面向电商数据的可视化查询系统①

陈先灏, 雒江涛

(重庆邮电大学 电子信息与网络工程研究院, 重庆 400065)

摘 要: 目前电子商务已经愈发成熟, 如何有效分析客户潜在价值成为不可忽视的问题. 针对此问题, 本文设计与实现了一种可视化查询系统, 用于分析用户购物消费偏好, 挖掘潜在价值. 本系统基于 SSH 框架, 前端采用 Highcharts 图表展示工具予以数据可视化处理; 后台利用 DPI 提取校园网流量中的电商数据建立用户购物行为数据库, 同时利用网络爬虫建立电商产品信息库. 系统经测试, 能达到预期效果, 对用户购物偏好具有一定利用价值.

关键词: 可视化; 查询系统; SSH; Highcharts; DPI

Visual Query System for Electronic Commerce data

CHEN Xian-Hao, LUO Jiang-Tao

(Electronic Information and Networking Research Institute, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

Abstract: How to analyse the potential value of a customer is a problem that cannot be neglected while the e-commerce is becoming more and more mature. Aiming at solving this problem, the paper designs and implements a visual query system, which is used for analysing customers' preference and digging their potential value. Based on the SSH framework, this system uses Highcharts to do the data visualization at the front-end, while the database of products is established via the web parser and the database of user-expense behavior extracted from the campus's traffic is established via DPI. This system reaches the expected target for data analysis and is good for shopping preference analysis after testing.

Key words: data visualization; query system; SSH; Highcharts; DPI

电子商务作为新一代商业模式, 为客户提供了便捷、高质量的服务体验. 目前电商业务已经从传统 PC 平台向移动平台迁移, 且愈发成熟. 作为电商服务提供者迫切需要分析客户购买偏好, 随时掌握市场动态. 因此, 如何有效分析大量电商数据, 具有非常重要的意义.

京东, 淘宝等电商相继建立数据挖掘品台, 它们拥有相当丰富并且完善的用户交易数据, 可以做到精准营销, 但对于新型电商而言, 数据分析较为不易.

针对上述问题, 本文在传统的查询系统的基础上, 阐述了利用网络流量进行数据存储和分析, 重点阐述电商数据前端与后端的信息传递方式和数据可视化, 实现了面向电商数据的可视化查询系统.

1 系统分析

1.1 基本思路

面向校园网流量的可视化查询系统旨在通过解析用户网络流量, 对用户的购物事件进行统计分析, 能全面真实的反应购物偏好.

本文主要关注学生网购行为, 而用户购物的一系列行为均在 HTTP 请求内容中有所体现. 所以分析 HTTP 数据包中相关内容, 进而得到用户购物行为[1].

以一个数据包为例, 可分析得知 URL 中有显著的标识字段以及商品 ID 字段, 通过商品 ID 建立与商品数据库的联系, 可以获得用户购物详细数据.

确定商品与 ID 的对应关系是数据分析的关键. 本文中采取的方案是利用网络爬虫构建信息库.

① 基金项目:重庆市应用开发计划(cstc2013yykfA40006)
收稿时间:2015-09-22;收到修改稿时间:2015-11-27

最后利用可视化思路, 进行电商数据可视化查询.

1.2 系统目标

基于以上思路, 系统的主要目标是实现面向电商数据的可视化查询系统. 系统利用 DPI 进行用户数据分析, 建立用户行为数据库, 并且利用爬虫建立商品数据库, 然后通过 SSH 构建前台与后台之间的数据传输通道, 最后在浏览器中利用 Highcharts 呈现图形化数据. 系统用户可以通过本系统分析目前消费者在各大主流电商网站上的消费情况.

1.3 可视化查询过程模型

图 1 给出了可视化查询过程模型, 包括 4 个主要过程, 分别概述如下:

(1)数据采集: 包括两方面, 一方面采集校园网学生用户数据, 为后续电商数据的提取以及购物分析提供原始数据; 另一方面采集商品数据以便对应学生购买的商品, 获得详细信息;

(2)数据解析: 根据前文思路, 基于学生购买商品事件, 系统对原始数据流量进行分析和重组, 建立针对每一位学生的消费事件表. 消费事件表结合商品信息库, 完成信息补充, 获得详细数据表;

(3)数据表存储: 将数据表存入数据库, 对已经解析完毕的原始流量予以删除, 缓解存储压力并提供日志记录;

(4)数据提取以及可视化: 系统根据不同用户, 以及不同需求, 查询数据库, 获得所需多维信息数据, 并返回客户端, 以各种图形呈现.

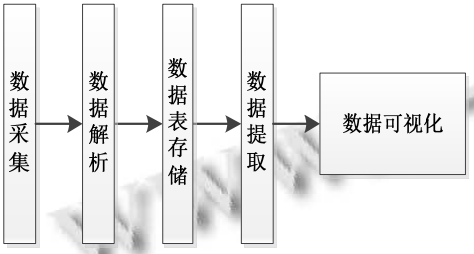

图 1　系统过程模型

2　系统总体设计

按照过程模型, 系统总体架构分为四层, 从上至下依次为用户层, 应用展示层, 逻辑层, 和数据层. 系统总体框架结构图如 2 所示.

用户层是本系统面向不同用户所呈现的具体系统页面. 系统主要面向一般用户以及管理员.

应用层以浏览器为媒介, 为用户提供数据管理以及可视化展示分析, 并且可以进行图形转换处理.

数据管理主要包括系统定时采集网络流量进行两部分数据更新, 并为用户提供通知, 同时系统用户可以手动选择是否删除以往数据.

统计分析功能依靠前端提供的数据搜索功能查询底层数据库.

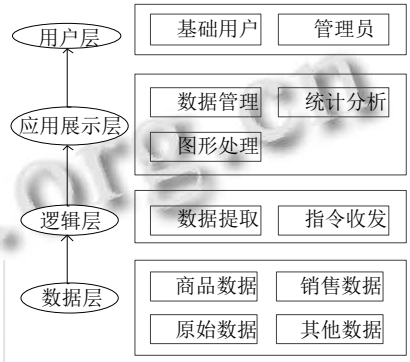

图 2　系统总体架构图

图形处理依靠前端可视化工具针对同一份数据做多图形展示, 便于用户选择以及人工分析.

逻辑层的设计主要通过 Struts2 和 Spring 框架来完成. 通过在项目中引入 Spring 框架, 完成 Service, Action, DAO 及依赖关系的管理[2]; 逻辑层完成用户在应用层下达的指令收发(包括数据收发, 数据更新, 以及数据传递). 数据传递利用 Hibernate 完成.

Hibernate 基于对象-关系映射, 即可以将对象与关系数据库进行对应, 生成相应 SQL 查询语句. 系统利用 SQL 语句查询相应结果, 并将获得的数据转为 JSON 格式, 作为响应提交给应用层以及用户层.

数据层提供数据采集, 以及数据存储等功能. 数据层采集存储结构图由图 3 所示.

如图所示, 网络爬虫完成产品信息提取以及更新; DPI(Deep Packet Inspection)完成消费者购物信息的提取.

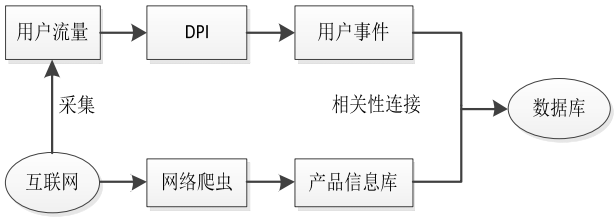

图 3　数据采集存储结构图

网络爬虫程序每次读取一个 URL, 下载其对应的

网页中的产品信息，并对其分析．若分析得到新的URL，则重复这一过程，直到满足一定条件，方可停止．

本系统中，网络爬虫定向抓取电商网站中的资源，建立产品信息库．信息库主要包括商品、商品 ID 以及商品描述等，与 DPI 模块相呼应．

DPI 即深度包检测技术是一种基于应用层的控制技术和流量检测技术，能够高效地识别出各种应用．普通报文检测通过端口号来鉴定应用类型．而当前网络上的非法应用会通过隐藏或假冒端口号的方式躲避．在这种情况下，L2~L4 层的传统检测方法已然失效．不同于传统方法，DPI 通过对网络流中的数据报文内容进行检测，从而确定数据报文的真正应用．

DPI 作为流量分析工具，其作用是分析底层流量，并利用数据特征对每个用户的流量进行分流，统计相关事件．系统利用此信息与产品信息库进行匹配，进而发现用户浏览电商网站详细行为，比如所浏览的产品，网站，时间等．

3 主要模块实现

3.1 数据表

网络爬虫按照 URL 将商品信息分网站，分类别，进行爬取，获得产品信息库并予以存储．产品信息主要包括网站名称，商品 ID，商品描述，一级目录，一级目录描述，二级目录，二级目录描述，三级目录，三级目录描述．如图 4 所示．

web_name	goods_id	goods_des	cat1	cat_des1	cat2	cat_des2	cat3	cat_des3
京东	768678	索尼SONYDSC-I	652654	摄影/摄像	831	数码相机	16538	索尼（SONY）
京东	1148061	索尼SONYDSC-I	652654	摄影/摄像	831	数码相机	16538	索尼（SONY）
京东	920648	索尼SONYDSC-I	652654	摄影/摄像	831	数码相机	16538	索尼（SONY）
京东	676676	索尼SONYDSC-I	652654	摄影/摄像	831	数码相机	16538	索尼（SONY）
京东	1082020	索尼SONYDSC-I	652654	摄影/摄像	831	数码相机	16538	索尼（SONY）

图 4 产品信息库

DPI 根据用户流量数据包，解析出关键信息：访问时间，访问网站，商品 ID，源 IP，目的 IP，源端口以及目的端口，如图 5 所示．源 IP，可以用来确定属于同一用户的数据包，从而定位其整个会话过程的网站浏览情况．

当数据库系统接受到查询指令，系统将用户事件信息，与产品信息库进行关联，得到用户在某段时间的具体浏览情况．

visit_time	web_name	goods_id	src_ip	des_ip	src_port	des_port	id
1388577305	淘宝	35411203986	113.250.154.47	182.140.238.140	57740	80	1
1388577306	淘宝	35829967598	113.250.153.1	182.140.238.150	63954	80	2
1388577349	淘宝	36341032674	113.250.154.47	182.140.238.140	57740	80	3
1388577369	淘宝	35529329253	113.250.154.47	182.140.238.140	57740	80	4
1388577391	天猫	35352667078	113.250.157.185	182.140.238.150	53057	80	5
1388577417	淘宝	5072407877	113.250.154.47	182.140.238.140	57879	80	6

图 5 用户浏览信息表

3.2 后台查询模块

本系统后台查询模块主要完成数据查询以及数据组装．考虑到用户有不同需求，数据组装功能实现了灵活显示部分字段、记录．与此同时，数据组装功能也包括数据格式转换．

后台代码根据前台请求，查询所需字段对应的数据．此过程为了简化代码设计，可以在 HTTP 请求中将所查询的参数按照 SQL 语句格式排列好后一并传入后台，后台只需读取上述已整理好的参数语句即可．简化代码如下：

```
.声明 sql 组装
public static String sqlCombing(String httpRequestAttr-ibutes, String httpRequestCondition){
.声明系统接收到的 HTTP request
HttpServletRequest request = ServletActionContext.getRequest();
.得到前台传入参数
String sqlAttributes = request.getParameter(httpRequesAttributes);
.得到前台查询条件，比如大于，小于某一个值
String sqlConditon = request.getParameter(httpRequestCondition);
.组装语句
String theSQLCombined = “select” + sqlAtttributes +sqlCondition;
```

查询模块将查询条件转为标准 SQL 语句之后，系统执行查找，得到所需数据并将其转为 JSON 格式，以 HTTP Response 的形式返回客户端．

3.3 前端模块

3.3.1 JQuery

jQuery 是一个优秀的 JavaScript 框架，该框架实现了 HTML、JavaScript、CSS 三者分离．凭借其优秀的页面效果以及交互性，已经被广泛使用；同时凭借其使用的简洁性，对 DOM 强大的操控性和易扩展性受

到 Web 开发人员的喜爱[3]. 本系统 JS 代码均有借助 JQuery 库, 方便 Ajax 交互技术.

3.3.2 页面数据请求

前台页面完成输入参数的组装问题以及请求的提交, 以下举例说明: 根据 3.1 所述, 商品属性包括网站包含 9 个属性, 所以页面上可供选择的属性有 9 个, 为了方便用户选择, 所有属性均有说明以及复选框, 部分相关 HTML 代码如下:

网站名称: <input id="web_name" type="checkbox" class="interest" name="interest" value="web_name">

商品描述: <input id="goods_des" type="checkbox" class="interest" name="interest" value="goods_des">

一级目录: <input id="cat_des1" type="checkbox" class="interest" name="interest" value="cat_des1">

二级目录: <input id="cat_des2" type="checkbox" class="interest" name="interest" value="cat_des2">

如上所示, 每一个属性的 type 均为 checkbox(复选框), 另外, name 均设置为 interest, 这种设置为 jQuery 选择器提供便利.

客户端得到所需要查询的字段, 通过 jQuery 自带的$.get(), 即可以 ajax 的方式, 异步请求数据.

3.3.3. Highcharts

数据可视化模块采用的是 Highcharts. Highcharts 是一个用纯 Javascript 图表库, 能够很便捷地在 web 网站或是 web 应用程序中提供直观, 交互性的图表[4].

Highcharts 引入数据方法及其简便, 只需要将组装好的数据赋予其 data 配置量即可.

3.3.4 可视化展示

系统前端页面利用 jQuery 异步请求得到的 JSON 数据, 经解析后作为传入 highcharts 的数据. 利用此方法以及 javascript 事件函数, 本系统完成了数据 3 层钻取并画图, 利于数据全面分析. 关键代码如下:

```
var onSuccess = function(response){
    var resHander = response.responseText;
    .解析 JSON 数据
    var resJson = Ext.decode(resHander);
    }
```

Highcharts 接收到上述数据后, 进行画图, 以下代码完成二级数据钻取:

```
    series:[{
    data: theJSONResults,
    events:{
    click:function(event){
    window.open("countcat3redirect.action?" +
    "catDes2="+cate1.category)
    .点击每一块数据打开一个网页以显示下一层数据.
      }
   }
}]
```

4 实验分析

4.1 图例分析

本文面向电商数据, 实现了基于时间段内的用户购物数据可视化分析. 综上所述, 系统用户可以清楚地分析出在某一时间段内学生在不同电商品牌的购物情况.

本系统实现了以条形统计图, 饼图, 折线图为主的数据显示方式, 并且所绘制图形提供下载, 保存, 同时数据图形本身提供对应的 excel 表, 方便系统用户使用.

系统为了更加直观地展示所有系列商品销售情况, 在上级视图中, 添加数据钻取功能. 以下举例说明:

如图 6 所示, 系统针对某时段采集到的数据, 绘制出淘宝商城本时段内销售的商品种类. 用户通过图形结果, 可以较为直观的看到服装类基本占据了销售的主导地位, 其次为鞋包. 基于以上结论, 基本可以肯定学生顾客在基本穿着方面消费较多.

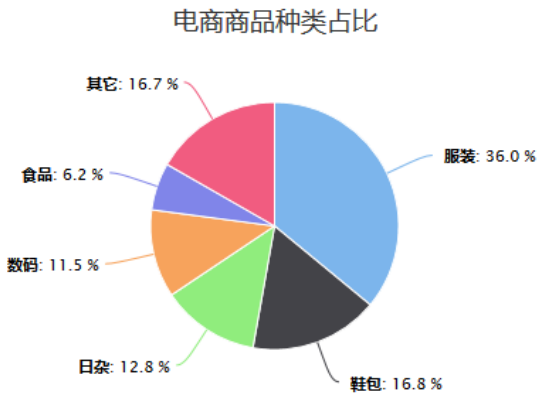

图 6 数据分布饼图

不过基于以上结论, 信息量并不足够, 用户需要知道更加详细的品牌信息以确定更加精确的销售策

略.

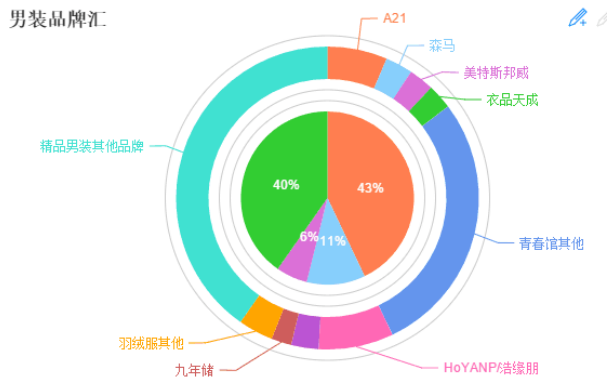

图 7 男装品牌分布饼图

用户可以通过点击图 6 中的服装, 展示出目前数据所采集到的相关男装以及女装品牌, 并提供相应视图, 如图 7 所示. 系统显示出了男装各大品牌的销售情况, 销售商可以根据不同时段的不同数据, 紧跟当下潮流, 使用相应的供货手段, 尽可能保持盈利.

4.2 系统展示优化策略

由于数据量大, 系统采取以下措施, 作为基本优化策略:

采用 ajax 方式, 即利用异步读取方式. 当客户需要更新页面部分图例, 采取局部刷新的方式, 减少用户请求数据量.

SQL 语句优化, 即尽可能减少表之间的连接, 建立合理索引, 以提高查询速度; 同时采取数据分页显示, 以缓解压力.

5 结语

本文针对电商产品数据以及用户购买行为, 设计了一种基于 SSH 架构的可视化查询系统. 后台通过网络爬虫以及 DPI 技术建立数据库, 具有数据覆盖面全的特点; 系统使用 B/S 架构, 具有安装, 部署方便等特点, 每一层次都实现了灵活的接口, 便于 2 次开发.

系统将进一步研究原始数据采集中的过滤模块, 减少原始数据存储压力, 以加快定时更新频率; 以及产品数据可视化的集成, 尝试实现系统调用外部数据集接口, 实现数据灵活的特点.

参考文献

1 杨军超,雒江涛,申健,邓生雄.基于 MapReduce 的校园网用户网购偏好分析.计算机系统应用, 2015,24(10):222–226.
2 张建军,刘虎,倪芳英.基于 SSH 与 Highcharts 整合架构的 Web 应用研究.计算机技术与发展, 2013,9:245–247,251.
3 周玲余.基于 jQuery 框架的页面前端特效的设计与实现.计算机与现代化,2013,1:61–63.
4 吴孟春,丁岚.HighCharts 组件在气象业务中的开发和应用.计算机与网络, 2014,12:65–68.