**基于概率模型的包装器健壮性优化方法**

李贵陈韶刚 李征宇 韩子扬 孙平 孙焕良

沈阳建筑大学信息控制工程学院

**摘 要：**本文研究了构建网页信息抽取包装器的健壮性优化问题。网页结构经常变化导致包装器失效，为了提高包装器的健壮性，本文在一个概率模型的基础上，通过自底向上地使用动态规划方法计算，来构建最优概率包装器候选集合。通过在实际网站上的实验，证明了算法可以有效应对网站的变化，优化了包装器的健壮性。

**关键字**：包装器；健壮性优化；网页信息抽取；

**中图法分类号**TP301.6　　 **文献标识码**A

**Optimal Approach for Robust Wrapper**

**Based on a Probabilistic Model**

Li Gui Chen Zhao-xin Li Zheng-yu Han Zhi-yang Sun Ping Sun Huan-liang

*（Faculty of Information & Control Engineering ， Shenyang Jianzhu University， Shenyang 110168， China）*

**Abstract** In this paper, we consider the problem of constructing wrappers for web information extraction that are robust to changes in websites. Webpages frequently change, and even very slight changes cause the wrapper to break and require them to be re-learned. Based on a probabilistic model,we using dynamic programming by computing bottom-up to construct optimal robust wrappers and improve the robustness of the wrappers. By evaluating on real websites, we demonstrate that in practice, our algorithms are highly effective in coping up with changes in websites, and reduce the wrapper breakage.

**Keywords** wrapper; robust optimal; web information extration;

# 1 引言

随着Web上数据的不断增加，访问Web数据已成为获取信息的重要手段。有些网站使用脚本从系统后台数据库获得数据并生成HTML，如购物网站、分类信息网站、学术资料网站等基于表单的网站。脚本生成的网页的结构相似，因此信息提取系统可以使用简单的规则，从网页中抽取信息。由于包装器可以频繁地用相对较少的样本来生成，所以对于脚本生成的网页来说，使用包装器是最佳抽取策略。

然而，由于包装器抽取数据时严重依赖网页的结构，经常会出现包装器破损的问题：由于Web数据动态性的特点，网页内容经常变化，甚至是一个细小的变化都可能引起包装器失效，即导致现有抽取规则失效，需要重新学习抽取规则。

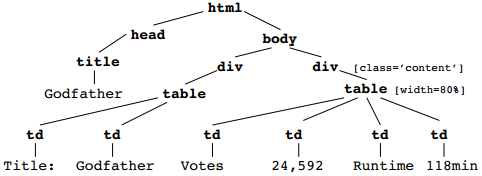


图1 一个HTML网页

举例说明，图1代表了一个脚本集成网页的XML文档树。抽取Votes的数量，我们可以用一个如下的XPath：

。

几处小变化可以使包装器失效，例如，第一个div被删除，一个新tr加入第二个div，Votes和Runtime的顺序改变等等。

关于包装器破损已经有大量研究[4,5]。为了适应网站的变化，包装器健壮性研究显得尤为重要。包装器按照其构造方式可以分为人工构造、半自动构造、全自动构造等方式。

现有的半自动抽取包装器需要用户手工标注样本训练数据，然后对训练数据进行学习并生成抽取包装器。很多的研究工作是基于Html结构的抽取，通过解析器将Web网页解析成DOM树，根据Html页面内在的结构特征，构造相应的抽取规则进行抽取。利用页面本身的层次结构，半自动或自动形式生成抽取规则。相关研究有XWRAP[6]，RoadRunner[7]，MDR[8]等。

直接构建更健壮的包装器问题的研究相对比较少，这些研究通过在网页变化后的版本上实验性的评估了包装器的健壮性。Myllymaki等人提出了一个比较健壮的包装器，该方法明显的降低了包装器的破损率。他们手工构建包装器，但是没有解决自动学习包装器抽取规则的问题。

第一个正式的关于包装器健壮性的框架，是Nilesh Dalvi[2]等提出来的。他们定义了一个包装随时间进化的模型，计算网页的转换概率，构建包装器用于数据抽取。这个模型可以评估包装器的健壮性。通过评估健壮性，可以从一个XPath集合中选取包装器，但是构建最健壮的包装器的问题仍没有解决。为了提高包装器的健壮性，本文提出了一种基于概率模型的包装器健壮性优化方法。

# 2 问题定义

## 2.1问题概述

本文中我们不使用简单的XPath规则，而是从一组抽取相同信息的互补XPaths集合中，采用多数表决制选取。只有在过半的XPaths失效时包装器才会破损。因此，健壮性的规则越多，我们得到的包装器就越健壮。我们的问题是：对于一个网页集合，如何构建最健壮的包装器？

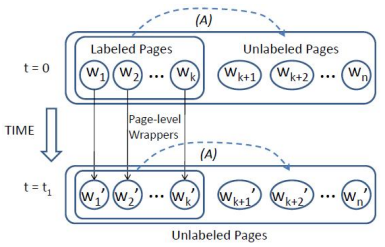


图1 包装器学习系统

图1展示了一个包装器学习系统如何工作。有两个抽取任务涉及到包装器：(A)通过归纳已标记的k个实例，从剩余的中抽取。(B)通过提高应对变化的健壮性，从新版网站中抽取。

在(A)中，网站包含了很多结构相似的网页；因此需要速度很快的，并可以经受网页结构细小变化的包装器。现有的一些标准的包装器算法[3, 11, 7, 16, 6, 1]可以解决(A)问题。

但是一旦网页结构随着时间有较大的变化，(B)需要更健壮的包装器，可以经受更大的网页结构变化。为了解决（B）的问题，我们进行以下两步操作：第一步，对于上一代中已标记的网页，通过比较与来确定标记在中的位置。第二步，我们有了中标记的位置，就可以使用以前的包装器算法来从中抽取。其中，只有第一步，解决标记转移（transfer）时需要高健壮性。在这一步中可以通过牺牲性能来得到高健壮性，只有一小部分标记的网页的标记会改变。

为了处理(B)的健壮性的问题，我们需要处理的是用旧版的网页来标记新版的网页。把一个网页看成一个整体，本文提出一个网页级别的健壮的包装器。一个有标记节点的网页，我们想从未来的某个版本中抽取信息。Nilesh Dalvi[2]等提出了一个包装器随时间变化的模型，同时可以基于文档数据的学习归纳包装器。在此模型的基础上，我们用算法来构建一个包装器，可以给出下一版本中标记最可能出现的位置。

## 2.2相关定义

**有序标记树**：令w代表一个网页。我们把w表示成一个有序的标记树，对应网页解析的HTML DOM树。

两个网页和，如果二者有同样的结构和标记，例如和的节点之间一一映射，就定义二者全等，。两个节点，，如果有，并且和也一一映射，那么和也是全等的。

**D-node**：每一个网页w都有一个特殊节点，包含了感兴趣的文本信息，例如，一个包含租房网页租金价格的节点。本文假设每个页面只有一个D-node。

**网页进化**：我们假设有一些进化过程通过对网页一系列的编辑操作，生成了一个新一代的网页。本质上就是一个编辑脚本。因为我们主要关心结构变化，忽略了那些不是针对有序标记树的变化。不过没有给出的定义，我们只是提供了新一代的网页。

**包装器**：是一个函数从一个网页到一个网页中节点的映射。表示一个含有D-node节点的网页，表示编辑脚本s作用在上之后得到的新网页。我们在中寻找的位置。如果，就表示包装器在新网页中有效，否则就称包装器失效或破损。

**健壮性**：是测量包装器在新网页中继续保持有效的能力。

**抽取可信度**：是一个包装器对一个特定新版本能起多大作用。如果和差别很大，抽取的可信度就会很低。不过，如果中所有改变都在远离D-node节点的部分，可信度就会很高。

# 3 基于树编辑模型的抽取方法

健壮抽取框架，如图2所示。主要包括概率模型和候选包装器生成两个模块，再通过包装器健壮性评估，选出健壮的包装器。



图2 健壮的抽取框架

**Archival Data（文档数据）**，一个网页，随着时间发生了很多改变。令,…代表的不同版本。文档数据是一个对的集合。可以通过监测网页随时间的变化来获得文档数据。

**Probabilistic Model（概率模型）** 概率模型是我们框架的重要组成部分，给定一个当前的网页快照，概率模型可以给出下一个状态的概率分布。例如，它给出一个概率，我们简写成。

**Model Learner（模型学习模块）** 模型学习模块以文档数据为输入，来学习生成一个最合适的模型。模型学习组件学习参数值，使得观察文档数据的概率最大化



**Training Data（训练数据）** 一个抽取任务是抽取感兴趣的领域，例如从anjuke网站上的租房页面抽取租金价格。

**Candidate Generator（候选包装器生成器）** 候选包装器生成器使用有标记的样本数据来生成一系列的候选包装器。关于使用标记样本数据来生成包装器的问题，已经有大量研究，其中有些是针对学习XPath规则的[3]。上面提到的技术都可以用于候选集合的生成。在第4节我们介绍了一种适合做健壮候选集的XPath包装器。

**Robustness Evaluator（健壮性评估）** 健壮性评估使用候选包装器集合，使用文档数据学习得到概率模型，使用该模型评估每一个候选包装器的健壮性，从而选择最健壮的包装器。我们定义一个包装器的健壮性为一个包装器在未来的网页中正常工作的概率。令W为一个包装器，X为一个网页，P是模型学习模块给出的概率模型。令X|=W表示W在X上有效。那么一个包装器在X上的健壮性定义如下：



# 4 概率模型健壮性优化

## 4.1概率模型

上一节中提到的概率模型是最近提出的一种获取网页随时间的变化的方法。在这个模型中，概率被定义为一个概率转换器，输入项是一个网页，输出项是网页的新版本，通过一个从概率空间提取的编辑操作的随机集合。这些编辑操作与第二节中描述的一样。

令为所有标记的集合。通过一个概率集合参数化转换器：每个标记，删除的概率为，每个标记插入的概率为，每对标记替换的概率为。此外，还有一个概率（表示转换器停止网页上编辑的概率）。在网页未来所有可能的状态上，最后的转换器定义了一个概率分布。表示为。

给定两个树、，大小分别为、，深度分别为、，Nilesh Dalvi[2]等人提出了一个在时间内计算概率的算法。他们也提出了一个有效的算法，来从文档数据中得到概率转换器的参数。

## 4.2概率健壮性

在概率模型中，中的编辑操作来自概率分布。根据一个网页未来所有可能的状态，给出了一个概率分布。概率健壮性衡量包装器在期望条件下的表现。

给定一个概率模型，我们定义在网页上，一个包装器的概率健壮性为，其中是从概率分布中随机抽取的。用来表示优化包装器。给定一个概率转换器和一个网页，我们的目的就是构建使得概率最大化的包装器。

给定两个网页和，及相应节点和，定义为转换器将转换成一个与同构的树的概率，同时节点映射到与同构的节点。

定义为包装器，对于给定网页的新版本，输出中节点使得有最大值：

 （3）

对于一个固定的，可以把计算的问题缩减为计算的问题。可以简单地把和的标记，替换成标记和，并为标记的删除和除外所有的替换的概率强制变成0。这样我们就能在时间内计算。令在中所有节点之间变动，重复这个过程可以直接得到一个最优概率包装器的算法，复杂度为。

## 4.3健壮性优化

在4.2提到的算法中，对的调用中有许多重复计算，所以可以消除冗余使算法更高效，复杂度更小。

给定树中一个节点，我们定义它的前缀是树中所有在它左边的节点，先序遍历这个树，在给定节点被访问之前已经被访问的节点组成了它的前缀。例如图3左边树中，是节点的前缀。每一个树中节点的前缀都可以看成是这个树的一个前缀。

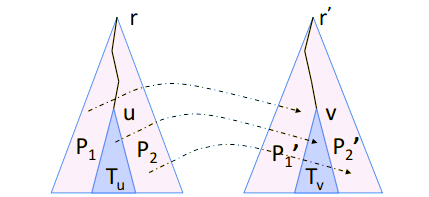


图3 概率包装器子树之间的映射关系图

令为通过一系列编辑操作得到的一个树。考虑和中节点之间的标准映射，该记录中节点如何与中节点对应。观察映射关系得出，如果节点a映射到b，a的孩子只能映射到b的孩子。同理，a左边的节点，只能映射到b左边的节点。例如，图3中，如果节点映射到节点，中的节点必须映射到的节点，中的节点必须映射到中的节点，并且中的节点必须映射到的节点。

我们得出以下结论：对任意等于下面三项的乘积：：转变成的概率。：转变成的概率。：转变成的概率。

优化算法基本思想是，提前计算子树的前缀变化的所有可能性，并计算对于所有的。概率包装器优化算法如下。

|  |
| --- |
| **Data**:,,  **Result**:New location ofinand confidence of extraction |
| Run TP(,)and recordand for all ;  tree under;prefix ofin;;  Runand record transformation probabilities ofto complete subtrees in;  **foreach** **do**  tree under;prefix ofin;;  **foreach** *in PostOrder* **do**  Compute transformation probabilities of prefix of in tree  to tree ;  **end**  Transformation probility of  to tree  ;  **end**  Return node with largest ,and |

算法1 最优概率包装器

概率包装器的复杂度为，将以前的算法复杂度减小了。我们定义抽取可信度就是正确抽取的概率，是用概率编辑脚本从得到的。抽取正确的概率即概率。算法1中也计算了这个变量，所有无需其他开销。

# 5 实验

## 5.1实验准备

**实验数据**：为测试我们方法的健壮性，我们使用了房地产领域网站进行实验。使用来自安居客和淘房网的记录作为训练数据，定期爬取获得不同时间段的网页新版本，从而对爬取的网页进行抽取实验。

**实验对比**：我们用Java实现了最优概率包装器。为了对本文提出的方法（OptProXPath）进行评估，主要与传统的全路径XPath方法（FullXPath）、Nilesh Dalvi[X]等提出来的基于概率模型的方法（ProXPath）进行性能比较。

**实验细节**：包装器的输入是：(a)一个旧版本的网页，(b)旧网页中D-node节点的位置，(c)新版本的网页。对于每一次执行，检查包装器是否在新版本中找到D-node节点。

## 5.2实验结果及分析

网页从旧版本到新版本所用的时间，我们用跨度K来表示。如果新旧网页的版本号相差5，则它们的跨度是5。对于所有可能的跨度大小，我们研究了三种方法的相对表现，平均编辑距离，花费时间。另外，我们也测量了每个包装器抽取所用的时间。包装器OptProXPath在Intel奔腾双核2.6GHz的处理器上平均花费500ms。

实验中，变换跨度大小，通过抽取中出现错误次数来评估三种包装器。选取3个跨度值（1，5，10），统计出现的错误数，见图4。y轴代表包装器失效的时间百分比。例如对于K=1，FullXPath包装器和ProXPath包装器失效时间大约是15%和5%，OptProXPath失效时间不到3%。优化的包装器表现得更好些。

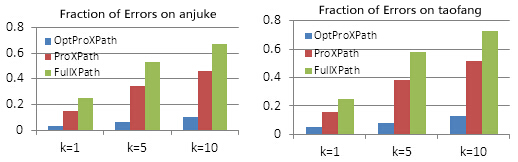


图4 在anjuke与taofang上的实验结果图

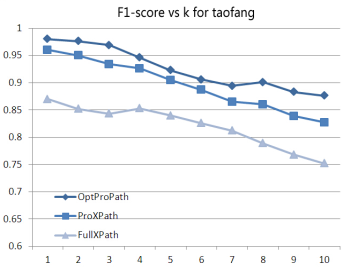
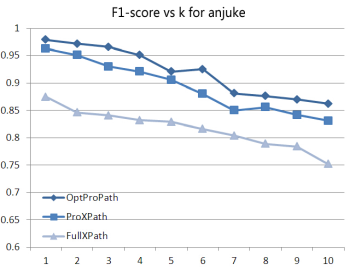


图5 F1-score随着K增长的对比图

实验中使用了准确率和召回率的综合指标F1-score（y轴），并绘制出三种方法随着跨度K（x轴）增长的对比图（图5）。从图中可见，随着跨度的增加，执行三种方法所得到的F1-score均收到影响，但我们的方法仍优于另两种方法。主要原因是FullXPath采用自顶向下的生成全路径的包装器表达式，仅考虑从页面抽取规则；ProXPath仅从概率情况考虑了网页变化，健壮性不高；我们的方法在概率模型的基础上构造最优健壮包装器，从而有效应对网站变化。

## 6 结论

包装器抽取是数据抽取与集成领域中的一个重要问题。本文提出了一个基于概率模型的包装器优化方法，展示了在概率模型下，构建优化包装器的算法。通过真实网站的评估，证明了该算法可以有效应对网站变化，并且与现有的技术相比，大大降低了包装器的破损率。

实验结果证明了本文方法的有效性。在未来的工作中，我们考虑包装器的自适应问题以及非结构化数据抽取问题。

参考文献

[1] LiuBing. Web Data Mining［M］,Yu Yong,Xue Guirong,Han Dingyi,Trans .Beijing Tsinghua University Press,2013.

（刘兵. Web数据挖掘［M］,愈勇，薛贵荣，韩定一，译.北京: 清华大学出版社,2013.）

[2] Nilesh N. Dalvi et. al. Robust web extraction: *an approach based on a probabilistic tree-edit model*. In SIGMOD, pages 335–348, 2009.

[3] Tobias Anton. Xpath-wrapper induction by generating tree traversal patterns. In LWA, 2005:126–133.

[4] M. Kowalkiewicz et. al. *Robust web content extraction*. In WWW,

2006:887–888.

[5] Marek Kowalkiewicz et. al. Myportal: *Robust extraction and aggregation of web content*. In VLDB, pages 1219–1222, 2006.

[6] L. Liu, C. Pu, and W. Han. XWrap: *An XML-enabled Wrapper Construction System for Web Information Sources*. Proceedings of the International Conference on Data Engineering, 2000:611-621.

[7] V.Crescenzi, G.Mecca, and P.Merialdo. Roadrunner: *Towards automatic data extraction from large web sites*. In Proc. of 27th Int. Conf. on Very Large Data Bases, 2001:109-118.

[8] Y. Zhai, , and B. Liu. Web data extraction based on partial tree alignment. In Proc. of WWW, 2005:76-85.