数字资源云服务推荐系统设计

张龙昌,张洪锐
(渤海大学 信息科学与技术学院,辽宁 锦州 121000)

摘 要:随着云计算技术在数字资源整合中的应用越来越广泛,云服务推荐系统的用户数据集呈现指数级增长的趋势,因而对于数字资源云服务的使用者来说,从海量的数字资源中找到自己真正感兴趣的云服务变得愈加困难,亦即出现了“信息超载”的问题。为了解决“信息超载”问题,设计并实现了数字资源云服务推荐系统。该系统可快速发现满意度最高的云服务并推荐给目标用户,提高了大数据环境下的推荐速度。同时,为了提高数字资源云服务的推荐命中率,所构建的推荐系统采用改进的协同过滤算法,向云服务使用者(待推荐的目标用户)推荐具有相似兴趣行为用户访问过的满意度高的云服务。实验结果表明,数字资源云服务推荐系统可以提高推荐速度,准确地向用户推荐满意度高的数字资源云服务,能够快速提升用户体验。

关键词:数字资源云服务;信息超载;推荐系统;满意度
中图分类号:TP302 **文献标识码**:A **文章编号**:1673-629X(2017)08-0139-06
doi:10.3969/j.issn.1673-629X.2017.08.029

Design of Recommender System with Cloud Services of Digital Resource

ZHANG Long-chang, ZHANG Hong-rui
(College of Information Science and Technology, Bohai University, Jinzhou 121000, China)

Abstract: As cloud computing technology has been widely used for integrating digital resources, user data set of recommender system of cloud service shows the tendency of exponential growth. Therefore, it is difficult for users of cloud services with digital resource to find their interested cloud services from the huge amounts of digital resources, especially producing the problem of information overload. In order to solve problem of information overload, a recommender system with cloud services of digital resource is designed and implemented, which can quickly find cloud services with the highest satisfaction for target users and improve recommendatory speed in large data environment. Meanwhile, in order to improve hit rate in the process of recommending cloud services with digital resources, the recommendation system has employed improved collaborative filtering algorithm to recommend the favorite cloud services the users with similar interest accessed for target users. The experimental results show that recommender system with cloud services of digital resource has improved the recommendatory speed, and accurately recommended the user cloud services of digital resource with high satisfaction, which has lifted the user experience.

Key words: digital resource cloud services; information overload; recommender system; satisfaction

0 引 言

数字资源[1]是文献信息的表现形式之一,是将计算机技术、通信技术及多媒体技术相互融合而形成的以数字形式发布、存取、利用的信息资源总和。商业化的数据库、机构或个人建立的数据库、各种网络免费资源等都属于数字资源。由于云计算具有资源分配动态化、需求服务自助化、网络访问便捷化、服务可计量化、资源虚拟化以及具有可共享及可大幅度降低成本的特点,受到国内外研究者的高度关注。其中数字图书馆应用云计算技术实现资源整合最为突出[2]。随着云计算技术的发展与应用,国内外应用云计算技术的整合大型的提供数字资源服务的系统不断涌现。

随着数字资源云服务的种类和数量规模不断增加,在给数字资源使用者带来便利的同时,也使用户面

收稿日期:2016-05-22 **修回日期**:2016-08-25 **网络出版时间**:2017-07-05
基金项目:教育部人文社会科学研究青年基金(15YJC870028);辽宁省教育科学技术研究一般项目(L2014451);辽宁省自然科学基金(2015020009);辽宁省社会科学规划基金(L15BTQ002)
作者简介:张龙昌(1977-),男,博士,研究方向为服务计算、云计算、物联网;张洪锐(1989-),男,硕士研究生,研究方向为服务计算、云计算、物联网。

网络出版地址:http://kns.cnki.net/kcms/detail/61.1450.TP.20170705.1649.014.html

对庞大的数字资源无所适从。针对这种状况,采用数字资源云服务推荐系统是一种有效的解决方案。为此,设计并实现了数字资源云服务推荐系统,运用基于 R 树的协同过滤算法查询 top-n 个最相似用户,计算这 n 个用户访问过的数字资源云服务的综合满意度,向待推荐目标用户推荐满意度最高的数字资源云服务即可。

1 系统结构

数字资源云服务推荐系统从系统功能角度划分为五个模块[3-4](见图 1):Web 前端组件模块、数据预处理模块、数据存储模块、推荐模块和数据更新模块。Web 前端组建模块主要用于收集用户的数据(包含用户隐式反馈的兴趣行为和显式反馈的用户评分);数据存储模块有文件存储(日志文件)和数据库等存储方式管理数据[5];数据预处理模块主要是对用户隐式反馈的数据进行清理、去冗余等操作[6];推荐模块则选用推荐算法向用户进行个性化云服务推荐[7];数据更新模块是随着用户兴趣的变化进行数据更新[8]。

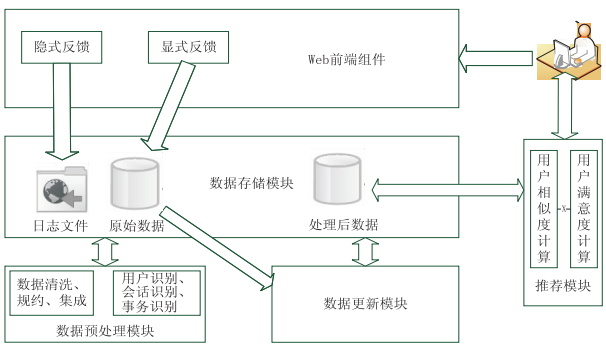

图 1 推荐系统结构

随着云计算技术的发展,客户端等软件逐渐会被淘汰,SaaS(软件即服务)越来越多地基于浏览器以插件的形式提供云服务。Web 前端组件模块主要是从浏览器及各 SaaS 服务供应商服务器中收集用户数据,不作介绍,重点介绍其他四个模块。

2 数据预处理模块

数据预处理工作在 Web 日志挖掘中具有基石的作用,高效正确的预处理方法关乎数据挖掘的成败,基于正确的有意义的历史数据做出的推荐算法才能有效地、正确地推送用户喜好的服务[9]。用户的历史行为数据和用户评分是研究推荐算法的关键资源,具体分为两类:显式反馈和隐式反馈[10]。显式反馈是用户对购买、使用的云服务做出评分等主观感受反馈,也就是数字资源云服务用户满意度模型计算所需的预期值和感知值;隐式反馈则是用户在浏览、购买云服务时产生的一系列客观行为,如收藏云服务、点击链接等,也就是数据资源云平台用户兴趣行为。

2.1 用户访问树

数据预处理的一个重要工作是从日志文件中准确地识别出访问数字资源云服务的用户,重建用户会话过程(即用户在购买使用云服务进行的一系列行为序列),其过程通常包括用户识别、会话识别、路径补充等。用户是指通过一个浏览器或者客户端访问一个或多个服务器的个体。由于缓存、代理服务器和防火墙的使用,导致准确识别用户及其事务很复杂。目前常用基于日志/站点、拓扑结构的办法进行识别,然而如果网站的拓扑结构比较复杂,在根据拓扑结构识别页与页之间的关系时,效率就会降低[11]。故采用了基于用户访问树的用户识别方法识别用户及其事务。

用户访问树采用孩子链表表示法,即用一组连续的空间来存储树上的节点,同时在每个节点上附加一个指针指向由其孩子节点构成的单链表。这种表示法找孩子节点比较容易,只要搜索 firstChild 指针指向的链表即可。其类型定义的存储结构如下:

```
typedef struct CTNode
{int child; //节点的序号
struct CTNode * next; //指向同一层的下一个节点
}CTNode, * ChildPtr;
typedef struct
{TelemType data; //节点存储的数据类型
ChildPtr firstChild; //孩子链表的头指针
}CTBox;
typedef struct
{unsigned char * url; //用户访问的 url 链接或者是表示页面或者云服务唯一的标识符
int * userBehave; //用户在 url 产生的兴趣行为
}TelemType;
typedef struct
{CTBox nodes[MAX_TREE_SIZE]; //MAX_TREE_SIZE 最大节点数
int n; //n 为节点总数
}CTree;
```

一个页面可以超链接到多个页面,按照当前用户访问的顺序依次构造用户访问树。当前用户访问一个链接(URL),遍历用户访问树,依据日志里参考页的信息,如果树中有链接(oldURL)可以链接到 URL,那么就将此链接的用户隐式反馈的相关信息(即为上述存储结构中的 TelemType)存储到 oldURL 所在节点下;如果树中没有链接可以链接到 URL,那么就重新产生一棵新树。

2.2 数据预处理流程

数据预处理通常分为数据清理、数据集成、数据变换和数据规约四步。由于一个用户的兴趣随着时间可能发生改变,所以用户识别、会话识别对于数据更新维

护、保持数据的时效性尤为重要。因此,在数据预处理流程中,用户识别、会话识别和事务识别同样必不可少。下面依次按照数据预处理流程(见图 2)对显式与隐式反馈的数据进行预处理操作[12]。

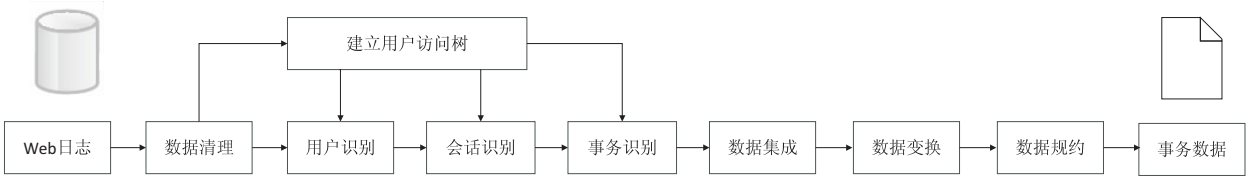

图 2　数据预处理流程

(1)数据清理。

数据清理主要包括遗漏数据填补、噪声数据处理、剔除无用数据等。

遗漏数据填补:简单的处理方式一般有采用默认值、平均值等对遗漏数据进行填补。但是这种方法可能会影响数据挖掘的效果。另外,复杂一点的方法是通过回归分析、贝叶斯方法或决策树推断最可能的值,这类方法充分利用历史数据,做到有理有据,因而效果最好,但复杂性较大。通常数据量比较大时,遗漏数据缺失值较多,通常也可以采用删除此条记录的方式进行处理,这样处理往往简单有效。

噪声数据处理:噪声是指收集变量信息时的随机错误或误差,包括错误的值或者偏离期望的鼓励点。通常采用的技术包括分箱方法、聚类、回归方法。

剔除无用数据:Web 日志挖掘中,日志文件存储着各式各样的文件,例如 gif、jpg 等图片文件与需要收集的变量值无关时,需要剔除这些文件,统一整理出新的文件内容。

(2)用户识别。

如何区分是否为同一个用户? 如果没有登录账户的话,不同的 ip 肯定不能算作同一个用户,相同的 ip,不同的用户代理 agent 也不属于同一个用户。那么相同的 ip 和相同的 agent 就一定属于同一个用户吗? 也不一定。用户访问树就用于划分不同的用户,因为一个用户当前可能停留在某一个链接上,然后,点击百度推送内容中进入另一个链接,这两个链接之间没有任何关系,那么暂时就把它们划分为不同的用户。这样做是为了保持用户访问树较小的原子性,有助于以后数据更好地关联。

(3)会话识别。

按照用户访问树的构造,一棵用户访问树对应一个用户访问过的链接请求等信息。当这棵用户访问树上页面请求的时间跨度比较大时,超过一定的阈值,有可能就是一个用户多次访问的同一个云服务应用或网站,这时,就可以识别同一个用户访问同一个服务的多次会话信息,并且清楚地知道最新访问的信息记录。能够区分出最新的用户反馈信息,在数据更新模块起着至关重要的作用。

(4)事务识别。

数据预处理中的事务,通常指的是页面及其表示的集合。按照 Web 使用记录挖掘任务来看,页面分两种类型:内容页与导航页。内容页是用户需求的信息所在的页面;导航页是用于指导用户搜索信息的页面。根据挖掘任务的不同,事务可以表示为一个页,也可以表示为一系列页的序列[13]。

用户访问树记录了任一个用户每一次会话的事务,所以事务识别不再是问题。只要确定某一个用户的用户访问树,按照时间阈值识别会话,就可以获取想要的事务。用户访问树中每个节点记录的不仅仅是云服务应用的标识或 URL,还有一些附属信息,即用户兴趣行为信息。

事务识别在所设计开发的推荐系统模型中的意义和承担的工作远不止于此。推荐系统结构中采集用户信息模块是 Web 前端组件模块,共分为两部分,一部分是隐式信息采集,也就是用户访问树中记录的信息集,通过事务识别,构建用户兴趣行为的数据模型;另一部分则是用户显式信息采集,也就是采集、计算获取用户满意度的工作也在事务识别时做,主要把获取到的用户满意度集成到数据存储模块。事务识别部分是区分用户、会话和事务的最清晰明了的地方,也是显式反馈和隐式反馈信息处理最方便的地方。用户兴趣行为数据和用户满意度在事务识别过程中同步处理。

(5)数据集成。

数据集成是将多个数据源中的数据结合起来并统一存储,建立数据仓库的过程实际上就是数据集成。其实,从用户识别到事务识别这个过程也是数据集成的过程,另外还包括 Web 日志文件集成,是把浏览器、客户端和服务器上的缓存、日志等文件进行数据集成。

(6)数据变换。

通过平滑聚集、数据概化、规范化等方式,将数据转换成适用于数据挖掘的形式。这一点对于基于距离的挖掘算法特别重要。

下面介绍一种数据变换的形式:设 $\boldsymbol{Y}=(y)_{m\times n}$ 为 m 个用户表现的 n 种兴趣行为的决策矩阵,y_{ij} 表示第 i 个用户的第 j 种行为的数据采集值。令 $\boldsymbol{Z}=(z)_{m\times n}$ 为进行数据标准化后的矩阵,同理,z_{ij} 表示第 i 个用户的

第 j 个行为的数据标准化后的值。$\boldsymbol{Y}=(y)_{m\times n}$ 转化成 $\boldsymbol{Z}=(z)_{m\times n}$ 如下所示：

$$z_{ij}=k\left(\frac{y_{ij}-y_j^{\min}}{y_j^{\max}-y_j^{\min}}\right) \tag{1}$$

其中，$y_j^{\min}$ 和 $y_j^{\max}$ 分别是 $\boldsymbol{Y}$ 中 j 列的最小值和最大值，也就是这 m 个用户中第 n 种用户兴趣行为的最小值和最大值；k 是各个属性的权重比例，在此取值为 1。

(7)数据规约。

一般意义上的数据规约是由于数据量非常大，进行数据挖掘时需要耗费很长的时间，严重影响了时间性能，所以应用一些数据规约技术得到数据集的规约表示，大大缩小了数据量，这部分数据仍然保持着原数据的完整性。采用基于 R 树的协同过滤算法(R_CF)，R 树是以用户之间的相似度进行构造，通过这种索引结构，可以大大缩减数据集进行协同过滤计算，大大提高时间性能，和数据规约有相同的目的和效果。

3 数据存储模块

数据存储模块主要有文件存储和数据库存储，文件存储主要是从服务器端、客户端和浏览器收集的原始数据集，以文件形式存储，和通常的 Web 日志存储格式基本相同。主要描述一下数据库存储及其存储数据的必不可少的几张表的设计及索引方面的工作。

3.1 Oracle Spatial 空间索引

Oracle Spatial 是 Oracle 的空间数据操作开发包，用来存储、管理、查询空间数据，提供了一套 SQL 方案和函数，用来存储、检索、更新和查询数据库中的空间要素集合[14]。主要由几何数据类型，空间索引机制，一套操作函数，管理工具组成。Oracle 支持自定义的数据类型，可以用数组、结构体或带有构造函数、功能函数的类来定义自己的对象类型。这样的对象类型能用于属性列的数据类型，也能用来创建对象表。而 Oracle Spatial 也正是基于该特性研发的一套空间数据处理系统。下面是空间索引的相关语法：

(1)空间索引的创建。

create index <INDEX_NAME> on <TABLE_NAME>(<COLUMNNAME>)；

Indextype is mdsys. spatial_index；

为了在表的字段上创建空间索引，应当始终指定 INDEXTYPE 为 mdsys. spatial_index。空间索引表存储在 SDO_INDEX_TABLE 字段中，总是以 MDRT 开头。

(2)空间索引的参数。

create index <INDEX_NAME> on <TABLE_NAME>(<COLUMNNAME>)；

Indextype is mdsys. spatial_index；

parameters('PARAMETER_STRING')；

PARAMETER_STRING 参数可以设置的变量主要有六个，分别为 tablespace、work _ tablespace、layer _ gtype、sdo_index_dims、sdo_dml_batch_size 和 sdo_level。

tablespace 用于指定哪个表空间来存储空间索引表。例如，parameters('TABLESPACE = gmapdata')是指定 gmapdata 表空间来存储索引表。

work_tablespace 用于指定工作表空间。在索引创建过程中，R-tree 索引会在整个数据集上执行排序操作，因此会产生一些工作表。不过这些工作表在索引创建过程结束时会被删除，会产生很多表空间碎片。设置 work_tablespace 参数，使其指定一个单独的表空间，就可以避免这种情况的发生。

layer_gtype 指定了索引列的几何数据为特定类型的几何体，有助于加快查询操作符的执行速度。

sdo_index_dims 指定空间索引维数，默认为 2。

sdo_dml_batch_size 用于指定一个事务中批量插入/删除/更新时的批量大小(对有大量插入的事务，该参数应设为 5 000 或 10 000)。默认为 1 000。

sdo_level 用于指定是创建 R-tree 索引还是四叉树索引。默认是 R-tree。

基于 Oracle Spatial 就可以实现对多维空间的用户兴趣行为数据集的存储。其中，每一个云服务的所有用户隐式的反馈用户兴趣行为集建立一棵 R-tree 索引，基于 R 树的协同过滤推荐算法就可以通过 R-tree 索引结构找到 top-n 个相似度最高的用户，然后根据这 top-n 个用户访问其他数字资源云服务的用户满意度，选择最满意的云服务向用户推荐。

3.2 数据库表设计

数据库表包括用户信息表(User)、云服务信息表(CloudService)、用户满意度表(UserSatisfaction)、用户兴趣行为信息表(UserInterestBehave)。用户信息表是用户通常的基本信息，这里主要对其他三张表的结构及关联关系进行描述。

(1)云服务信息表。

该表内的数据是数字资源云平台提供的云服务的信息列表，表名为 CloudService。ID 同样也是顺序自动生成。主要信息如表 1 所示。

(2)用户满意度表。

该表主要存储用户对数字资源云平台提供的云服务的满意度的信息，表名为 UserSatisfaction。用户对访问、购买或者使用过的数字资源云服务进行主观评分后，获取用户满意度。此部分工作是在数据预处理模块(2.2 节中的事务识别)完成。表结构见表 2。其中，USER_BEHAVE_ID 是用户兴趣行为表的 id，用户满意度表是由 USER_BEHAVE_ID 字段进行空间索引的，该字段关联的是用户兴趣行为表(UserInterestBe-

have)的主键。

(3)用户兴趣行为表。

该表表名为 UserInterestBehave,用来收集记录用户兴趣行为的信息,如收藏标签、删除标签和点击链接等操作。经过数据预处理后,这几个行为动作构成一个空间向量,此表就存储用户的这些行为空间向量。为简单处理,只列出如下三个维度,如表 3 所示。

表 1　云服务信息表结构

主键	名称	数据类型	可空	备注
是	ID	NUMBER(19)	否	云服务 id
否	SERVICE_PROVIDER	NVARCHAR(30)	否	云服务提供商
否	SERVICE_NAME	NVARCHAR(30)	否	云服务名称
否	SERVICE_URL	VARCHAR(40)	否	请求云服务的 url
否	REMARK	VARCHAR(240)	是	备注

表 2　用户满意度表结构

主键	名称	数据类型	可空	备注
是	ID	NUMBER(19)	否	用户满意度 id
否	USER_ID	NUMBER(19)	否	用户表主键(User. ID)
否	CLOUD_SERVICE_ID	NUMBER(19)	否	云服务表主键(CloudService. ID)
否	CLOUD_SATISFACTION	NUMBER(19)	否	用户满意度
否	USER_BEHAVE_ID	NUMBER(19)	否	空间索引列(UserInterestBehave. ID)

表 3　用户兴趣行为表结构

主键	名称	数据类型	可空	备注
是	ID	NUMBER(19)	否	顺序自动生成 id
否	LABEL_BEHAVE	NUMBER(19)	否	标签操作行为
否	LINK_NUMBER	NUMBER(19)	否	链接行为
否	BROWSE_TIME	NUMBER(19)	否	浏览时间

4　推荐模块

推荐模块是推荐系统的核心部分,主要功能就是运用不同的推荐算法推送有效满足用户需求的云服务。该部分主要采用基于 R 树的协同过滤算法(R_CF)实现推荐,包括前面的数据预处理和数据库存储结构的设计等一系列工作也是为了算法更好的实现。R_CF 算法只是获取和目标用户相似度最高的 top-n 个用户。下面描述了如何向目标用户推送云服务。

推荐引擎,是主动发现用户当前或潜在的需求,并主动推送信息给用户的信息网络。其主要功能是挖掘用户的喜好和需求,主动向用户推荐其感兴趣或者需要的对象。推荐引擎有助于选择哪种推荐算法或者推荐策略。基于 R_CF 算法,可以快速查找到 top-n 个最相似用户。下面列出两种向目标用户推荐数字资源云服务的策略:

(1) sim_{gi} 表示目标用户 g 与 top-n 中第 i 个用户的相似度,设 n 个用户共访问过 m 个云服务,s_m^i 表示第 i 个用户对云服务 m 的满意度。sim_{gi} 和 s_m^i 都可以通过用户满意度表获取。由式(2)计算后,向用户推荐 cloud_m 值最大的这个云服务即可。

$$\mathrm{cloud}_m = \sum_{i=1}^{n}(\mathrm{sim}_{gi} \times s_m^i) \tag{2}$$

(2)有可能 n 个用户共访问过的 m 个云服务,用户都没有进行评分,所以无法获取用户满意度,也就是 n 个用户可能恰好都是非注册用户,那么所有云服务 cloud_m 都为 0,使用步骤(1)中的方法就无法进一步筛选出用户最喜好的云服务了,此时,可以推送 m 个云服务中用户数最多的那个云服务。

5　数据更新模块

随着云计算技术在数字资源的发展与应用,各种系统、服务器等资源的整合使得数据量急剧上升。传统的大数据更新方式是夜间离线进行,白天推荐系统分析的数据是前一天的数据,晚上用户量急剧减少后,再进行批量离线大数据更新。这样使得数据分析和数据更新分时地访问数据仓库系统,隔离了相互影响。夜间离线更新在数据时效性方面存在巨大缺陷,云计算技术的发展,数字资源的应用不仅仅局限于中国。例如,国外的文献检索等数字资源越来越广泛地应用于高校师生,中国白天是美国夜间,美国夜间又是中国白天,这种时段的划分又难以分得清楚,可见,离线更新不符合当前大数据业务处理的需求[15]。

在线更新数据必然降低了数据查询的效率,数据查询对在线更新操作也会造成一定的阻力,两者之间

的相互影响限制了数据在线更新的发展。为了降低两者相互间的影响,图 3 展示了一种数据在线更新的设计思路[16]。

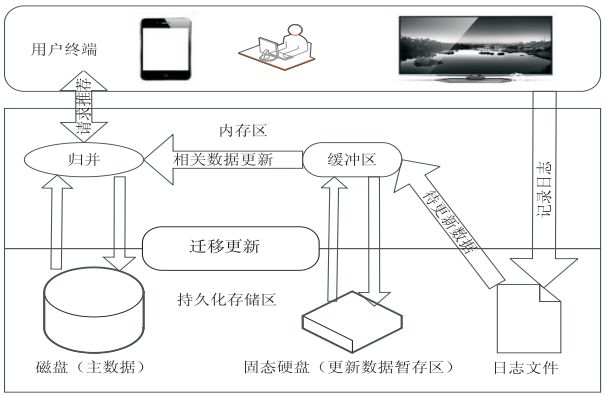

图 3 数据在线更新

图 3 中,内存、缓存区、磁盘、文件和固态硬盘可以存储数据。内存大部分还是要用于数据分析处理,小部分缓存区用于缓存需要更新的数据,磁盘主要还是存储主数据,日志文件是数据分析的原数据,而固态硬盘是分担缓存区来不及向内存更新的数据。固态硬盘具有良好的读性能,可以降低在线更新对查询操作的影响。固态硬盘的容量只需要是硬盘容量的 1% 即可,节省成本[17]。

服务器的日志文件记录了用户终端传递过来的最新的用户兴趣行为数据,日志文件经预处理后生成待更新的记录,待更新的记录都包含需要更新的数据记录的主键、操作(插入/修改/删除)以及具体的更新数值(插入的值/修改后的新值),新注册用户的用户兴趣行为等记录主键自动生成。如图 3 所示,待更新数据首先存放在缓冲区,缓冲区的数据和磁盘查询出来的主数据采用归并算法合并数据,供上层推荐算法分析处理;如果缓冲区已满,就将待更新数据暂时存储至固态硬盘;如果固态硬盘将满或者超过设定的阈值时,就将固态硬盘里的数据采用现有的归并机制进行合并回写到磁盘主数据,并清除固态硬盘数据。

持久化存储区加入了固态硬盘协助数据更新,不仅利用其高效的读写能力加快查询、更新操作,而且还承担了缓冲区的一部分工作,从而节省出内存区供推荐算法分析处理,从历史数据层面上有助于提高云服务推荐的效率和准确率。

6 结束语

为了解决“信息过载”问题,设计并实现了数字资源云服务推荐系统。该系统主要包括“数据预处理”、“数据存储”、“推荐算法”和“数据更新”四个模块。系统通过数据预处理模块对原始数据进行处理,建立用户兴趣行为模型;采用改进的协同过滤算法向目标用户推荐数字资源云服务。数字资源云服务推荐系统可以提高推荐速度,准确地向用户推荐满意度高的数字资源云服务,能够快速提升用户体验,有效解决“信息过载”问题。

参考文献:

[1] 吴小清. 广东高职院校图书馆数字资源建设现状与共建共享研究[J]. 资治文摘:管理版,2010(2):165-166.
[2] 胡新平. 云图书馆构想[J]. 情报理论与实践,2010,33(6):29-32.
[3] Oh J,Kim S,Kim J,et al. When to recommend:a new issue on TV show recommendation[J]. Information Sciences,2014,280:261-274.
[4] Sheng J,Liu S. A knowledge recommend system based on user model[J]. International Journal of Digital Content Technology & Its Applications,2010,4(9):168-173.
[5] 周玲元,段隆振. 个性化图书推荐系统设计与实现—以南昌航空大学图书馆为例[J]. 图书馆理论与实践,2014(12):106-109.
[6] 钟克吟. 基于标签与协同过滤算法的学术资源推荐系统的构建[J]. 图书馆理论与实践,2014(9):80-82.
[7] 张 瑶,陈维斌,傅顺开. 基于大数据的高校图书馆推荐系统仿真研究[J]. 计算机工程与设计,2013,34(7):2533-2541.
[8] 孟祥武,纪威宇,张玉洁. 大数据环境下的推荐系统[J]. 北京邮电大学学报,2015,38(2):1-15.
[9] 童恒庆,梅 清. Web 日志挖掘数据预处理研究[J]. 现代计算机:下半月版,2004(3):6-9.
[10] 印 鉴,王智圣,李 琪,等. 基于大规模隐式反馈的个性化推荐[J]. 软件学报,2014,25(9):1953-1966.
[11] 刘加伶,范 军. 基于用户访问树的 Web 日志挖掘数据预处理[J]. 计算机科学,2009,36(9):154-156.
[12] 李 燕,冯博琴,鲁晓锋. Web 日志挖掘中的数据预处理技术[J]. 计算机工程,2009,35(22):44-46.
[13] 胡秦斌,李广原. Web 使用记录挖掘前的事务识别方法[J]. 广西师范学院学报:自然科学版,2007,24(4):97-99.
[14] 闫 斌. 基于分布式的空间数据库引擎设计与实践[D]. 成都:电子科技大学,2011.
[15] 陈世敏. 大数据分析与高速数据更新[J]. 计算机研究与发展,2015,52(2):333-342.
[16] Athanassoulis M,Chen S,Ailamaki A,et al. Online updates on data warehouses via judicious use of solid-state storage[J]. ACM Transactions on Database Systems,2015,40(1):1-42.
[17] Gupta R. System to recommend related search queries:EP,EP2701080[P]. 2014.