基于向量的点击流聚类算法①

徐 昊, 谢文阁

(辽宁工业大学 电子与信息工程学院, 锦州 121001)

摘 要: 点击流数据是分析互联网用户心理倾向的关键, 互联网用户的聚类可以通过分析点击流数据实现. 本文提出了一种基于向量的相似度计算方法, 将点击流数据转化为向量数据. 通过对向量的计算来得出聚类的结果. 算法克服了传统的聚类算法的一些缺点, 更能符合研究人员研究 Web 点击流数据时关于个性化聚类的要求.

关键词: 点击流; 余弦相似度; 聚类算法

Clickstream Clustering Algorithm Baced on Vector

XU Hao, XIE Wen-Ge

(School of Electronic and Information Engineering, Liaoning University of Technology, Jinzhou 121001, China)

Abstract: Clickstream data are the key to analysis the psychological tendency of Internet users. Internet users clustering can be realized by analyzing the clickstream data.This paper presents a similarity calculation method based on vector. Clickstream data can be converted to vector data.Through the calculation of vector clustering, results could be obtained. Algorithm overcomes the drawbacks of the traditional clustering algorithm. It can meet researchers study of web clickstream data about the requirements of personalized clustering more.

Key words: clickstream; cosine similarity calculation; clustering algorithm

1 引言

当前, 基于网络的个性化服务越来越多, 为消费者提供了多样的选择. 网络用户个性化要求能否得到满足, 被认为是互联网产品能否成功的关键. 准确的识别出每一个网络用户的浏览习惯, 并为之提供个性化的服务, 可以优化用户使用体验, 提高用户忠诚度. 点击流是用户访问网站时, 浏览的网页所形成的序列. 具有相似喜好和浏览习惯的用户, 在点击流序列中就会表现出某种相似的特征. 通过点击流聚类可以实现对用户的归类. 存储于服务器上的 Web 日志, 是了解互联网用户心理倾向的重要数据源. Web 日志中存储了详细的用户访问资料, 但它不能被直接用于数据挖掘, 必须通过 ETL 工具转化为数据库数据或其他易于分析的数据格式.

传统的聚类算法以 K-MEANS 为代表, 在处理点击流数据时, 有以下缺点: ①需要用户合理估计簇内数目. ②不具备增量挖掘功能. ③只能将数据点严格划分到一个簇中[1,5]. 针对于点击流数据的算法当中, Fu Yongjian, Sandhu K 等利用 URL 对会话中的网页进行归一化处理, 再用 BIRCH 算法聚类, 该方法有效降低了聚类所用的特征向量的维度[2]; 李晓明等提出的 WCSCluster 利用了公共访问序列中的访问时间作为距离的度量[1]. 前面所述的两种方法, 或实现起来十分复杂, 或局限于公共的访问序列, 无法全面覆盖点击流数据的特征. 本文将点击流表示为向量, 利用向量间夹角的大小来判定点击流的相似度. 实验证明, 该算法具有较好的聚类效果.

2 点击流聚类

点击流是用户访问 Web 站点时的动作记录. 它可以被组合到会话中, 形成微观操作时间序列. 通过对点击流的分析, 我们可以更好地理解用户的需求, 并

① 收稿时间:2013-09-28;收到修改稿时间:2013-10-15

© 中国科学院软件研究所 http://www.c-s-a.org.cn

对此做出响应. 聚类是将数据划分成群组的过程. 它主要研究如何在没有训练的情况下, 仅仅根据数据自身的特征, 将数据划分成若干类.

本文将点击流数据看做向量, 按照向量的方式来计算点击流数据的相似度[3]. 由于向量具有方向性, 因而可以根据向量间的夹角大小来判断向量的相似程度. 夹角愈小, 我们就认为其愈相似. 由于点击流向量每个维度的取值只可能为大于等于零的整数, 因此, 向量的夹角范围只能在 0°～90° 之间.

点击流聚类方法有两种: 第一种是把点击流看作具有进化特性的数据流, 在只进行单次或较少次数扫描情况下, 以较少内存开销, 实时完成聚类. 第二种是把点击流看作静态数据(主要是存储于 WEB 服务器之中的日志数据), 以牺牲一定处理速度为代价, 获得更加准确的聚类结果[2]. 本文采用的是第二种方法来收集点击流数据. Web 日志文件一般有三种: 服务器端日志文件、代理服务器端日志文件、客户端日志文件. 一般来情况下, 我们通常使用的是第一种文件格式. 原始日志文件数据并不能直接用来进行 Web 挖掘, 通常需要将其进行日志预处理操作. 常见的日志数据格式有: W3C 联盟规定的常规日志格式 CLF 和扩展日志格式 ECLF[4].

下面我们将根据上述内容, 给出算法所需概念的定义.

定义 1 点击流序列:用户访问 Web 站点时的一组连续的访问序列, 称作点击流序列, 记作 <P1,P2,P3,……,Pn>, 设 I 为点击流访问全集, Pi∈I, 0≤i≤n.

定义 2 点击流维度: 将用户访问的点击流页面按照一定的规则分为 m 个类,则称这 m 个分类为点击流维度 M. 例如, 可以将访问/sports/目录下的所有页面划归为 sports 类, 将访问/music/目录下的所有页面划归为 music 类.

定义 3 点击流向量: 将用户的点击流访问序列按照点击流维度 A 的分类规则映射得到的向量, 即为在点击流维度 A 下的点击流向量. 例如, 我们共划分了三个点击流类甲, 乙, 丙, 用户 1 的点击流访问了甲类 2 次, 乙类 1 次, 丙类 0 次, 则用户 1 的点击流向量为 (2, 1, 0).

定义 4 余弦相似度: 设 α, β 为点击流维度 M 下的两个点击流向量, 设阈值为 λ. 用公式

$$\cos\theta=\alpha\beta/(|\alpha||\beta|)$$

来计算点击流向量 α, β 的夹角大小. 如果两个向量的夹角小于某一阈值 θ, 即 >λ, 则称 α, β 是相似的点击流向量.

定义 5 簇特征向量序列: 用于标识簇特征的向量序列. 通过簇特征向量序列, 可以计算点击流与簇的相似度, 进而判定点击流是否应归入该簇内.

3 算法研究与实现

3.1 算法描述

本文采取了面向对象的方法来描述整个算法过程. 具体分为点击流类, 簇类, 数据更新类, 主类四个类. 点击流类用来存储点击流的向量信息和用户名称. 簇类用来存储簇的特征向量序列, 簇内点击流个数及每个点击流的用户名称. 数据更新类用来计算余弦相似度并负责更新簇类信息. 主类将负责协调执行程序. 算法中起核心作用的是点击流类, 簇类和数据更新类.

算法的过程为: 首先初始化数据库连接, 创建点击流类对象, 簇类对象, 数据更新类对象. 取第一个元组值赋值给点击流对象 a, 将点击流对象 a 的向量序列赋值给第一个簇. 然后顺序地取出数据库的其他元组, 赋值给点击流对象 a, 计算 a 与每一个簇的相似度. 相似度的计算根据定义 4, 如果两个向量的夹角小于一定的数值, 即余弦相似度公式所算出的值大于 λ, 则将其划归到该簇内. 如果与所有的簇计算过后, 该点击流数据仍没有加入到一个簇中, 则新建一个簇, 将该点击流的序列赋值给新建的簇. 如此循环往复, 直到所有的点击流数据都被划分为止.

对于点击流维度的划分, 要注意结果精度与可分析性之间的平衡. 如果维度很多, 则有可能造成每个点击流数据独占一个簇, 失去了分析的意义. 如果维度太少, 则大量的点击流被分到了几个簇内, 没有太高的分析价值. 考虑到一个大型的站点其页面成千上万, 如果将每一个页面都作为一个维度, 则有可能导致这样一种结果: 算法划分了大量的簇, 而且每个簇只有一个点击流数据. 那么这里就可以采用分类的方法, 将网站中的页面按类别划分, 如军事、体育、音乐等, 划分的程度可视具体情况而确定.

对于 λ 的取值, 同样也会关系到聚类的精度. 对于大部分的应用来说, 我们希望每个簇的边界划分能够使得每个数据点都只属于一个簇, 即全部的点击流

© 中国科学院软件研究所 http://www.c-s-a.org.cn

数据数目正好等于每个簇内的点击流数据之和. 由于 λ 的域为(0,1), 且随着角度的增加, 余弦值单调减少. 因此可以采用折半查找的方法, 先令 λ =cos45°, 计算是否满足条件. 如果满足条件, 则继续缩小角度, 令 λ =cos67.5°; 如果不满足条件则扩大角度令 λ =cos22.5°. 如此反复, 直到查找到满足条件的临界值为止. 当然, 采用这种方法计算的 λ 为一个临界值, 由该值而得到的聚类也不免缺乏弹性. 可以用此值作为参考, 适当调低 λ 的取值.

点击流聚类算法的流程图如图 1 所示.

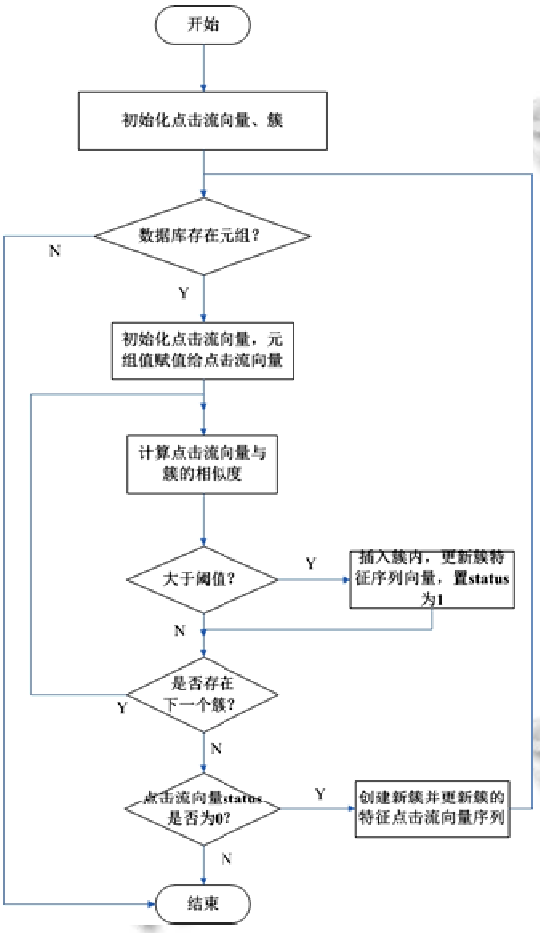

图 1　点击流聚类算法流程图

有关于点击流数据的聚类算法中, 各种方法的主要差别在于点击流数据的表示方法和距离的度量方法. 本文将点击流数据表示为向量, 使得所有的页面数据被划归到向量之中, 模型的实现并不复杂, 有关于簇特征向量序列的计算相较于 BIRCH 的 CF 树构建也省时许多. 相比于 K-MEANS, 本文所提算法并不需要指定聚类簇的数目, 并且在实现算法之中, 可以一次性地将所有数据输入并有效归类(特征存储于簇的特征向量序列之中). 也正是由于簇特征向量的存在, 因此可以进行增量挖掘. 对于最后聚类簇的划分, 可以根据 λ 的取值, 来确定某个点击流向量是否会被划分到某一个簇中, 本算法并不严格将其划分到某一个簇中, 而是有可能会被划分到多个满足条件的簇中. 在当前的 WEB 站点中, 访问量十分巨大, 如果算法的复杂度太大, 也难以形成效益. 文所述算法实现简单, 核心代码少. 对于聚类所得的用户, 并不严格区分到某一类中(通过 λ 取值实现), 因此可以适应复杂的互联网环境.

3.2 算法实现

以下为程序的核心伪代码

```
public class 点击流聚类项目 {
  public static void main(String[] args) {
  ClickStream clickstream=new ClickStream();
  UpdateData updatedata=new UpdateData();
  Cluster cluster[]=new Cluster[X];
  con=DriverManager.getConnection();
  sql=con.createStatement();
  rs=sql.executeQuery(SQL 语句);
  rs.next();
  while(数据库中存在元组){
    for(所有的簇){
      if(updatedata.similar>= λ ){
        updatedata.updatecluster();
        clickstream.点击流状态=1;}
      if(clickstream.点击流状态==0){
        i=i+1;
        updatedata.updatecluster();} }}
class ClickStream{
  点击流名称;
  点击流向量维值;
  点击流状态;}
class Cluster{
  簇特征向量序列;
  簇内点击流名称;
  簇内点击流数目;}
class UpdateData{
  double similar(ClickStream,Cluster){
  double value;
```


© 中国科学院软件研究所 http://www.c-s-a.org.cn

value=余弦公式计算所得数值;
return value;} //计算点击流数据于簇的相似度
updatecluster(ClickStream,Cluster){
cluster.number=cluster.number+1;} }

4 实验与分析

4.1 实验数据与实验环境

实验数据为 NASA-HTTP, 来源网址为(http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html). 其格式为(host, timestamp, request, HTTP reply code, bytes in the reply), 格式含义为, host 代表请求主机, timestamp 代表请求时间, request 代表请求页面, HTTP reply code 代表返回代码, bytes in the reply 代表字节返回数.

本文所使用的算法程序全部在 PC 机上实现. 电脑配置: CPU Inter T2450, 内存 3GB, 硬盘 500GB, 操作系统为 Microsoft Windows 7 专业版 SP1, ETL 工具为 VS 2005 BI 工具,编程语言为 JAVA, IDE 为 Netbeans. 数据库采用的 Microsoft SQL Server 2005. 经过处理, 可得到 15952 条点击流数据.

4.2 结果分析

实验中将页面分为了 5 个类别: software, htbin 等目录下的页面访问被划为 a; shuttle 页面被划为 b; images 被划为 c; history 被划为 d; 其余访问被划为 e. 为简化起见, 本次实验抽取了数据源中的 200 条数据进行计算.

根据第三节所述的算法, 利用折半查找, 找到了当 λ =0.95 时, 全部的点击流数据数目正好等于每个簇内的点击流数据之和.

在计算过程中, 我们可以得到有关于向量夹角大小, 也即 λ 的取值和聚类数目的关系, 如表 1:

表 1 λ 的取值和聚类数目的关系

λ 取值	聚类数目
Cos0°=1	32
Cos10°=0.9848	20
Cos20°=0.0.9396	12
Cos30°=0.8660	10
Cos40°=0.7660	8
Cos50°=0.6428	7
Cos60°=0.500	7
Cos70°=0.3420	7
Cos80°=0.1736	7
Cos90°=0.0000	1

通过上表可见, 随着对于阈值夹角的扩大, 聚类的数量越来越少, 点击流数据被聚类到若干个大的簇中; 而当夹角减小时, 聚类的数量开始变多. 不同的阈值对应了不同的聚类结果.

根据实验所得结果, 全部数据共分成了 16 个簇. 簇 1 的特征向量序列为(0,0,1,1,0), 簇内共有 4 个对象. 那么这四个对象在访问 c, d 类页面上具有共同的倾向性, 即他们都比较喜欢访问 images 目录和 history 目录下的文件, 且没有其他的页面访问爱好. 簇 2 中的对象个数为 32 个, 特征向量序列为(0,1.125,0,0,0), 该结果说明这 32 个对象只喜欢浏览 shuttle 页面下的文件, 而不喜欢浏览其他页面. 其余结果的分析可参考上述分析.

5 总结

互联网的环境十分复杂, 对于用户的聚类方法也各有不同. 传统的聚类算法对点击流数据的处理不尽如人意, 需要制定簇的数目, 不能增量挖掘, 数据被严格的划分到一个簇中. 而对点击流数据较为有效的算法 BIRCH, 则实现比较复杂, 需要多次迭代才能得出满意的结果.

本文所提出的算法将点击流数据表示为向量, 利用向量之间的夹角大小作为距离的衡量尺度. 通过实验的结果, 我们可以看到, 随着阈值的变化, 聚类的簇也在不断的变化. 某一个点击流向量可能会被同时划分到不同的簇中. 同时, 可以通过折半查找, 寻找到使得聚类数目最大化的 λ 的值. 实验结果表明, 随着 λ 值的增大, 聚类的数目也越来越多. 在聚类好的簇中, 我们可以通过簇的特征向量序列, 来分析簇内成员的共同点. 在具体的应用中, 可以根据实际情况调整 λ 的取值, 以得出满意的聚类结果.

参考文献

1 李晓明.Web 点击流数据的聚类技术研究[硕士学位论文].沈阳:东北大学,2009.
2 马超,沈微.基于闭合有间隔频繁子序列的点击流聚类.计算机工程,2010,36(23):72–75.
3 刘嘉,祁奇,陈振宇,惠成峰.ESSK:一种计算点击流相似度的新方法.计算机科学,2012,39(6):147–150.
4 Kimball R, Merz R. 张丽萍等译.Web 数据仓库构建指南.北京:清华大学出版社,2005:1–60.
5 陈燕.数据挖掘技术与应用.北京:清华大学出版社,2011:134–149.

© 中国科学院软件研究所 http://www.c-s-a.org.cn