基于大数据挖掘构建游戏平台个性化推荐系统的研究与实践

尤海浪[1],钱　锋[2],黄祥为[1],胡亮亮[1]
(1. 炫彩互动网络科技有限公司　南京 210029;
2. 中国电信股份有限公司广东研究院　广州 510630)

摘　要: 给出了一种基于大数据挖掘的手机游戏平台个性化推荐机制,通过对游戏用户行为数据的获取、存储、清洗、挖掘,由改进的余弦相似度算法计算出游戏的相似度,向用户推荐其喜欢的游戏。该机制可以有效提升游戏推荐的准确性,增强用户黏性,为游戏平台创造更多价值。

关键词: 大数据;手机游戏;个性化推荐
doi: 10.3969/j.issn.1000-0801.2014.10.005

Research and Practice of Building a Personalized Recommendation System for Mobile Game Platform Based on Big Data Mining

You Hailang[1], Qian Feng[2], Huang Xiangwei[1], Hu Liangliang[1]
(1. Dazzle Interactive Network Technologies, Nanjing 210029, China;
2. Guangdong Research Institute of China Telecom Co., Ltd., Guangzhou 510630, China)

Abstract: The personalized recommendation system was presented, which was based on data mining technology in mobile game platform, including the game data acquisition, data storage, data cleaning, data mining, and the cosine similarity algorithm was used to improve the similarity of the games. The mechanism can effectively improve the accuracy of game recommendation, enhance user stickiness, and create more value for the game platform.

Key words: big data, mobile game, personalized recommendation

1　引言

随着互联网的发展,人们逐渐从信息匮乏的时代进入信息过载(information overload)的时代。在这个时代,无论是作为信息消费者的普通用户,还是作为信息生产者的内容提供商都遇到了很大的挑战。比如手机游戏行业,用户关心的是如何从大量游戏中找到自己感兴趣的游戏,而游戏平台商则关心如何让自己平台的游戏脱颖而出。

个性化推荐[1]是根据用户的兴趣特点及行为向用户推荐其感兴趣的信息或产品,主要解决如何在海量信息中发现用户感兴趣的信息。对于手机游戏平台来说,通过基于大数据挖掘技术构建个性化推荐系统,能有效帮助用户发现喜欢的游戏,实现游戏消费者和游戏平台商的双赢。

结合笔者在手机游戏平台进行数据分析的相关工作经验,本文给出了在游戏平台上基于大数据挖掘技术,构建个性化推荐系统的实践,先存储用户的海量行为数据,然后基于 Hadoop 框架处理离线数据,进行游戏之间的相似度矩阵计算,运用 Redis 存储中间结果和最终推送结果,最后通过手机客户端向用户提供推荐列表。

2 推荐算法择优

2.1 推荐算法选择

个性化推荐系统算法通常有 ItemCF(基于商品的协同过滤)、UserCF(基于用户的协同过滤)、Content_Based(基于内容的推荐)、Slope One、SVD(singular value decomposition,奇异值分解)、组合算法等。

ItemCF 是当今很多大型网站都在采用的核心算法之一,适用于项目(item)的增长速度远远小于用户(user)且 item 之间的相似性比较稳定的场景,可以在离线系统中将 item 的相似度矩阵计算好,以供线上近乎即时地进行推荐。UserCF 常用于咨询服务类的应用,可以发现和用户具有同样爱好的人,因为用户的相似用户群非常敏感,所以需要频繁地计算出用户的相似用户矩阵,导致运算量会非常大。Content_Based 一般用于文本挖掘的项目中,每天都要根据 Web 生成的或者通过爬虫抓取的资讯,不断地计算 item 之间的相似性,提取关键词,该算法可以很好地解决推荐系统冷启动问题,比如想推出一个新的 item,因为没有一个人有对这个新 item 的评分和行为,所以之前的算法不可能推荐新的东西给用户,但可以用基于内容的算法将新的 item 计算出它属于哪个类,然后根据用户对该 item 类的喜好程度推荐新 item。Slope One 算法简单实现了 ItemCF 算法,该算法相比普通的 ItemCF 只需要一半(甚至更少)的存储量,更容易计算,但是准确性方面不够稳定,鲜用于商业系统。SVD 实际上是提取一般实矩阵“特征值”的算法,该算法拿到“特征值”后,可以分析出主成分因子,也就是说,可以对原来庞大的、常常又非常稀疏的矩阵进行降维和分解,可以大大降低矩阵的维度,提高运算的速度,但是需要付出较大的空间资源。组合算法多种多样,主要是对上面一些算法的组合操作,比如将多种算法计算出来的结果,加权之后排序推荐给用户,也可以将多种算法计算出来的结果,各取前几个推荐给用户,增加推荐结果的多样性等。

对比基于内容的协同过滤(content filtering)算法与基于行为的协同过滤(collaborative filtering)算法[2,3],发现基于内容的过滤算法主要利用物品的内容数据或者外部资讯,认为用户会喜欢和他以前喜欢的在内容上相似的物品;而基于行为的协同过滤算法通过分析大量的用户对物品的行为数据,从中找出特定的行为模式,据此来预测用户的兴趣并给用户做出推荐。对于手机游戏平台来说,因游戏产品分类属性尚未标准化,基于内容过滤的推荐算法难以反映用户的真正需求;相对而言,游戏平台对用户的访问、下载与付费等行为数据均有记录,现阶段适合采用协同过滤算法。

基于行为的协同过滤算法大体上分为基于用户的 UserCF 算法和基于物品的 ItemCF 算法,UserCF 给用户推荐那些和他有共同兴趣爱好的用户喜欢的物品,需要维护一个用户相似度的矩阵,而 ItemCF 给用户推荐那些和他之前喜欢的物品类似的物品,需要维护一个物品相似度矩阵。现阶段游戏平台一般更注重用户消费行为而不是社交行为,而且一般用户的基数远大于游戏的基数,存储 UserCF 的用户相似度矩阵的开销远远大于存储 ItemCF 的物品相似度矩阵,所以选择基于物品的协同过滤算法。

对用户的个性化推荐,主要有两个步骤:一是计算游戏之间的相似度;二是根据游戏的相似度和用户的历史行为生成游戏推荐列表。

要计算游戏的相似度,需先确定用户与游戏关系的矩阵。如图 1 所示,最左边是用户下载游戏的集合,用户下载过某款游戏,则认为该用户对该游戏感兴趣,每一行代表一个用户感兴趣的游戏集合。对于每个物品集合,将里面的物品两两进行组合,得到一个新的矩阵,这些矩阵“相加”得到最右边的 C 矩阵,$C_{[i][j]}$表示同时下载游戏 i 和游戏 j 的用户列表。

游戏相似度的计算有如下几种算法。

(1)基本算法

计算式为:

$$w_{ij}=\frac{|N(i)\cap N(j)|}{|N(i)|} \tag{1}$$

其中,w_{ij} 是游戏 i 和游戏 j 的相似度,分母$|N(i)|$是喜欢游戏 i 的用户数,而分子$|N(i)\cap N(j)|$是同时喜欢游戏 i 和游戏 j 的用户数。

(2)余弦相似度(cosin-base)算法

计算式为:

$$w_{ij}=\frac{|N(i)\cap N(j)|}{\sqrt{|N(i)||N(j)|}} \tag{2}$$

通过降低游戏 j 的权重,该算法能减轻热门游戏和很多游戏相似的可能性,从而提升推荐的质量。

(3)余弦相似度 α(cosin-alph)算法

计算式为:

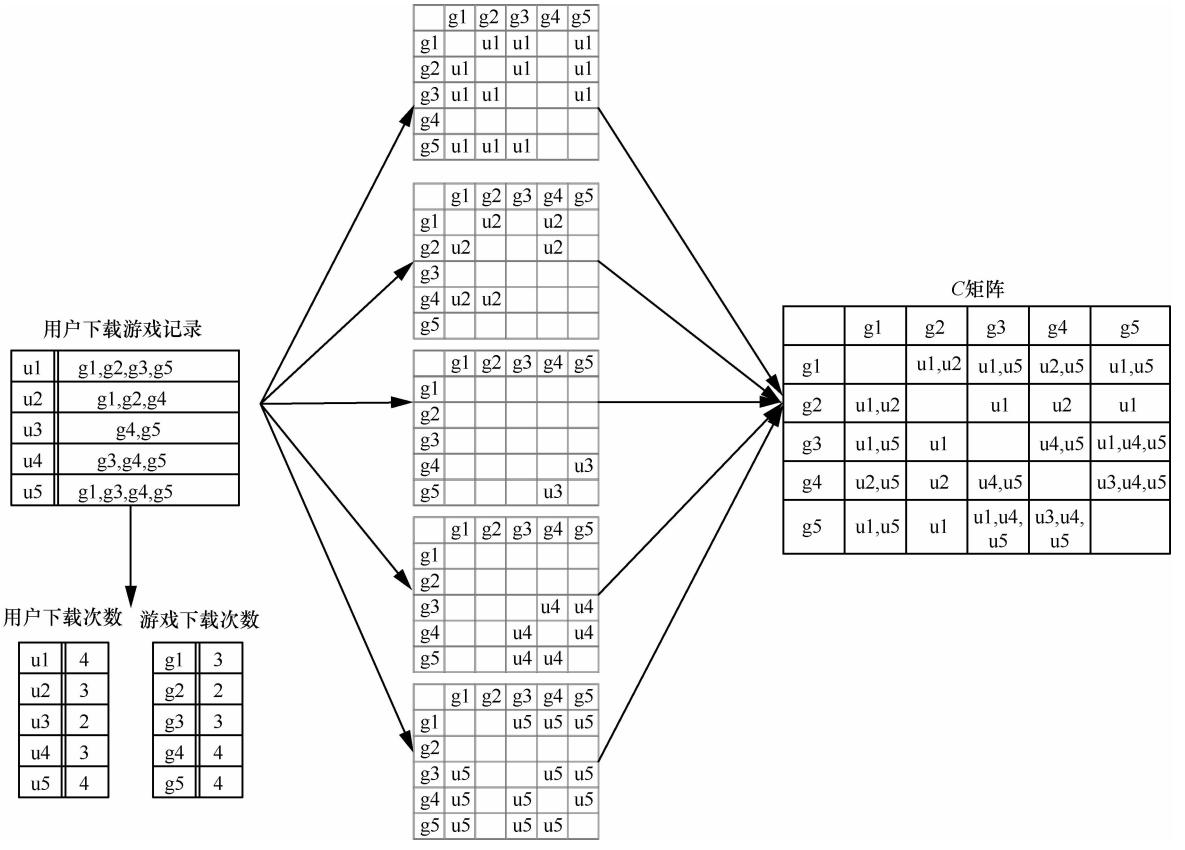


图 1　用户与游戏关系矩阵

$$w_{ij}=\frac{|N(i)\cap N(j)|}{|N(i)|^{\alpha}|N(j)|^{1-\alpha}} \tag{3}$$

该算法进一步降低了游戏 j 的权重,可以根据实际的应用效果指定 α 的取值。

(4)改进的余弦相似度(cosin-iuf)算法

对于游戏平台来说,存在部分恶意下载用户,为了保证游戏之间相似度的可靠性,需要修正活跃用户对游戏相似度的贡献,即对同一款游戏来说,已经下载了 100 款游戏的用户的贡献度要小于只下载了 10 款游戏的用户,调整后的计算式如下:

$$w_{ij}=\frac{\sum_{u\in N(i)\cap N(j)}\frac{1}{\mathrm{lb}(1+|N(u)|)}}{\sqrt{(|N(i)||N(j)|}} \tag{4}$$

该式只是对活跃用户做了一种软性的惩罚,在实际的计算中,对于过于活跃的用户,为了避免相似度矩阵过于稠密,一般直接忽略其兴趣列表,不将其纳入相似度计算的数据集中。

(5)改进的余弦相似度的归一化算法

计算式为:

$$w_{ij}=\frac{w_{ij}}{\max\limits_{j} w_{ij}} \tag{5}$$

在改进的余弦相似度算法基础上进行归一化,可以进一步增加推荐的准确度,也可以提高推荐的覆盖率和多样性。选择该算法为游戏平台的推荐算法进行实践。

完成第一步的游戏相似度计算后,通过如下计算式计算用户 u 对游戏 j 的兴趣:

$$p_{ij}=\sum_{i\in N(u)\cap S(i,k)} w_{ji}r_{ui} \tag{6}$$

这里的 $N(u)$是用户喜欢的游戏的集合,$S(i,k)$是和游戏 i 最相似的 k 个游戏的集合,w_{ji} 是游戏 j 和游戏 i 的相似度,r_{ui} 是用户 u 对游戏 i 的兴趣(对于游戏平台来说 r_{ui}=1)。通过该算法,和用户历史上感兴趣的游戏越相似的游戏, 越有可能在用户的推荐列表中获得比较高的排名。

2.2　算法评价指标

推荐算法的优劣需要关注精度、覆盖、多样性等方面,具体指标如下。

(1)精度指标:召回率 *recall*/准确率 *precision*

对用户 u 推荐 N 个游戏记为 $R(u)$,用户 u 在测试集上喜欢的物品集合为 $T(u)$,通过准确率/召回率评测推荐算法的精度,召回率描述有多少比例的用户—游戏下载记录包含在最终的推荐列表中,而准确率描述最终推荐列表中有多少比例是发生过的用户—游戏下载记录,具体定义如下:

$$recall=\frac{\sum_u|R(u)\cap T(u)|}{\sum_u|T(u)|} \tag{7}$$

$$precision=\frac{\sum_u|R(u)\cap T(u)|}{\sum_u|R(u)|} \tag{8}$$

(2)覆盖率指标 *coverage*

覆盖率表示最终的推荐列表中包含多大比例的游戏。如果所有的游戏都被推荐给至少一个用户,那么覆盖率就是 100%。覆盖率反映推荐算法发掘长尾的能力,覆盖率越高,说明推荐算法越能够将长尾中的游戏推荐给用户。采用最简单的覆盖率定义如下:

$$coverage=\frac{|\cup_{u\in U}R(u)|}{|I|} \tag{9}$$

其中,$|\cup_{u\in U}R(u)|$表示通过推荐系统推荐给用户的游戏去重数,$|I|$指"爱游戏"平台中所有的游戏数。

(3)多样性指标 *diversity*

多样性描述了推荐列表中游戏两两之间的不相似性。因此,多样性和相似性是对应的,计算式如下所示,其中 $s(i,j)\in[0,1]$定义了游戏 i 和游戏 j 之间的相似度。

$$diversity=\frac{\sum_{i,j\in R(u),i\neq j}(1-s(i,j))}{|R(u)||R(u)-1|} \tag{10}$$

推荐系统的整体多样性可以定义为所有用户推荐列表多样性的平均值:

$$diversity=\frac{1}{|U|}\sum_{u\in U}diversity(R(u)) \tag{11}$$

3 基于大数据挖掘的个性化推荐系统体系架构

3.1 数据获取与存储

对上述推荐算法在"爱游戏"平台进行了实践,"爱游戏"平台是由炫彩互动网络科技有限公司(中国电信游戏基地)全力打造的互动娱乐平台,目前用户数超过 2 亿户,日均数据增量 100 GB。

业务平台的数据分析首先需要获取数据并存储数据。游戏平台的个性化推荐系统一般采用用户的下载行为作为用户的行为数据,一旦用户下载了一款游戏,则视该用户对游戏产生了一个正向喜欢。采用如图 2 所示的架构进行数据获取与存储。

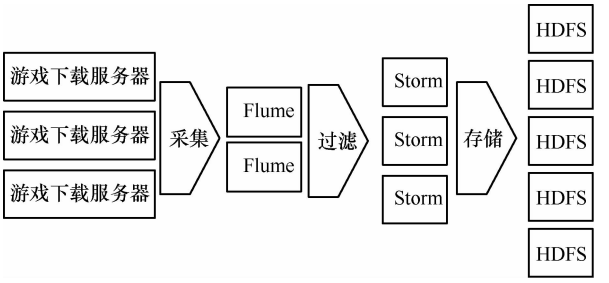


图 2 游戏数据获取与存储

游戏下载服务器为用户提供游戏下载功能,当用户发出游戏下载请求,下载服务器在本地日志记录一条用户下载记录。利用实时日志采集系统 Flume 对日志数据进行高效、实时的采集,然后传递给实时计算系统 Storm,Storm 按照设定的规则进行数据过滤,最后将有效数据存入 Hadoop 分布式文件系统(HDFS)[4]进行固化。

HDFS 对硬件的需求比较低,可以运行在低廉的商用服务器集群上,能充分利用老旧机器的存储能力。通过 HDFS 的"一次写入、多次读取"机制[5,6],能够非常快速地处理用户的海量访问数据,通过分布式的文件存储机制,可以将用户的历史访问记录存储很久,从而为分析用户行为提供坚实的数据支撑。

3.2 数据清洗与挖掘

游戏数据清洗与挖掘如图 3 所示。

数据存储好之后,采用 MapReduce 计算框架[7],可以快捷地对大型数据矩阵进行计算,从而为个性化推荐系统提供计算支持。首先进行数据清洗,过滤掉非法的用户和游戏,其中每个用户对每个游戏只能下载一次。在实际计算前需要进行数据重构,把用户和游戏的标识唯一化,同时生成

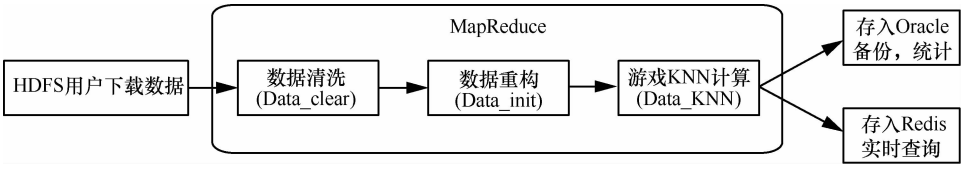


图 3 游戏数据清洗与挖掘

用户的下载数表和游戏被下载次数表。相似度矩阵计算由 Data_KNN 来完成,计算出的结果存两份,一份由 Oracle 数据库进行存储,供推荐系统的评测和统计使用,另一份由 Redis 高速缓存服务器进行存储,供前端各类应用实施查询。

3.3 游戏个性化推荐流程

如图 4 所示,面向用户的游戏推荐流程如下。

(1)首先用户通过客户端访问游戏平台,点击进入任意游戏详情页面。客户端发送用户的访问请求给后台程序。

(2)后台程序获取用户当前访问的游戏 ID,并根据用户 ID 来获取用户的历史记录。

(3)通过 Redis 获取该游戏的相似度矩阵。

(4)使用推荐算法根据用户的相似度矩阵、当前访问游戏 ID、用户历史访问游戏 ID 计算用户可能喜欢的游戏列表。

(5)对用户可能喜欢的游戏列表按照相似度排行。

(6)取前 Top*N* 个游戏,并返回结果给客户端,客户端将相应的游戏显示在“猜你喜欢”栏目中。

4 推荐算法验证

“爱游戏”平台目前每天的下载用户数为 8 万户左右,人均下载 3~5 款游戏,累计 3 个月的用户下载数据为 3 000 万条左右,具有下载行为的用户 300 万户,游戏相似度矩阵规模为 5 000×5 000。应用余弦相似度推荐算法,对数据进行了计算,结果见表 1。

表 1 算法比较(推荐游戏数为 10 款)

算法	准确率	召回率	覆盖率	多样性
cosin-base	0.116	0.051	0.465	0.905
cosin-alph	0.125	0.056	0.353	0.881
cosin-iuf(归一)	0.133	0.063	0.468	0.960

从表 1 中的结果可以看出,改进的余弦相似度的归一化推荐算法相对基本算法在准确率、召回率等各个指标上均有所提升。通过降低热门游戏的权重,能有效提升准确率和召回率。通过降低活跃用户的权重,能有效提升游戏覆盖度和多样性,从而强化推荐系统发掘长尾的能力。

推荐算法还有一个重要的影响因素,即向用户推荐的游戏个数,针对该因素影响情况进行针对性的效果分析,分析结果见表 2。

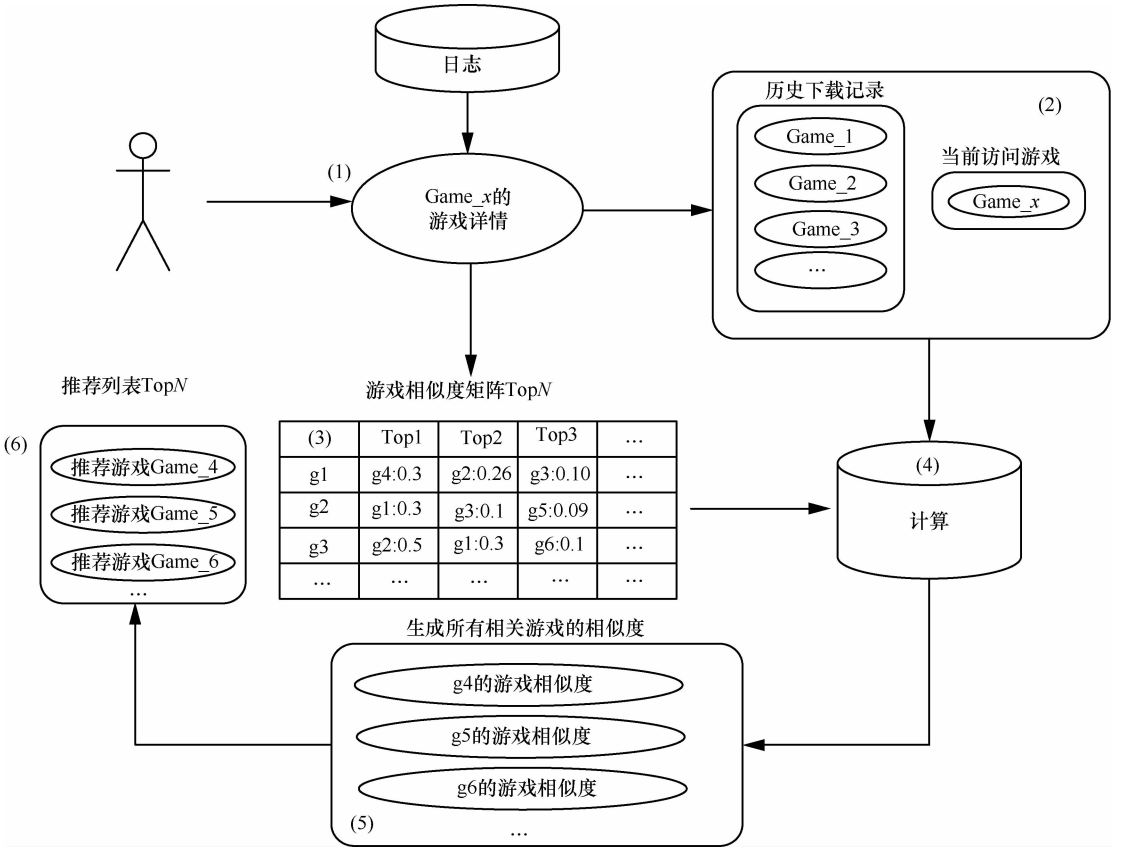


图 4 游戏推荐流程

表 2　不同推荐数量下的算法效果

推荐数/款	准确率	召回率	覆盖率	多样性
5	0.082	0.075	0.458	0.938
10	0.133	0.063	0.468	0.960
15	0.141	0.041	0.508	0.962
20	0.173	0.038	0.517	0.964
30	0.224	0.033	0.590	0.981
50	0.299	0.026	0.691	0.990
80	0.376	0.021	0.721	0.992
100	0.413	0.018	0.721	0.994

由图 5 可知,随着游戏推荐数的增大,游戏的准确率、覆盖率明显上升,召回率则逐步下降,与实践情况相符,从而说明了算法的正确性和实用性。

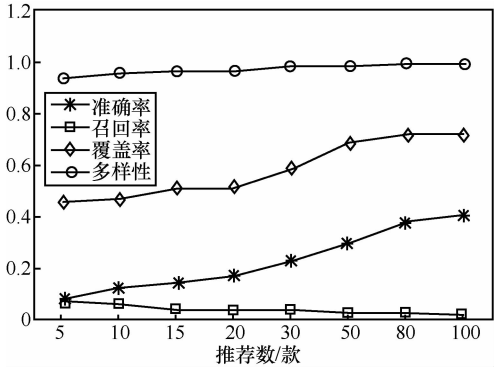


图 5　不同推荐数量下采用改进的
余弦相似度的归一化算法的效果比较

5　结束语

本文介绍了基于大数据挖掘技术,构建游戏平台个性化推荐系统的方法与实践, 采用 Hadoop 框架处理离线数据,进行游戏之间的相似度矩阵计算,运用 Redis 存储中间结果和最终推送结果,其中重点结合游戏平台实际情况研究余弦相似度算法,并通过降低热门游戏权重和降低活跃用户权重等几种方法进行算法改进,根据实践计算结果对相应算法进行了对比和分析,构建了一种适用于游戏平台个性化推荐的机制和方法,为其他业务平台大数据分析提供了良好的参考和借鉴。目前的研究基于用户的下载行为, 随着游戏用户行为和游戏数据趋于多样化和复杂化,需要对数据源做进一步的拓展, 并考虑不同数据源的权重,提升个性化推荐的效果,这是下一步研究的方向和目标。进一步考虑的数据包括用户访问、用户付费、用户已安装的应用软件、游戏的描述信息等,将采用复合权重相加的方式拟合物品相似度矩阵。

参考文献

1　Han J W, Kamber M, Pei J. 数据挖掘:概念与技术. 范明,孟小峰译. 北京:机械工业出版社,2013

2　李慧,胡云,施珺. 社会网络环境下的协同推荐方法. 计算机应用,2013(11)

3　邓晓懿,金淳,韩庆平等. 基于情境聚类和用户评级的协同过滤推荐模型. 系统工程理论与实践,2013(11): 2945~2953

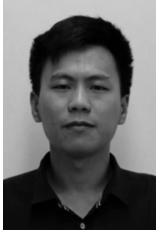
4　Shin-gyu Kim, Junghee Won, Hyuck Han, *et al*. Improving Hadoop performance in intercloud environments. Performance Evaluation Review, 2011, 39(3):107~109

5　周江,王伟平,孟丹等. 面向大数据分析的分布式文件系统关键技术. 计算机研究与发展,2014, 51(2)

6　周吉寅,陈媛,姚晨等. 使用 Hadoop 实现应用商店中的相关推荐模型. 现代计算机:上下旬,2013(17):20~23

7　Fang W, Pan W B, Cui Z M. View of MapReduce: programming model, methods, and its applications. IETE Technical Review, 2012, 29(5)

[作者简介]



尤海浪,男,炫彩互动网络科技有限公司工程师,主要研究方向为大数据、手机游戏等。

钱锋,男,中国电信股份有限公司广东研究院工程师,主要研究方向为大数据、云计算等。

黄祥为,男,炫彩互动网络科技有限公司工程师,主要研究方向为大数据、移动互联网等。

胡亮亮,男,炫彩互动网络科技有限公司工程师,主要研究方向为大数据、手机游戏等。

(收稿日期:2014-09-25)