训练。如果已经完成了 k 次交叉验证，那么就可以利用每次的验证子集来评估推荐

$$Precision = \frac{\sum_{u \in S_t} \sum_{i \in P_{c,n}} G_{u,i,t}}{n \times | S_t |} \tag{8}$$

要为 S_t 中的每一个用户（记为：S_{tk}）创建推荐就要知道他们对产品的评分。首先，应该挖掘出与 S_{tk} 有相近兴趣偏好的用户簇；然后，为 S_{tk} 提供簇中的每一个推荐结果；接着通过研究用户的购买历史，如果发现 S_{tk} 以前购买过这个产品，我们就使用 S_{tk} 对产品的现有评分；否则，就用基于邻居的协同过滤算法计算出的评分。我们使用了式（8）来获得高兴趣推荐的准确度，表 4 中则显示了不同推荐数量和不同阀值下的推荐准确度。

表 4　不同阀值下的用户推荐准确度

	m=5	m=10	m=20
t=2	97.1064%	97.766%	97.8351%
t=3	92.2766%	94.0532%	94.1702%
t=4	63.2553%	74.4255%	69.75%

从表 4 中我们可以看出，给验证用户提供的推荐中包含了大量的高准确度推荐。当推荐系统中新增了用户时，即 m=10 时，推荐的准确度达到了 74% 优于其平均值。另外要注意的是：如果把所有的验证集合都用上，推荐准确度可能会趋近于 1，进而产生大量的高准确度推荐。

5　结束语

目前，仅仅使用隐式反馈而且没有明确评分指标的推荐系统还很少见。虽然，我们可以给用户提供一些产品的推荐，但是，并不能保证推荐的产品用户一定喜欢。采用隐式数据可能会降低推荐的确定性，但是在许多推荐系统中，甚至在有显式评价的推荐系统中，仅仅有少数产品得到了用户的评分。

现在，我们能通过隐式数据预测出高兴趣的推荐，但是还有一些问题还需要在以后的工作中解决。第一个就是聚类算法，DHCC 的目的是在分类数据中找到潜在的结构。虽然在本文中产品看作分类数据来处理，但是为了进一步增加簇的准确性，我们还需要对 DHCC 算法进行优化。

另外，DHCC 算法是用二叉树来组织数据的，我们就需要研究每一对结点（簇）之间的依赖关系。一旦知道了依赖关系，我们就可以根据那些用户偏好相似但又不一样的结点提供推荐。通过这种方法，我们就可以为用户提供大量可能的推荐。但是，当增加新用户时，我们当前的方法只能通过重新构建簇来产生子簇，这就需要有一个更好的方法来实现簇的更新，而不是通过重建簇这样一种繁琐的方法。

参考文献：

[1] Guy I, Zwerdling N, Ronen I, et al. Social media recommendation based on people and tags [C] //Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2010: 194-201.

[2] Li M, Dias B, El-Deredy W, et al. A probabilistic model for item-based recommender systems [C] //Proceedings of the ACM Conference on Recommender Systems. ACM, 2007: 129-132.

[3] Li L, Wang D, Li T, et al. SCENE: A scalable two-stage personalized news recommendation system [C] //Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2011: 125-134.

[4] Hu Y, Koren Y, Volinsky C. Collaborative filtering for implicit feedback datasets [C] //Eighth IEEE International Conference on Data Mining, 2008: 263-272.

[5] Hongli G, Juntao L. The application of mining association rules in online shopping [C] //Fourth International Symposium on Computational Intelligence and Design, 2011: 208-210.

[6] Li Y, Zhong N. Ontology based web mining for information gathering [M]. Web Intelligence Meets Brain Informatics. Springer Berlin Heidelberg, 2007: 406-427.

[7] Desrosiers C, Karypis G. A comprehensive survey of neighborhood-based recommendation methods [M]. Recommender Systems Handbook. Springer US, 2011: 107-144.

[8] Gantner Z, Rendle S, Freudenthaler C, et al. MyMediaLite: A free recommender system library [C] //Proceedings of the Fifth ACM Conference on Recommender Systems. ACM, 2011: 305-308.

[9] Pradel B, Sean S, Delporte J, et al. A case study in a recommender system based on purchase data [C] //Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2011: 377-385.

[10] Xiong T, Wang S, Mayers A, et al. DHCC: Divisive hierarchical clustering of categorical data [J]. Data Mining and Knowledge Discovery, 2012, 24 (1): 103-135.

[11] Seyerlehner K, Flexer A, Widmer G. On the limitations of browsing top-N recommender systems [C] //Proceedings of the Third ACM Conference on Recommender Systems. ACM, 2009: 321-324.

[12] Simon R M, Subramanian J, Li M C, et al. Using cross-validation to evaluate predictive accuracy of survival risk classifiers based on high-dimensional data [J]. Briefings in Bioinformatics, 2011, 12 (3): 203-214.