Web日志分析系统研究

阳小兰,钱 程,赵海廷
(武汉科技大学 中南分校 信息工程学院,湖北 武汉 430223)

摘 要:Web日志分析系统不仅能改进Web网站结构,提高Web服务器性能,而且能识别用户的喜好、满意度,发现潜在用户,增强网站服务竞争力。介绍了Web日志挖掘的各个阶段,设计并实现了一个Web日志分析系统。分析了传统的频繁项集挖掘算法与序列模式挖掘算法的不足之处,根据日志数据的特性,将用户属性引入频繁项目集的生成过程,有效地减少了候选项集的数目,并根据候选集的特点,逐轮压缩数据库。将连续序列引入到ApiroriAll算法的候选集合并过程中,实现了改进算法。通过实验比较了改进算法与传统算法的效率,证明了改进算法的有效性。

关键词:日志分析;数据预处理;频繁项目集;序列模式
中图分类号:TP393 **文献标识码**:A **文章编号**:1673-629X(2011)09-0211-05

Research on Web Log Analysis System

YANG Xiao-lan, QIAN Cheng, ZHAO Hai-ting
(College of Information Engineering, Zhongnan Branch, Wuhan University of Science and Technology, Wuhan 430223, China)

Abstract: Web log analysis system can not only improve the Web site structure and improve Web server performance, but also identify the user's preferences, satisfaction, identify potential customers and enhance the competitiveness of Web services. The stages of Web log mining are described, and a Web log analysis system is designed and implemented. The shortcomings of traditional frequent itemsets mining algorithm and sequential pattern mining algorithm are analyzed. According to the characteristics of log data, the user attributes are added into the generation process of frequent item sets, effectively reducing the number of candidate items. According to the characteristics of the candidate set, by round of compressed database. ApiroriAll continuous sequence introduced into the algorithm and the process of candidate set. An improved algorithm is implemented. In the experiment, the efficiency of improved algorithm and traditional algorithm is compared, the effectiveness of the improved algorithm is proved.

Key words: log analysis; data preprocessing; frequent itemsets; sequential patterns

0 引 言

在过去几年中,网站的数目、大小及复杂度都以一个惊人的速度增长着,万维网已经成为世界上规模最大的公共数据源。与此同时,Web网站和Web服务器设计的复杂度以及浏览网站的复杂性随之增加。如何设计网站的一个重要依据就是分析网站该如何使用。网页的使用分析包括简单的统计分析,比如页面访问频率的统计,更复杂分析形式如通过网站寻找共同的遍历路径等。这些网站使用信息还可以用来重构网站,以便给用户提供更好的服务[1~4]。

Web使用挖掘又叫Web日志挖掘,其数据源为Web服务器访问日志,通过挖掘相关的Web日志记录,来发现用户访问Web页面的模式。通过对Web日志的分析,不仅能改善Web站点的性能,改进Web站点的设计,还能根据用户过去的访问模式,预测用户将来的访问,给用户提供个性化的服务及开展商业智能服务[5,6]。

1 系统设计

数据挖掘主要分为数据的预处理、模式挖掘和模式分析几个阶段。数据预处理的主要任务是从原始Web日志记录中提取有用的日志数据并将数据转换为模式挖掘中所需要的类型。模式挖掘阶段根据挖掘的目的,如分类、聚类、关联规则发现或者序列模式分析等决定选择何种挖掘算法。将模式挖掘发现出来的模式进行分析,剔除冗余和无关模式[7~9]。

Web数据挖掘系统模块设计如图1所示。

1.1 数据预处理模块

在开始数据预处理之前,必须完成原始数据的收

收稿日期:2011-03-02;**修回日期**:2011-06-25
基金项目:湖北省自然科学基金项目(2010CDB11102)
作者简介:阳小兰(1978-),女,湖北广水人,硕士,讲师,研究方向为数据挖掘、软件工程。

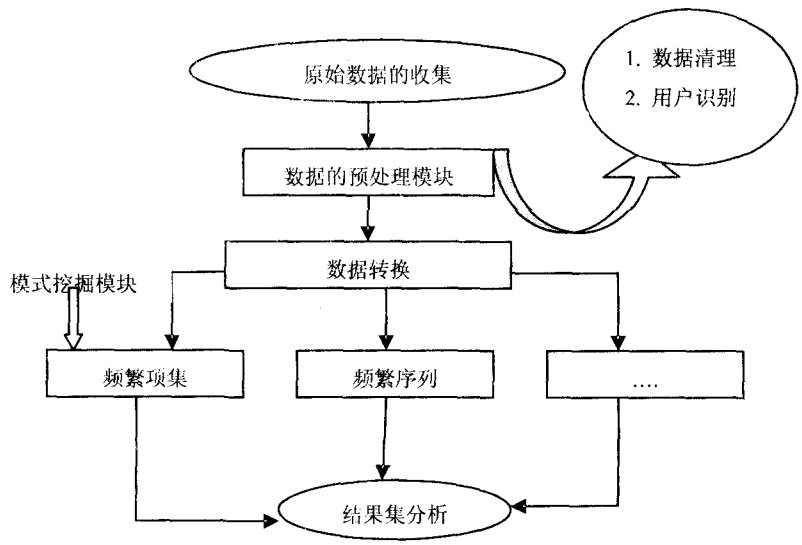

图 1 Web 挖掘系统模块图

集,将原始的 Web 日志数据导入数据库中,建立原始 Web 访问日志数据表 WALS(Web Access Log Set)[10]。Web 访问日志数据表由 id, ip, identd, user, time, url, status, size, refer, agent 几部分组成。

数据的预处理主要包括数据清理,用户识别,会话识别,路径填充这四个任务[11,12],流程图如图 2 所示。

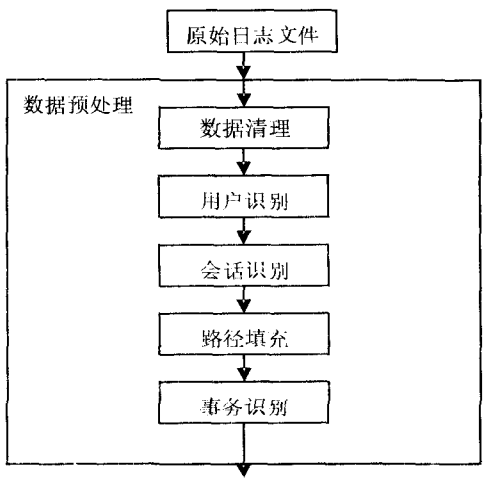

图 2 数据预处理流程图

1.2 模式挖掘模块

模式挖掘模块通常采用 Apriori 算法,Apriori 算法中用于频繁项目集生成部分的算法是基于逐级搜索的思想。它采用多轮搜索的方法,每一轮搜索扫描一遍整个数据集,并最终生成所有的频繁项目集。Apriori 算法常常会产生数量巨大的项集,随着序列模式的长度增加,候选序列会成指数级增长,虽然通过 Apriori 的向下封闭属性缩小了候选项集的大小,但是算法的时间复杂度仍然达不到比较理想的程度。另外,该算法需要多次的扫描日志数据库,只要候选序列的长度增加,就必须扫描一遍数据库,这样对整个算法的执行效率有很大的影响。

文中对算法进行了一些改进。改进主要体现在两个方面,一方面是缩小候选集的大小,另一方面就是缩小事务数据库的大小。具体算法的改进在第 2 部分详细阐述。

1.3 数据库设计

在本系统中,主要涉及到下面几张数据表:WALog 表,存储原始 Web 访问日志。WALog 表中的所有记录按时间顺序依次存储,主键 id 字段自增,IP 字段表示用户访问的 IP 地址,Date 字段是用户访问该页面的访问时间,Method 字段取值为“GET”或者“POST”,URL 字段存储用户访问的资源,Version 字段是 HTTP 协议的版本号,Status 字段表示该次访问是否成功,Bytes 字段表示发送的字节数,Refer 字段为引用页面值,BrowserOS 是指存储浏览器类型和操作系统类型。CWALS 表,数据清理后的 Web 访问日志表,CWALog 表中包含 ID, IP, Date, URL, Refer, Browse 字段,其含义与 WALog 表中相同。UILog 表,用户识别后的日志数据表,包含 UID, IP, Date, URL, Refer 字段,其中 UID 为每个用户的唯一标识,其他字段意义不变。USILog 表,用户会话识别后的日志数据表,包含 USID, Date, URL, Refer 字段,其中 USID 为每个用户会话的唯一标识。PSLog 表,路径填充后的 Web 日志表,包含 USID, Date, URL, Rlength 字段,其中 Rlength 为页面引用长度,即用户在每个页面上的花费时间。CPS 表为内容页面表,包含 ID 和 URL 字段,该表为一个辅助表,用于路径填充阶段。FILog 表,存储频繁项目集,其中 Itemsets 存储频繁项目集合,ID 字段表示该项目是属于 ID-频繁项集。SPLog 表,序列模式挖掘后存储序列模式,Sequences 分别表示频繁序列,ID 字段表示该序列属于 ID-频繁序列。

2 模式挖掘算法及改进

2.1 频繁项目

在日志数据预处理中,Web 日志预处理与传统数据预处理的不同点就是用户识别过程。在候选项集的生成过程中,考虑每个序列 UserID 属性,每个候选项集都会有与之相应的一些 UserID,在候选项集的合并过程中,如果两个待合并的候选项集有共同的 userID 属性,那么就可以将这两个集合合并成一个更大的集合,否则,这两个集合就不会生成候选项集了。事实上,如果两个候选项集没有共同的 userID 属性,那么它的子集一定不会是频繁项目集,也一定会在算法的剪枝过程中被去掉。所以这样的两个集合是没有必要

生成候选项集的。与此同时,候选集缩小之后,也间接地减少了扫描整个数据库的次数。

对于事务数据库的缩减,可以基于下面两个很明显的性质来缩减事务数据库:①在对候选序列 F_{k+1} 计数时,可以提前删除长度小于 k+2 的事务,因为长度小于 k+2 的事务肯定不会出现在序列集 F_{k+2} 中;②对于任意的事务 t∈T(事务数据库),如果 i∈I 并且 i∈t,f∈F_k且 i∉f,则 i 可以删除,这就是说,在一次扫描事务数据库 T 之后,可以把事务中的非频繁项删除。

2.2 频繁序列

AprioriAll 算法是一个经典的序列模式挖掘算法,但是该算法在生成候选序列的过程中,要求序列中的项目集必须有顺序,但是并不要求连续出现。而在很多实际应用中,要处理的序列不但要求有顺序,而且一定是连续出现的。假设 F_{k-1} 是 k-1 阶频繁序列,C_k 表示 k 阶候选序列。那么,在生成候选序列的合并阶段中,将 F_{k-1} 和 F_{k-1} 合并生成候选序列,对于其中的两个序列 s_1 和 s_2,如果将 s_1 的第一个项目去掉后得到的余串与将 s_2 的最后一项去掉后得到的余串相同,则将 s_1 和 s_2 合并。

2.3 算法设计

频繁项目集生成过程如下所示:

输入:

U={U1,U2,…,Ui}　//用户集合

T={t1,t2,…,tk}　//会话表

S　//支持度

输出:

·频繁项目集(Sequential Patterns)

(1)将会话表 S 中的事务按照 UserID 排序,然后对于每个 session 按照时间顺序排序;

(2)生成 1-频繁项集 F_1,F_1 中的每个项都应该包含一个对应的 UseID,F_1={large 1-itemsets};

(3)循环 for k=2;F_{k-1} ! =null;k++;　//随后的各轮搜索

(4)C_k= candidate-gen(F_{k-1},S,U);　//生成 k-候选项集

(5)扫描事务数据库中的每个事务 t_i;

(6)C_i= subset(C_k,t_i)

(7)扫描 C_i 中的所有候选项集集 c

(8)c. count ++;//c 的计数加 1

(9)如果 t_i 的长度小于 k+1,提前删除 t_i;//删除长度小于 k+1 的项

(10)如果 t_i 不属于 C_i,则将 t_i 删除;　//删除非频繁项

(11)对任意的 t∈t_i

(12)F_k={c | c. count>S};//找出支持度大于最小支持度的候选集作为频繁集

(13)从 F 中寻找最大引用项集

(14)结束

·候选项集的生成过程:candidate-gen(F_{k-1},S,U),如下所示:

(1)初始化,Ck = null;

(2)扫描 Lk-1 中的每个项集 Li

(3)对 Lk-1 中的每个项集 Lj

(4)如果 Li 和 Lj 有相同的 U　//某个用户同时访问了 Li 和 Lj

(5)c= Li 合并 Lj　//合并两个项集生成更大的候选项集

(6)如果 c 的 k-1-子集中存在非频繁项集,则删除 c

(7)否则,将 c 加入到 Ck;

(8)返回 Ck,结束;

·候选序列的生成过程:candidate-gen(F_{k-1},S),如下所示:

(1)初始化,C_k= null;

(2)扫描 L_{k-1} 中的每个项集 L_i

(3)对 L_{k-1} 中的每个项集 L_j

(4)如果 L_i 和 L_j 有相同的 U　//某个用户同时访问了 L_i 和 L_j

(5)如果 L_i 的后 k-2 余串和 L_j 的前 k-2 个余串相同　//连续的两个序列

(6) c= L_i 合并 L_j　//合并两个项集生成更大的候选项集

(7)如果 c 的 k-1-子集中存在非频繁项集,则删除 c

(8)否则,将 c 加入到 C_k;

(9)返回 C_k,结束;

3 实验结果及分析

实验主要包括两个部分:数据预处理部分和序列模式部分。数据预处理实验主要是路径填充阶段的实验。

3.1 数据源及实验环境

实验测试的数据源来源 http://fimi. cs. helsinki. fi/data/网站下的 kosarak. dat,数据源的大小为30M,该数据是匈牙利的一个新闻门户网站的用户点击流数据,已经经过了匿名处理,并且该数据集中的会话平均长度较短,相当多会话的长度都只有 1 到 3 个项。采用的软件为:Windows 7 旗舰版,JDK1.6,Mysql 5.0。

3.2 实验结果及分析

3.2.1 数据预处理实验

路径填充实验:截取一个用户会话中的访问页面,

如表1所示。

表1 用户会话中的访问页面

时间	访问页面	引用页面
…	168	
…	205	168
…	64	205
…	71	64
…	10	64
…	42	64
…	42	42
…	50	42
…	19	50
…	49	64

实验结果如图3所示。

usid	sequence	combined	complete
1	168-205-64-71-10-42-42-50-19-49	168-205-64-71-10-42-50-19-49	168-205-64-71-64-10-64-42-50-19-50-42-64-49
2	18-134-48-72-4-388-388-40-19-498	18-134-48-72-4-388-40-19-498	18-134-48-72-48-4-388-40-19-40-388-48-498-60

图3 路径填充实验结果

实验数据对应usid为1的元组，在combined字段中，可以看到连续两个相同的访问页面42已经合并，而complete字段中，在连续两个访问页面71和10中补充了访问页面64，这是因为71和10的引用页面都是64，这说明用户访问了页面71，然后回退到页面64，再访问页面10的。

3.2.2 频繁模式实验

截取事务数据库中的5个事务，如图4所示。

tid	Sequences
1	<{1 5} {2} {3} {4}>
2	<{1} {3} {4} {3 5}>
3	<{1} {2} {3} {4}>
4	<{1} {3} {5}>
5	<{1 5} {2} {3} {4}>

图4 事务数据库中的5个事务

设定支持度为2生成的候选序列集合如下所示：

·1-序列

$C_1=\{<1>_{1,2,3,4}<2>_{1,3}<3>_{1,2,3,4}<4>_{1,2,3,5}<5>_{1,2,4,5}\}$

$S_1=\{<1>_{1,2,3,4}<2>_{1,3}<3>_{1,2,3,4}<4>_{1,2,3,5}<5>_{1,2,4,5}\}$

支持度分别为4,2,4,4,4

·2-序列

$C_2=\{<1\ 2>_{1,3}<1\ 3>_{1,2,3,4}<1\ 4>_{1,2,3}<1\ 5>_{1,2,4}<2\ 1>_{1,3}<2\ 3>_{1,3}<2\ 4>_{1,3}<2\ 5>_{1}<3\ 1>_{1,2,3,4}<3\ 2>_{1,3}<3\ 4>_{1,2,3}<3\ 5>_{1,2,4}<4\ 1>_{1,2,3}<4\ 2>_{1,3}<4\ 3>_{1,2,3}<4\ 5>_{1,2,5}<5\ 1>_{1,2,4}<5\ 2>_{1}<5\ 3>_{1,2,4}<5\ 4>_{1,2,5}\}$

$S_2=\{<1\ 2>_{1,3}<1\ 3>_{1,2,3,4}<1\ 4>_{1,2,3}<1\ 5>_{1,2,4}<2\ 3>_{1,3}<2\ 4>_{1,3}<3\ 4>_{1,2,3}<3\ 5>_{1,2,4}<4\ 5>_{1,2,5}\}$ 支持度分别为2,4,3,2,2,2,3,2,2

·3-序列

$C_3=\{<1\ 2\ 3>_{1,3}<1\ 2\ 4>_{1,3}<1\ 3\ 4>_{1,2,3}<1\ 3\ 5>_{1,2,4}<1\ 4\ 5>_{1,2}<2\ 3\ 4>_{1,3}<2\ 3\ 5>_{1}<2\ 4\ 5>_{1}<3\ 4\ 5>_{1,2}\}$

$S_3=\{<1\ 2\ 3>_{1,3}<1\ 2\ 4>_{1,3}<1\ 3\ 4>_{1,2,3}<1\ 3\ 5>_{1,2,4}<2\ 3\ 4>_{1,3}\}$

·4-序列

$C_4=\{<1\ 2\ 3\ 4>_{1,3}\}$

$S_4=\{<1\ 2\ 3\ 4>_{1,3}\}$

·5-候选序列

分析由S_2生成C_3的过程，对于$<1\ 2>_{1,3}$和$<1\ 5>_{1,2,4}$这两候选项，由于它们具有相同的usid属性1，所以在Apriori All算法中，这两个候选项肯定会合并为一项$<1\ 2\ 5>_1$，然而由于文中改进的算法中考虑到了序列之间的连续性，所以这两个候选项在C_3中并没有合并成候选序列，而在S_3生成C_4的过程中，$<1\ 3\ 4>_{1,2,3}$和$<1\ 3\ 5>_{1,2,4}$以及$<1\ 2\ 3>_{1,3}$和$<1\ 3\ 5>_{1,2,4}$也都没有生成候选序列，因为它们也不是两个连续的序列。所以可以证明该算法确实有效地缩减了候选序列的数目。截取会话表中的4个会话，支持度设为30%，并且支持度的衡量是依据用户数，而不是依据会话数。如图5所示。

usid是用户会话的id属性，而uid是用户的id，不同的uid代表不同的用户。该会话生成的候选项集如下所示：

·1-项集 候选项集数目5

$C_1=\{<A>_{1,2,3}<B>_1<C>_{1,2,3}<D>_3<E>_{1,3}\}$

$S_1=\{<A>_{1,2,3}<B>_1<C>_{1,2,3}<D>_3<E>_{1,3}\}$

·2-候选项集 候选项集数目9

$C_2=\{<A,B>_1<A,C>_{1,2,3}<A,D>_3<A,E>_{1,3}<B,C>_1<B,E>_1<C,D>_3<C,E>_{1,3}<D,E>_3\}$

$S_2=\{<A,B>_1<A,C>_{1,2,3}<A,D>_3<A,E>_{1,3}<B,C>_1<B,E>_1<C,D>_3<C,E>_{1,3}<D,E>_3\}$

usid	uid	itmes
1	1	<A,B,C>
2	1	<B,C,E>
3	2	<A,C>
4	3	<A,C,D,C,E>

图5 会话表

·3-项集 候选项集数目7

$C_3=\{<A,B,C>_1\ <A,B,E>_1\ <A,C,D>_3\ <A,C,E>_{1,3}<A,D,E>_3$
$<B,C,E>_1<D,C,E>_3\}$

$S_3=\{<A,B,C>_1\ <A,B,E>_1\ <A,C,D>_3\ <A,C,E>_{1,3}<A,D,E>_3$
$<B,C,E>_1<D,C,E>_3\}$

·4-项集 候选项集数目2

$C_4=\{<A,B,C,E>_1<A,C,D,E>_3\}$

$S_4=\{<A,B,C,E>_1<A,C,D,E>_3\}$

·5-候选项集 候选项集数目0

观察2频繁项生成3候选项的过程,如$<A,B>_1$和$<A,D>_3$这两项,由于它们不具有相同的用户id属性,所以在C_3中并没有生成候选项,S_3生成C_4的过程中,对于项$<A,B,E>_1$和$<A,D,E>_3$,C_4中并没有生成<A,B,D,E>,也是因为它们不具有相同的用户id属性的缘故。

4 结束语

介绍了Web日志挖掘的各个阶段,设计并实现了一个Web日志分析系统。介绍了一种改进的序列模式挖掘算法和频繁项集算法,根据日志数据的特性,将用户属性引入频繁项目集的生成过程,有效地减少了候选项集的数目,并根据候选集的特点,逐轮压缩数据库。将连续序列引入到ApiroriAll算法的候选集合并过程中,实现了一种改进的ApiroriAll算法,在该算法中,待合并的两个候选序列必须是连续序列,通过实验证明了改进算法的有效性。

参考文献:

[1] Han Jiawei, Kamber M. 数据挖掘概念与技术[M]. 范　明, 孟晓峰,译. 北京:机械工业出版社,2005.

[2] 郭崇慧,田凤占. 数据挖掘教程[M]. 北京:清华大学出版社,2006.

[3] Han Jiawei, Meng Xiaofeng, Ang Jing. Research on Web Mining[J]. Journal of Computer Research & Development, 2001, 38(4):405-414.

[4] Liu Bing. Web数据挖掘[M]. 俞　勇,薛贵荣,韩定一,译. 北京:清华大学出版社,2009.

[5] Liu Haibin, Kes V. Combined mining of web server logs and web contents for classifying user navigation patterns and predicting users' future requests[J]. Data and Knowledge Engineering, 2006(7):307-309.

[6] Spiliopoulou M, Mobasher B, Berendt B, et al. A framework for the evaluation of session reconstruction heuristics in Web usage analysis[J]. Informs Journal of Computing, Special Issue on Mining Web Based Data for E- Business Applications, 2003, 15(2):171-190.

[7] Mikroyannidis A, Theodoulidis B. A theoretical framework and an implementation architecture for self adaptive Web sites [C]//Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence(WI'04). Beijing: IEEE Press, 2004.

[8] 邵峰晶,于忠清. 数据挖掘原理与算法[M]. 北京:中国水利水电出版社,2003.

[9] 毛国君,段立娟,王　实,等. 数据挖掘原理与算法[M]. 北京:清华大学出版社,2005.

[10] 董志峰,陈俊杰,付裕峰. Web日志会话的个性化识别方法的研究[J]. 计算机工程与应用,2008,44(8):179-182.

[11] 马瑞民,李向云. web日志挖掘中数据预处理技术的研究[J]. 计算机工程与设计,2007,28(10):2358-2359.

[12] 李烈彪,张海鹏,周亚峰. Web日志挖掘中数据预处理方法的研究[J]. 计算机技术与发展,2007,17(7):45-48.

(上接第210页)

参考文献:

[1] 王宏伟, 赵国庆. 在GSM网络中实现高压电路线路检测[J]. 仪器仪表学报,2007,28(12):2260-2262.

[2] 徐妙君,张晓霞. 短消息控件的设计与实现[J]. 计算机技术与发展,2007,17(8):64-66.

[3] 管　仲,王　伟,蔡德林. 基于GSM短消息的远程信息显示系统设计[J]. 计算机技术与发展,2009,19(12):158-161.

[4] 宏晶科技. STC12C5628AD系列单片机器件手册[EB/OL]. 2009-10. http://www.STCMCU.com.

[5] 基于nRF24L01的近距离无线数据传输[J]. 应用科技,2008,35(3):55-57.

[6] Zhang Guiming. Research and design of remote control and alarm system based on GSM/SMS[J]. Journal of Sichuan Normal University:Natural Science,2004,27:51-53.

[7] 西门子公司. TC35/TC37硬件接口描述[M]. [s.l.]:[s.n.],2002.

[8] Siemens Mobile, TC35i AT Command Set, Version 00.01, DocID TC35i_ATC_V00.01(2003)[M]. [s.l.]:[s.n.], 2003.

[9] 黄镇谨,李春贵,欧阳浩. 基于GPS和GIS的公共交通监控平台[J]. 微机发展(现更名:计算机技术与发展),2004,14(zl):123-126.

[10] MapInfo Corporation. MapX4.5 Developer's Guide[EB/OL]. 2002. http://www.mapinfo,com.

[11] Fotheringham S, Rogerson P. Spatial Analysis and GIS[M]. [s.l.]:Taylor & Francis,1994.

[12] 林　勇,应新洋. DELPHI下用MSCOMM32实现串行通信[J]. 计算机应用研究,2003,20(8):158-160.

[13] 林　涛,何建农. 基于组件GIS技术的电信基站管理系统[J]. 福州大学学报,2004,32(zl):123-126.