

基于 SaaS 模式的电子商务推荐平台

刘 嘉^{1,2,3*}, 惠成峰^{1,2}, 都兴中^{1,2}, 陈振宇^{1,2}

(1. 计算机软件新技术国家重点实验室(南京大学), 南京 210093; 2. 南京大学 软件学院, 南京 210093;

3. 南京大学 工程管理学院, 南京 210093)

(* 通信作者电子邮箱 liujia@software.nju.edu.cn)

摘 要:针对一些电子商务网站缺乏资源难以实施推荐系统的现状,提出一种基于“软件即服务”(SaaS)模式的推荐服务架设方式并实现了原型平台。该平台使用统一的脚本收集电子商务网站的用户行为信息,并通过标准的接口提供推荐服务,从而实现平台与电子商务网站的低耦合以降低部署成本。平台上线运行结果表明,该模式的推荐服务能够有效帮助电子商务网站提升转化率和增加订单量。

关键词:电子商务;推荐系统;推荐算法;SaaS 模式;转化率

中图分类号:TP311.1 **文献标志码:**A

E-commerce recommendation platform based on SaaS pattern

LIU Jia^{1,2,3*}, HUI Cheng-feng^{1,2}, DU Xing-zhong^{1,2}, CHEN Zhen-yu^{1,2}

(1. State Key Laboratory for Novel Software Technology (Nanjing University), Nanjing Jiangsu 210093, China;

2. Software Institute, Nanjing University, Nanjing Jiangsu 210093, China;

3. School of Management and Engineering, Nanjing University, Nanjing Jiangsu 210093, China)

Abstract: Some E-commerce sites cannot deploy independent recommender systems themselves due to limited resources. In order to help these sites deploy recommender systems quickly and conveniently, an E-commerce recommendation platform based on Software-as-a-Service (SaaS) pattern was proposed and implemented. This platform used unified scripts to collect user action information and provided recommendation services through standard interface. It realized a low coupling between platform and E-commerce sites so that the cost of implementation was reduced. The results of online operations show that this platform can help E-commerce sites increase the conversion rate and the volume of orders.

Key words: E-commerce; recommender system; recommendation algorithm; Software-as-a-Service (SaaS) pattern; conversion rate

0 引言

电子商务^[1]的爆发式增长带来了海量的在线商品信息,传统的依靠用户主动搜索关键词的信息过滤方式在电子商务领域已经越来越无法满足用户的需求,在这样的背景下,推荐系统^[2]成为电子商务的重要组成部分。通过收集和分析用户的行为数据,推荐系统为用户提供更加个性化的推荐,方便用户快速找到符合自己兴趣的产品。在实际应用中,Amazon 推荐系统^[3]有着非常良好的表现,成为了推荐系统实际应用的经典案例。学术界经过对推荐方法十多年的研究,形成了协同过滤^[4-6]、内容过滤^[7-9]和混合推荐^[10-12]三个主要分类。协同过滤的主要思想通过计算相似用户或相似产品来产生推荐。内容推荐则通过分析用户曾经感兴趣的产品的特征来产生推荐。混合推荐将协同过滤和内容推荐相结合,综合利用两者的特点来进行推荐。

在电子商务网站的实际应用中,通常的做法是为单个电子商务网站单独建立一套推荐系统,这种方式可以根据目标电子商务网站销售的产品和用户群的特征进行定制化配置。比如对产品和用户聚类^[13]是推荐系统常用到的技术,单独建立的专为一个电子商务网站服务的推荐系统可以依据适合该网站特征的方式来聚类,比如客户对产品的价格敏感,可以按

照产品的价格区间聚类;不同年龄层次消费习惯明显不同的产品类别,可以按照用户年龄进行聚类。定制的推荐系统还可以利用网站特有的数据来提高推荐系统准确率,比如有用户给产品标注标签信息^[14]的网站可以利用标签信息来改进协同过滤的准确性。这些方法都使推荐系统更符合目标电子商务网站的场景和特点,但也造成了推荐系统与电子商务网站系统极高的耦合性。

当前电子商务行业的趋势是中小型电子商务网站快速发展,各个细分行业形成了许多垂直型的电商网站,这些中小型电子商务网站销售的产品各不相同,面对的消费者特征也各不相同,为每一个网站单独设计建立一套适合的推荐系统极大地浪费中小型电子商务网站的资源,这导致绝大多数中小型电子商务网站还没有部署推荐系统。

为此,本文提出了基于软件即服务^[15](Software-as-a-Service, SaaS)模式的推荐服务解决方案,为中小型电子商务网站建立统一的推荐平台,该平台可集中为多个电子商务网站提供推荐服务,帮助各电子商务网站快速部署推荐系统。基于 SaaS 模式的推荐平台有以下优点:1)使用统一的脚本收集各网站信息,部署快速简便;2)SaaS 模式使得推荐平台与电子商务网站耦合度很低,所有推荐相关信息的存储和处理都在推荐平台完成,对电子商务网站无额外负担;3)提供多样化的普适的推荐方法,满足绝大多数网站的基本需求。

收稿日期:2012-03-02;修回日期:2012-05-10。 基金项目:教育部人文社科基金资助项目(10YJC870020, 10YJC630283)。

作者简介:刘嘉(1976-),男,河北任丘人,讲师,博士研究生,主要研究方向:电子商务、推荐系统、社会网络分析;惠成峰(1988-),男,江苏张家港人,硕士研究生,主要研究方向:电子商务、推荐系统;都兴中(1988-),男,辽宁沈阳人,硕士研究生,主要研究方向:电子商务、推荐系统;陈振宇(1978-),男,广东梅州人,副教授,博士,主要研究方向:软件分析与测试、电子商务。

1 基于 SaaS 模式的电子商务网站推荐平台

SaaS 模式的特征是通过统一的软硬件系统平台为多个用户提供服务,平台为不同用户提供相同的服务,通过对元数据的配置来实现一定的个性化需求。目前电子商务行业的趋势是许多细分行业垂直型电子商务网站快速发展,这些网站规模不大,单个网站资源有限,很难独立构建自己的推荐系统,但推荐系统又是提升客户体验不可或缺的一部分。SaaS 模式推荐平台很好地解决了这个问题,为电子商务网站提供推荐服务调用接口,电商网站通过统一的脚本与推荐平台交互,交互的内容包括用户行为信息收集以及推荐结果的请求及交互。平台为网站提供多种实用推荐方法,每种推荐方法都实现为一个服务,用户可方便地配置使用哪些推荐服务。所有推荐服务都可以基于用户的行为数据实时返回推荐结果。

1.1 SaaS 模式

SaaS 模式的核心思想在于将软件以服务的模式提供给客户,最常见的应用是向企业提供信息系统平台服务。使用该模式,企业无需搭建硬件平台,也无需购买软件,而是通过互联网远程租用软件服务。该模式大大降低了企业构建信息平台的初始开销,同时不再需要负责软硬件的维护,既为企业节约了成本,也为企业使用软件提供了方便。

基于 SaaS 模式的电子商务推荐平台将推荐系统从电子商务网站中独立出来,在服务器端搭建一个通用的推荐系统平台,使用统一的服务接口与使用服务的电子商务网站交互。各电子商务网站只需挂载脚本来收集用户行为,就可以远程使用推荐服务。数据处理和推荐运算都在云端完成,不给原电子商务网站造成任何多余负担。

推荐平台架构如图 1 所示。各电商网站分别通过统一的服务接口与推荐平台交互。首先在各电商网站中挂载统一的脚本,脚本收集电商网站用户的行为信息,将信息通过服务接口发送到推荐平台,由用户行为收集服务器接收,并将数据存入数据库中。当电商网站需要推荐服务时,向推荐平台发送推荐请求,由推荐服务器响应,推荐服务程序分析数据库中记录的该网站用户的行为数据,返回推荐结果。

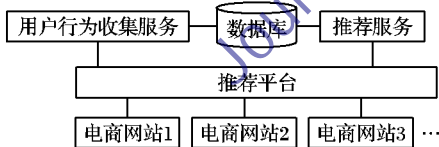


图 1 SaaS 模式推荐平台架构

1.2 前端脚本

为方便电子商务网站快速部署推荐服务,推荐平台使用统一的脚本收集用户行为信息。由于各网站结构不同,挂载脚本之前需要在脚本中配置基础行为参数,用于准确收集相关信息。具体参数在 parameter.js 中配置:

- 1) hostname;
//定义网站名,以此区分来自不同网站的数据和请求
- 2) productURLpattern;
//不同网站产品页的 URL 模式不同,用于识别产品页
- 3) shopcartID; //“购物车控件 ID”,用于识别该控件
- 4) orderButtonID; //“下单控件 ID”,以此识别该控件
- 5) orderedItemsID;
//订单页中“产品元素 ID”,用于获取被下单产品

脚本主要收集 3 种用户行为数据:用户的浏览行为、加入购物车行为和购买行为。收集信息的主要格式如表 1~3 所示。

表 1 用户浏览行为

数据	描述	来源
网站名	用于确定数据来源	在脚本中配置
行为	用于表示用户的行为,此处为浏览	由脚本自动生成
用户 ID	进行本次浏览的用户的 ID	由脚本自动生成
产品 ID	用户浏览的产品的 ID	脚本自动从网页获取
时间	用户浏览页面的时间	脚本自动获取

表 2 用户加入购物车行为

数据	描述	来源
网站名	用于确定数据来源	在脚本中配置
行为	用于表示用户的行为,此处为加入购物车	由脚本自动生成
用户 ID	进行本次加入购物车行为的用户的 ID	由脚本自动生成
产品 ID	用户加入购物车的产品的 ID	脚本自动从网页中获取
时间	用户加入购物车的时间	脚本自动获取

表 3 用户购买行为

数据	描述	来源
网站名	用于确定数据来源	在脚本中配置
行为	用于表示用户的行为,此处为购买	由脚本自动生成
用户 ID	进行本次购买的用户的 ID	由脚本自动生成
产品 ID	用户购买的产品的 ID	脚本自动从网页中获取
时间	用户购买的时间	脚本自动获取

本系统通过 user.js、visit.js、shopcart.js、order.js 共 4 个脚本收集用户浏览、加入购物车和购买信息。

user.js 脚本负责生成新用户 ID 和识别老用户 ID,老用户的 ID 存放在浏览器 cookie 中。脚本伪代码如下:

- 1) if (cookie not exists) { //若 cookie 不存在
- 2) generate cookie; //创建 cookie
- 3) generate user_id; //创建 user_id
- 4) save user_id in cookie; //将 user_id 存入 cookie
- 5) }
- 6) else if (cookie exists) { //若 cookie 存在
- 7) read cookie from browser; //读取 cookie
- 8) if (user_id not exists in cookie) {
- 9) generate user_id; //创建 user_id
- 10) save user_id in cookie; //将 user_id 存入 cookie
- 11) set cookie expiration time; //设置 parameter.js 中定义的 cookie 过期时间
- 12) }
- 13) else if (user_id exists in cookie) {
- 14) read user_id from cookie; //读取 user_id
- 15) }
- 16) }

visit.js 负责收集用户的浏览过程信息,用户每浏览一个页面,visit.js 都会向服务器发送用户的浏览信息。脚本伪代码如下:

- 1) read hostname from parameter.js; //读取网站名
- 2) read user_id from user.js; //读取用户 ID
- 3) read URL of current visiting page from browser //读取用户访问网页的 url
- 4) read product ID; //根据 parameter.js 中产品页 url 模式识别产品 ID
- 5) read time from system; //获取系统时间
- 6) send visit information; //根据 parameter.js 中的服务器地址发送用户浏览信息

shopcart.js 负责收集用户加入购物车行为,当用户将某产品加入购物车时,shopcart.js 会向服务器发送该用户将产品加入购物车信息。脚本伪代码如下:

```
1) read shopcartID from parameter.js;  
    //从 parameter.js 中读取加入购物车控件 ID  
2) add listener to shopcart button;  
    //根据 shopcartID 对加入购物车控件加监听  
3) if (shopcart button clicked) {  
    //如果加入购物车控件被点击  
4) read hostname form parameter.js; //读取网站名  
5) read user_id from user.js; //读取用户 ID  
6) read page URL from browser; //读取当前网页 url  
7) read productID from URL;  
    //根据 parameter.js 中产品页 URL 模式读出产品 ID  
8) read system time;  
9) send add to shopcart information;  
    //根据 parameter.js 中的服务器地址发送加入购物车信息  
10) }
```

order.js 负责收集用户购买行为,当用户下单购买时,order.js 会向服务器发送用户购买行为信息。脚本伪代码如下:

```
1) read orderButtonID from parameter.js;  
    //从 parameter.js 中读取下单控件 ID  
2) add listener to order button;  
    //根据 orderButtonID 对下单控件加监听  
3) if (order button clicked) { //如果下单控件被点击  
4) read hostname form parameter.js; //读取网站名  
5) read user_id form user.js; //读取用户 ID  
6) read orderedItemsID from parameter.js;  
    //读取 parameter.js 中订单页中产品元素 ID  
7) read productID;  
    //从 orderedItemsID 区域中读取被下单的产品 ID  
8) read system time;  
9) send order information;  
    //根据 parameter.js 中的服务器地址发送下单信息  
10) }
```

各脚本文件关系如图 2 所示。作为配置脚本,parameter.js 中包含所有其他脚本需要的基本配置信息,这些信息在各个网站中各不相同,需要人工配置。user 脚本需要 parameter 脚本提供用户 ID 在 cookie 中的存储时限,即过期时间;visit、shopcart 和 order 脚本都需要从 parameter 脚本中获取网站名信息和服务器地址信息;visit 和 shopcart 脚本需要 parameter 脚本提供产品页 URL 模式信息以识别产品页对应的产品;shopcart 脚本需要从 parameter 脚本获取加入购物车控件的 ID 信息,以此来识别产品页中的加入购物车控件;order 脚本需要从 parameter 脚本获取下单控件 ID 和订单页中产品元素 ID,以此来识别下单控件和从订单页中读取被下单的产品。user.js 用来生成用户 ID,并识别在 cookie 中有记录的老用户,这些用户信息会被 visit、shopcart 和 order 这三个记录用户行为的脚本使用。

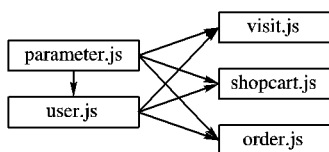


图2 前端脚本关系

1.3 服务器端信息接收

服务器端使用统一的格式接收各电子商务网站的信息。各电子商务网站将信息发送到统一服务器,服务器根据不同网站信息中不同的网站名参数,将网站信息分别存入该网站

对应的数据库。

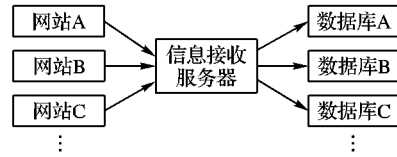


图3 信息接收流程

1.4 可配置的推荐算法

平台提供了 7 种通用性较强的推荐算法。网站使用不同的推荐请求标号来区别是由哪种页面发出的推荐请求,再根据配置情况选择相应的推荐算法,返回推荐结果。

典型的推荐请求内容如表 4 所示,提供的推荐方法如表 5 所示。

表4 推荐请求内容

数据	描述	来源
网站名	用于确定数据来源	在脚本中配置
推荐标号	不同类型的页面使用不同的标号	在脚本中配置
用户 ID	目标推荐用户的 ID	由脚本自动生成
产品 ID	用户正在浏览的产品,若用户正在浏览非产品页面,则无	脚本自动获取
返回推荐结果数量	返回推荐结果中推荐产品的个数	在脚本中配置
时间	用户浏览页面的时间	脚本自动获取

表5 推荐方法

推荐方法	方法描述
最多浏览推荐	统计一段时间内被浏览次数最多的 N 个产品,适用于所有类型的页面,在加入购物车数据和购买数据较少的电子商务网站使用
最多加入购物车推荐	统计一段时间内被加入购物车次数最多的 N 个产品,适用于所有类型的页面,在购买数据较少的电子商务网站使用
最多购买推荐	统计一段时间内被购买次数最多的 N 个产品,适用于所有类型的页面
共同浏览推荐	统计一段时间内与目标产品被用户共同浏览次数最多的 N 个产品,适用于产品页面
共同加入购物车推荐	统计一段时间内与目标产品被用户共同加入购物车次数最多的 N 个产品,适用于产品页面
共同购买推荐	统计一段时间内与目标产品被用户共同购买次数最多的 N 个产品,适用于产品页面
内容推荐	分析用户最近浏览、加入购物车和购买的产品的特征,推荐具有这些特征的产品

通过推荐方法配置页面配置推荐对应表,电子商务网站可以将每个推荐标号对应到一种推荐方法,当具有该标号的页面发出推荐请求时,推荐服务器会根据推荐标号返回对应推荐方法产生的推荐结果。配置页面推荐对应表示例如表 6 所示。

表6 配置推荐表

推荐标号	推荐方法	推荐标号	推荐方法
rec1	最多浏览推荐	rec4	共同购买推荐
rec2	共同浏览推荐	rec5	最多购买推荐
rec3	内容推荐		

推荐标号 rec N 始终对应到某一类页面,例如首页推荐使用推荐标号 rec1,用户只需在推荐配置表中配置推荐方法即可更改首页使用的推荐方法,但注意每类页面适用的推荐方法有限制条件,例如首页并非产品页,无法使用共同浏览推荐和内容推荐等输入参数包含某一特定产品的推荐方法。

1.5 推荐平台部署

推荐相关服务与电商网站服务挂载在不同的服务器上,推荐服务作为第三方服务提供,与电商网站现对独立。终端用户通过互联网访问电商网站,通过挂载在电商网站上的JS脚本收集用户行为,并将这些信息通过互联网发送至推荐服务器上的相关服务接口,信息接收服务将用户信息保存在部署于推荐服务器上的数据库中,电商网站通过互联网请求推荐服务器提供推荐,推荐服务从数据库中读取该电商网站相关信息,产生推荐结果,将推荐结果通过互联网发送给电商网站,通过电商网站呈现给终端用户。

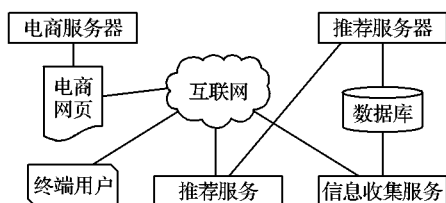


图4 推荐平台部署

2 实验及结果分析

2.1 实验背景

本实验实施于一个内衣电子商务网站。实验在该网站上调用推荐平台的服务进行了为期3周的实验。实验中网站首页和产品页面挂载了推荐服务,由于该网站购买数据并不充足,所以在首页使用了最多浏览推荐,在产品页面使用了最多共同浏览推荐。在为期3周的实验开始时,未使用历史数据,全部使用最新数据进行推荐。

2.2 实验结果分析

在实验过程中,网站总访问量35887次,产品页访问量8389次,推荐产品被点击1269次,用户下单131次,下单中有26次是由推荐产生的。

衡量电子商务网站流量质量的最重要的指标之一为转化率,即流量转化为订单的比例。计算公式如下:

流量转化率 = 下单产品数 / 产品浏览数

将推荐产生的流量和非推荐产生的流量分开统计:非推荐流量的转化率为1.47%,推荐流量的转化率为2.05%,推荐将转化率提高了39.4%。推荐的产品更符合用户的兴趣,带来了更高的转化率。

如果将粒度定位到单个用户上,推荐的效果更为明显。用户转化率计算公式如下:

用户转化率 = 下单的用户数 / 访问网站的总用户数

同样点击过推荐的用户和未点击过推荐的用户分开计算:访问过推荐链接的用户中有2.86%最终下了订单,而未访问推荐链接的用户仅有0.7%下了订单。

如果考察推荐对网站的贡献度,也可看出推荐流量质量更高。从图5可看出,推荐以15%的流量贡献了20%的订单。

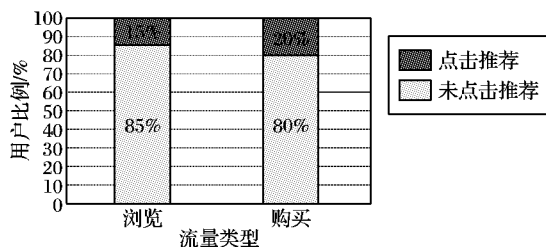


图5 推荐的订单贡献

根据以上信息,可以计算推荐对网站销售的贡献情况。假设:1)推荐未提升总流量,即推荐的流量都来自占用原非推荐流量。2)所有订单的金额相同,都为 P 。若无推荐系统时

访问量为100,根据非推荐转化率1.47%计算,总销售额为1.47 P 。加入推荐服务后,总访问量还为100,有85%为非推荐流量,按1.47%的转化率计算,销售额为1.25 P ,15%为推荐流量,按2.05%的推荐流量转化率计算,销售额为0.3 P ,则加入推荐服务后,总销售额为1.55 P ,相对于无推荐系统时的1.47 P ,提高销售额5.9%。

3 结语

本文提出的基于SaaS模式的电子商务推荐平台很好地解决了中小电子商务网站由于缺乏资源部署推荐系统困难的问题。未来,利用该推荐平台还需要对以下问题进行研究:第一,如何利用大量电子商务网站产生的数据来分析用户的消费行为,指导不同网站的推荐策略;第二,如何在不同站点之间实现交叉推荐,使推荐系统平台成为连接各网站的纽带。

参考文献:

- [1] ZHENG QIN. Introduction to E-commerce [M]. Berlin: Springer-Verlag, 2009.
- [2] SCHAFER J B, KONSTAN J, RIEDL J. Recommender systems in E-commerce [C]// EC '99: Proceedings of the 1st ACM Conference on Electronic Commerce. New York: ACM Press, 1999: 158 - 166.
- [3] LINDEN G, SMITH B, YORK J. Amazon.com recommendations: Item-to-item collaborative filtering [J]. IEEE Internet Computing, 2003, 7(1): 76 - 80.
- [4] SARWAR B, KARYPIS G, KONSTAN J, et al. Item-based collaborative filtering recommendation algorithms [C]// WWW '01: Proceedings of the 10th International Conference on World Wide Web. New York: ACM Press, 2001: 285 - 295.
- [5] KONSTAN J A, MILLER B N, MALITZ D, et al. Grouplens: Applying collaborative filtering to usenet news [J]. Communications of the ACM, 1997, 40(3): 77 - 87.
- [6] RESNICK P, IACOVU N, SUCHAK M, et al. Grouplens: An open architecture for collaborative filtering of netnews [C]// CSCW '94: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work. New York: ACM Press, 1994: 175 - 186.
- [7] BALABANOVIC, MARKO, SHOHAM. Fab: content-based, collaborative recommendation [J]. Communications of the ACM, 1997, 40(3): 66 - 72.
- [8] MOONEY R J, ROY L. Content-based book recommending using learning for text categorization [C]// DL '00: Proceedings of the 5th ACM Conference on Digital Libraries. New York: ACM Press, 2000: 195 - 204.
- [9] PAZZANI M J, BILLSUS D. Learning and revising user profiles: the identification of interesting Web sites [J]. Machine Learning, 1997, 27(3): 313 - 331.
- [10] CLAYPOOL M, GOKHALE A, MIRANDA T, et al. Combining content-based and collaborative filters in an online newspaper [C]// Proceedings of ACM SIGIR Workshop on Recommender Systems. New York: ACM Press, 1999: 40 - 48.
- [11] PAZZANI M J. A framework for collaborative, content-based, and demographic filtering [J]. Artificial intelligence review - Special issue on data mining on the Internet, 1999, 13(5/6): 393 - 408.
- [12] SOBOROFF I, NICHOLAS C. Combining content and collaboration in text filtering [C]// Proceedings of the IJCAI '99 Workshop on Machine Learning for Information Filtering. Stockholm: [s. n.], 1999: 86 - 91.
- [13] PHAM M C, CAO Y-W, KLAMMA R. A clustering approach for collaborative filtering recommendation using social network analysis [J]. Journal of Universal Computer Science, 2011, 17(4): 583 - 604.
- [14] LIPCZAK M, MILIOS E. Learning in efficient tag recommendation [C]// RecSys '10: Proceedings of the 4th ACM Conference on Recommender Systems. New York: ACM Press, 2010: 167 - 174.
- [15] Wikipedia. Software as a service [EB/OL]. [2010-01-07]. http://en.wikipedia.org/wiki/Software_as_a_service.