Web 中的行情数据获取与预测研究

于春燕[1,2],胡学钢[1]
YU Chun-yan[1,2],HU Xue-gang[1]

1.合肥工业大学 计算机与信息学院,合肥 230009
2.滁州学院 计算机科学与技术系,安徽 滁州 239000
1.School of Computer Science and Information Engineering,Hefei University of Technology,Hefei 230009,China
2.Department of Computer Science and Technology,Chuzhou University,Chuzhou,Anhui 239000,China
E-mail:yuchy@chzu.edu.cn

YU Chun-yan,HU Xue-gang.Research on market data extraction and forecast on Web.Computer Engineering and Applications,2009,45(20):202-204.

Abstract: It is significant to extract market data in Web pages for prediction and analysis.An extraction algorithm for Web pages is proposed.Taking into account the common practice that "market data are usually displayed in the largest table on a Web page",the market data extraction algorithm first detects the largest table on a Web page and then transfers it into a DOM tree,and in the end gets the node values of the tree.This algorithm is different from traditional ones in that it can automatically detect market data and does not need a data extraction region to be specified by the users.A prototype system for agriculture product price prediction is designed and developed.The system extracts market price data from a given website automatically and predicts the price in the future months.Experimental results show the prediction results are satisfying.
Key words: Web content mining;market data extraction;market data prediction

摘 要:抽取网页中的行情数据进行预测和分析具有重要意义。提出了 Web 中的行情数据抽取算法,该算法主要基于"行情数据通常在网页中表现为区域最大的数据表格"等实践规律,首先自动识别出最大的数据表格,然后转换为 DOM 树结构,最后抽取 DOM 树的结点值。与传统算法不同,算法自动抽取行情区域而无需用户定义抽取数据区域。设计了一个农产品价格预测原型系统,该系统针对某个农产品,自动从特定网站获取价格数据,对月度价格进行预测,实验表明预测性能较好。
关键词: Web 内容挖掘;行情数据抽取;行情预测
DOI:10.3778/j.issn.1002-8331.2009.20.059 **文章编号:**1002-8331(2009)20-0202-03 **文献标识码:**A **中图分类号:**TP18

1 引言

万维网(World Wide Web)已成为最大的信息载体,其中蕴含了各种行情信息,例如,农产品销售网站发布了农产品每日交易的价格;天气预报网站发布天气变化的数据;导购网站提供商品的报价等等。针对某个具体领域的行情数据抽取,具有明确的实际应用目的。例如,Yukitaka[1]等从拍卖网中自动抽取出售者对物品的特征描述,从表格、项目、语句中学习和抽取特征,提供给想要购买的人。Doorendos[2]等通过识别在线目录和电子商务网站,抽取价格列表和订单信息,从不同站点汇总信息并发现有趣的交易。行情数据多表现为表格形式,目前常用的抽取方法及分析如下:

(1)分析 HTML 标记构造包装器抽取数据。包装器(wrapper)负责将隐含在 HTML 文档中的信息提取出来,并且转换成能被进一步处理的、以某种数据结构存储的数据[3]。包装器方式需要手工分析待抽取页的 HTML 代码,并定义抽取的标记区域后进行抽取。该方式依赖于 Web 页面的布局、格式等,一旦页面格式改变,往往要重新编写包装器。近年来的一些研究通过机器学习方法自动识别表格[4]和视觉线索识别并抽取表格[5],改善了抽取性能。

(2)利用 DOM(Document Object Model)构造包装器抽取数据。此方式将 HTML 页规范化后转换为 DOM 树结构,用 XPATH 表示抽取的数据区域。其抽取步骤为[6-9]:①根据用户指定的 URL 获取样本网页数据。②利用 HTML TIDY[10]网页转换为 XML 文档(实际为 XHTML 文档)后解析为 DOM 树结构。③定义抽取规则。利用 XPATH 技术,在 DOM 树上定位数据块。④抽取。可以用 DOM、SAX(Simple APIs for XML)、正则表达式三种方式实现。⑤将抽取出来的数据存储在数据库或 XML 文档中。该方式仍未改变抽取器对页面格式的依赖,抽取过程需要人工定义数据块。一个普通的 HTML 网页包含了千百个标记,完整地转换整个文档为 DOM 树结构、空间性能和时间性能差。

(3)利用自然语言处理方法进行抽取。该方式不仅仅分析

基金项目:安徽省科研项目(the Scientific Research Project of Anhui Province of China under Grant No.KJ2008B033)。
作者简介:于春燕(1979-),讲师,主要研究领域为 Web 技术应用;胡学钢(1961-),博士,教授,主要研究领域为人工智能与数据挖掘。
收稿日期:2008-04-21 **修回日期:**2008-07-15

HTML 标记,还利用关键字和语义线索,采用自然语言处理方法进行抽取。自然语言处理方法要求大量例子的训练,且处理速度比较慢,在表格形式的行情数据抽取中应用尚不多见。

(4)基于本体的抽取方法。利用本体对抽取页面的类型进行描述,并根据领域特点设计出数据框架,用于给规则匹配。基于本体的方法与待抽取 Web 页面格式无关,甚至在领域改变时也只要改变应用本体即可,其应用效果较好。然而,本体构造中的数据框架设计非常困难。

2 网页中行情数据抽取的思路与算法描述

2.1 基本思想

网页设计的实践经验表明,行情数据通常是表格形式的动态区域,对应于<table></table>表格标记。因此,抽取行情数据的任务就转化为 Web 页中表格数据的抽取。但在网页中,表格既作为数据列表,也大量用于网页内容布局。

HTML 语言和 Web 页制作中存在一些规律:(1)一个 Web 页可能有几十个表格,其中大部分是布局表格,而数据表格有时也不止一个。(2)Web 页的行情数据大多位于表格标记<table></table>中,表现为一个宽度最大、高度大于一定阈值且无内嵌表格的表格 MaxDataTable,MaxDataTable 视觉上占据了最大的区域,在对应源代码中所占字符数也最多。(3)宽度小于一定阈值或高度小于一定阈值的表格不是数据表格,其形状为一个横长条或竖长条,一般为修饰用的布局表格。(4)尽管 HTML 代码中很多标记的格式不良好,但表格元素的源代码是格式良好的,否则浏览器解析显示时将出现格式混乱的表格。

2.2 算法描述

基于以上分析,提出了自动识别网页最大区域的行情数据抽取算法 MDT-E(MaxDataTable-Extraction)。算法首先识别 Web 页中字符数最多、无嵌套且高度大于阈值的表格,然后仅将识别出的 MaxDataTable 转换为 DOM 结构,再抽取 DOM 树中<td>节点的值。根据实践经验,阈值设为 300px。具体算法步骤如下:

(1)获取网页数据。获取指定 URL 的网页内容,并读取纯文本形式的源代码。

(2)规整化网页。先去除网页中的<head></head>部分,留下 HTML 源代码的<body>部分,再删除<Script></Script>脚本、<input>表单元素、<img>图片、<object>内嵌对象、<a>链接对象等。删除的目的不仅可以去无用数据,还减少了因这些数据包含在<table>中增大了表格长度,从而造成错误结果的可能。

(3)从网页的所有表格中识别出宽度最大、高度大于一定阈值且无内嵌表格、字数最多的表格,即 MaxDataTable。

很多<table>标记中都有属性标记(形如<table class="m">),因此无法按照"<table>"分割。将 Web 页的源代码按"<table"分割后存储到数组 Alltables 中。对每一个数组元素 AllTables(i),判断其中有无"</table>标记,若有,则对 Alltables(i)中第一个"</table>"前的内容,判断其 *width* 值是否满足大于阈值的条件,若满足则将该 Alltables(i)存入数组 SingleTables。对每一个 AllTables(i)分析完毕后,得到了存放所有未嵌套表格的数组 SingleTables,在 SingleTables 中找出长度最大的数组元素,并在其开头加上"<table"字符,最后加上"</table>"字符,即得到 MaxDataTable。

(4)用 Tidy 程序规范化最大表格后,转换为 DOM 树。

(5)抽取单元格内数据。获取该表格内<td></td>间的数据并存入数据表。

```
算法:MDT-E
输入:URL
输出:MaxDataTable
FindMaxDataTable(URL)
html=Readtotext(URL)//读入指定 URL 的 Web 页源代码
DeleteTags("script","")
AllTables=split(html,"<table")
    FOR i=0 to AllTables.upperbound
        IF AllTables(i).indexof("</table>")=TRUE THEN
            temptable=split(html,"</table>")
            IF width>=300 THEN
              SingleTables=temptable
              END IF
      END IF
    END FOR
    FOR i=0 to SingleTables.upperbound
        IF maxtable<SingleTables(i).length THEN MaxDataTable=
SingleTables(i)
    END FOR
    Table=TidyHTML(MaxDataTable)
    obj=GenerateDOMTREE
      DO While obj.Read()
        IF Node="<TR>" THEN
            NEXT NODE
        ELSEIF Node="<TD>" THEN
        insert Node.value into table
       END IF
      END DO
```

2.3 算法分析

本文算法与目前基于 DOM 的抽取算法,有以下不同之处:(1)对于错误的表格元素,浏览器解析显示时将表现为混乱的表格,因此,表格元素的标记一般是标准的。有鉴于此,算法在识别表格前没有调用 Tidy 程序规范网页,而是在识别出最大表格后再调用 Tidy 程序规范化,减少了规范化的网页内容。(2)自动识别出最大的数据表格,而不是由用户指定数据区域或抽取规则,可实现自动数据抽取。(3)仅将识别出的数据表格转换为 DOM 树结构,而不是将整个文件转换为 DOM 树,减少了空间损耗。(4)本文算法针对行情数据的抽取,不适于首页或导航页,因为首页或导航页中的页面布局划分并不分明,通常没有明显的大块数据区域。(5)本文算法不能连续自动地抽取多页行情数据(通常应用了动态网页数据库技术),需要人工指定翻页按钮,才能实现翻页后的数据抽取。

2.4 行情数据抽取实验

人工挑选的 20 个农产品批发市场的价格行情页面、20 个天气预报网页和 20 个销售信息网页。识别其中的最大表格,并提取该表格中的数据项。实验在 Pentium 2.8 G/512 M 机器上运行,实验分两阶段,先识别 MaxDataTable 再抽取数据。部分实验结果见表 1。

表 1 行情数据抽取实验

	农产品网页	天气预报网页	销售信息网页
行情数据识别率/(%)	85	90	80
行情数据抽取率/(%)	85	80	75

表 2 某批发市场 2001~2006 年间莲藕的月度均价

X_1	1.863 333	X_{13}	1.301 667	X_{25}	1.900 000	X_{37}	3.328 000	X_{49}	3.257 143	X_{61}	3.438 462
X_2	1.042 308	X_{14}	1.040 000	X_{26}	2.230 000	X_{38}	2.733 333	X_{50}	2.596 552	X_{62}	2.406 452
X_3	1.153 846	X_{15}	1.316 667	X_{27}	1.906 250	X_{39}	2.790 000	X_{51}	2.472 414	X_{63}	2.946 667
X_4	1.229 412	X_{16}	1.325 000	X_{28}	2.000 000	X_{40}	2.796 552	X_{52}	2.519 355	X_{64}	2.996 774
X_5	1.734 211	X_{17}	1.556 250	X_{29}	1.900 000	X_{41}	5.137 931	X_{53}	3.896 552	X_{65}	4.575 862
X_6	1.373 529	X_{18}	1.290 000	X_{30}	2.000 000	X_{42}	5.250 000	X_{54}	4.016 129	X_{66}	4.427 586
X_7	1.394 444	X_{19}	1.266 667	X_{31}	2.700 000	X_{43}	3.596 774	X_{55}	3.532 258	X_{67}	3.587 097
X_8	1.260 714	X_{20}	1.600 000	X_{32}	3.000 000	X_{44}	2.303 571	X_{56}	2.460 000	X_{68}	2.510 000
X_9	1.558 333	X_{21}	2.875 000	X_{33}	3.230 769	X_{45}	2.923 333	X_{57}	3.031 034	X_{69}	3.160 000
X_{10}	2.465 385	X_{22}	2.409 524	X_{34}	2.090 000	X_{46}	2.093 548	X_{58}	2.220 690	X_{70}	3.160 000
X_{11}	1.556 250	X_{23}	1.986 364	X_{35}	2.148 276	X_{47}	2.093 333	X_{59}	2.414 286	X_{71}	3.160 000
X_{12}	1.009 091	X_{24}	1.317 857	X_{36}	2.350 000	X_{48}	2.564 516	X_{60}	3.132 258	X_{72}	3.160 000

实验表明,只要被提取内容为表格且占据了最大区域的页面,都能正确提取。不能识别的页面都是因为网页中出现了多个宽度和内容相似的区域。最大表格抽取节点值的准确性也令人满意,未正确抽取的情况有页面自身的设置等原因,如 http://www.hz18.com/Price.aspx 正确抽取了数据表格,但由于抽取出的文字为乱码,从而未能获得正确的节点值。实验结果也验证了本文算法不适于框架页、没有主要表格的页面。

3 农产品价格预测系统的设计与实现

该系统从某大型批发市场网站 http://www.xinfadi.com.cn/Price.asp 获取 2001 年至 2006 年间莲藕的每日价格数据,可以显示指定范围的价格信息,并预测未来数月的价格,系统界面见图 1 所示。获取数据采用 MDT-E 算法,由于每日价格数据以动态表格形式出现,编程模拟提交实现翻页再获取页面中的价格数据。每日报价数据不适于预测莲藕未来数月或更长时间的价格,因此,将每日价格数据按月分组求平均,得到 6 年来莲藕的每月均价数据,见表 2 所示(X_i 为每月均价)。采用时间序列预测模型[11],选择不同的历史数据,对比二次曲线预测模型和线性预测模型的预测结果,见表 3 所示。

表 3 不同样本数下线性与二次曲线季节模型的预测结果

月份	实际值	二次曲线预测值			线性预测值		
		样本数 60 个月	样本数 48 个月	样本数 36 个月	样本数 60 个月	样本数 48 个月	样本数 36 个月
1	3.438 462	3.289 731	3.388 306	3.310 616	3.347 667	3.422 735	3.495 245
2	2.406 452	2.767 531	2.870 976	2.771 897	2.830 434	2.926 277	2.714 788
3	2.946 667	2.859 440	2.849 170	2.849 551	2.939 532	2.931 333	2.899 463
4	2.996 774	2.897 738	2.900 471	2.872 356	2.994 677	3.013 297	2.944 083
5	4.575 862	4.068 707	4.214 963	4.010 426	4.227 639	4.423 444	4.808 896
6	4.427 586	3.978 447	4.229 458	3.898 268	4.156 827	4.485 520	4.840 362
7	3.587 097	3.427 972	3.409 976	3.337 997	3.602 040	3.656 031	3.872 411
8	2.510 000	3.043 557	2.984 645	2.944 318	3.216 706	3.236 327	3.132 267
9	3.160 000	3.937 077	3.880 349	3.782 617	4.185 792	4.257 005	3.668 155
10	3.160 000	3.483 143	2.816 668	3.322 483	3.725 669	3.127 646	2.531 157
11	3.160 000	2.930 700	2.634 286	2.774 531	3.154 194	2.961 903	2.590 464
12	3.160 000	2.708 154	2.647 220	2.543 712	2.933 116	3.015 114	3.069 717
MAPE 值		10.740 7	10.834 9	10.954 8	10.049 6	8.768 5	10.153 9

预测步骤:(1)选择样本数据;(2)平滑数据 $Y_t=(0.5*X_{t-6}+X_{t-5}+X_{t-4}+X_{t-3}+X_{t-2}+X_{t-1}+X_t+X_{t+1}+X_{t+2}+X_{t+3}+X_{t+4}+X_{t+5}+0.5*X_{t+6})/12$;(3)分别选择二次曲线预测模型和线性预测模型 $Y_t=C(1)*T^2+C(2)*T+C(3)$,$Y_t=b*T+C(4)$ 分别求得 $Y_t=0.056\,662\,552\,66*T^2-0.000\,332\,186\,186\,7*T+0.954\,805\,266\,2$,$Y_t=0.032\,412\,961\,03*T+1.253\,883\,563$;(5)加入季节调整的预测 $F_t=Y_t*Factor$。

实验表明,总体上样本数与预测性能成正比。平均绝对误差率 MAPE 值[12]均在 10~11 之间,表明线性季节模型和二次曲线季节模型的预测性能都较好。线性季节模型的预测性能略好于二次曲线模型。采用线性季节模型用近 60 个月的数据预测 2006 年的价格数据,如图 2 所示。

图 1 原型系统主界面

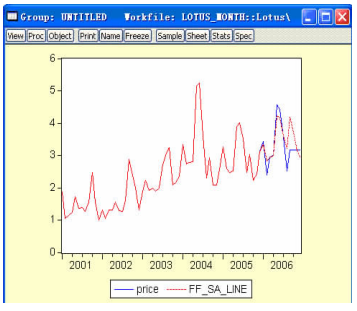

图 2 线性季节模型预测(虚线为预测曲线)

4 结束语

提出的 MDT-E 算法,充分利用网页结构特征和设计规律,无需用户干预即可自动识别最大数据区域并存储。最后设计并实现了一个农产品价格预测系统,从特定网站中获取价格数据并预测。实验表明行情数据识别与抽取的性能较好,农产品价格预测系统也具有一定的实际意义。

参考文献:

[1] Kusumura Y,Hijikata Y,Nishida S.Extracting fixed information from miscellaneous documents on net auction[C]//17th International Conference on Advanced Information Networking and Applications,AINA 2003:446-453.

(下转 248 页)