一个基于范例推理的时序预测模型*)

汤 胤 彭 宏 郑启伦
(广州华南理工大学计算机学院 广州510640)

摘 要 本文在相似模型的统一描述的基础上,提出一个多层次的抽象范例重用框架,适用于进行描述和时序的预测。在时间序列的问题下,本文描述了多层次范例推理的方法,并且讨论了一些CBR循环常见的问题在时序预测中的情况。本文最后提供一个期货预测的例子,对本文的模型作了说明。

关键词 范例推理,范例表示,时间序列预测,模式识别

A Case-Based Model for Time Series Extrapolation

TANG Yin PENG Hong ZHENG Qi-Lun
(Institute of Computer Science,South China University of Tech.,GuangZhou 510640)

Abstract In this paper current time series similarity models are described in a unified way,based on which a general hierarchical model is proposed for reusing cases at several levels of abstraction,within a time series extrapolating context. For time series similarity assessment,the paper shows how the case is represented,how cases are abstracted to higher lever and refined to lower lever,and how similarity assessment is performed in a single hierarchical framework, followed by several discussion of several key issues in CBR circle. Finally,an implementation on futures prediction using this framework illustrates the idea we developed.

Keywords Case-based reasoning, Case representation, Time series, Pattern recognition

1 引言

范例推理(Case-Based Reasoning)是近年来人工智能界的热点。简单地说,范例推理在历史范例的解决方案的基础上,对新问题进行匹配,利用旧方案得到新问题的解决方案。在时间序列预测中,CBR表现出与在其他应用领域不同的一些特性。

时序预测已有不少方法,如分解[14],指数平滑[4,27],随机模型[22],状态空间模型[1,25],贝叶斯模型[9],以及一些新方法如神经网络[13]和模糊逻辑[6,17,18],人工智能领域的遗传算法和模式识别方法等。Branko Pecar 的 APRE 方法[23]也对时序预测作了初步的尝试。

本文介绍一个多层次的时序范例框架,模仿人们思考的方式,递进式地接近相似性。Barry Smyth[29]在电厂控制软件设计中应用了。Ralph Bergmann[2]也解决了范例概化问题并提出了一个范例多层次重用一般框架。Michel Jaczynski[15]也做过类似的工作。然而,在时序预测这个问题上,范例的表示、匹配、重用问题有着很大的不同。

2 相似性模型与范例表示

2.1 相似性模型概览

与多数基于范例推理的系统一样,相似性度量都是其中的关键因素之一,在相似性度量上有许多方法。欧式距离(Euclidean Distance)将每个序列视为 n 维点的序列并计算其欧式距离,其复杂度为 $O(n)$;Goldin 与 Kanellakis[20]的距离标准化方法(Normalization method)将均值和方差标准化,以方便比较;动态时间弯曲(Dynamic Time Warping)[3]将时间拉长或缩短使得不同时间长度的序列可以比较;概率模型(Probabilistic Generative Modeling Method)[10]基于给出的序列 Q 构造概率分布模型 M_Q,然后通过计算 $p(Q'|M_Q)$ 来度量新的模式 Q' 的相似性;路标(Landmark Similarity[24])提出了新的数据表示方法,寻找最小特征集合,大大减少了存储和计算量。所有这些相似模型都有一个共同的特征,将原始的时间序列转换为不同的格式的数据,这些数据通常是元组的序列,很容易存储。图1中,利用 MA(Moving Average)以参数 s 进行转换,结果是一个新的时序 $\langle t,y\rangle$;DFT(Discrete Fourier Transformation) 也得到序列 $\langle t,a_0,a_1,a_2,a_3\rangle$;参数 p 和 q,路标模型转换后得到元组 $\langle t,y\rangle$ 的序列。

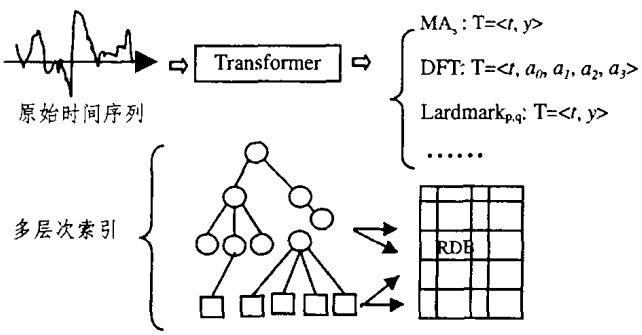


图1 时序相似模型统一描述

2.2 范例表示

范例 R 主要由两个时间点间的序列 (t_0,t_f) 和范例头 Header 组成。(1)Time Segment。指向时间序列实际存储位置

*)国家自然科学基金重点项目(30230350),广东省科技攻关项目(A1020103)。汤 胤 博士生,研究方向:人工智能与数据挖掘;彭 宏 教授,博导,研究方向:智能计算技术;郑启伦 教授,博导,研究方向:智能计算技术,神经网络理论及应用。

的指针，这部分还依赖于我们选择什么样的转换；(2)Record Context。范例头，带有范例名称，描述以及约束等；(3)Adjustment(可选)。调整曲线 Adjustment 主要描述了具影响力的外部因素的发生情况，一般调整曲线是某个时间粒度下的一段时间序列。

2.3 构造时序范例库的一般过程

不管数据如何表示和索引，序列都必须转换为同样格式的数据才可以比较，这里相似性模型实际上是起到了转换和类比的作用。

预处理：将序列分解为长期趋势，周期波动，季节波动，随机扰动等，从而得到平稳时间序列；

选择转换：DFT，ARMA(Auto-regression Moving Average)[4]等都可以作为转换协议，当然也可以用 MA 或 ES(Exponential Smoothing)。有时针对某个抽象层次的应用不同的转换器往往比对所有层次用一个转换器效果要好；

转换与存储：确定滑动窗口长度和间距，在每个滑动窗口用指定的模型做拟合，并将结果(比如 DFT 的转换结果是系数元组)索引并且概化，存入数据库中；

构造实际范例：实际时序存储后，需要从数据库中构造范例，包括调整曲线，范例头，以及时间段的指针，指向时间序列存储的尾部。比如，用 DFT 将序列转换，这样范例可能包括一个范例头描述范例的基本信息，一个起始时间指向数据库的某行，一个长度指定这个时间序列的长短，以及一个调整曲线。

构造范例抽象层次：在上述具体范例的基础上，我们在不同级别上实施概化(Abstraction)操作，具体见第3节。

3 多层范例库架构

HCBR[29]有不少好处：(1)多层次模型可以避免范例库平面组织带来的范例搜索的盲目性；(2)概化可以降低范例复杂度；(3)高层的范例可以作为原型并且做一族具体范例的替代。

3.1 抽象时序范例构造

3.1.1 模型

定义1(概化) $A^n(T)$定义为时间序列 T 的第 n 抽象层上的操作：

$$T^n=A^n(T^{n-1}) \tag{3.1}$$

$$T^0=A^0(T)=T \tag{3.2}$$

值得注意的是 A^n 操作应用于 T^{n-1}，是 A^{n-1}的结果，因此不同层的概化操作可以不同。我们可以选择如 MA、DFT、Landmark、ARMA 等等作为概化操作方法，比如：A^1：Moving Average，A^2：DFT，A^3：ARMA 等等。

$T^n(n>0)$称为抽象范例而 T^0称为具体范例，具体范例位于整个抽象层次的叶子节电上。

高于/低于：对两个范例 C^i 和 C^j，当 if $i>j$ 时我们说 C^i 高于 C^j，或 C^j 低于 C^i，或 C^i 比 C^j 更抽象。一个时序 s 比具体范例 C_2更相似于 C_1，当 s 匹配 C_2所在的最低层次比 s 匹配 C_2 所在的最低层次 C_1。从低向高层提取操作为概化(Abstraction)，从高向低称为求精(Refinement)。

不难推断，整个抽象层次呈现一个树形，具体范例为叶子节点。如果用 $H(n,S,A,C)$ 来代表相似模型 S 和概化操作集 $A=\{A^i\}$以及 S 和 A 操作对象范例集合 $C(i\in n,n$ 代表抽象的层次数)。

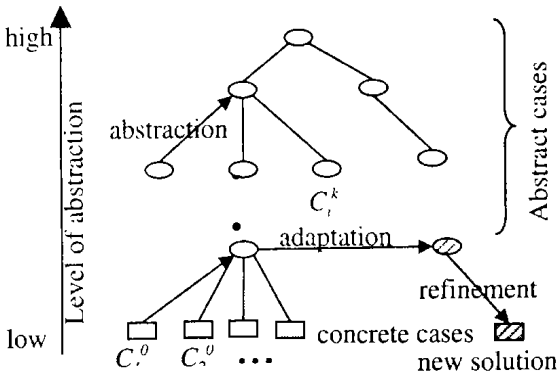


图2 范例库的多层次架构[2]

定义2(范例相似) 给定两个时序 a 和 b，及一个相似模型 M_t，以及一个阈值 ε，a 相似于 b，即 $a\approx b$，当且仅当

$$M_t(a,b)<\varepsilon \tag{3.3}$$

相似的范例获取：给定一个目标时序 $\vec{t}$，一个时序集 X，及一个相似性度量模型 d，和容许阈值 ε，寻找一个范例集合 C，使得：

$$C=\{\vec{x}\in X\mid d(\vec{t},\vec{x})\leqslant\varepsilon\} \tag{3.4}$$

3.1.2 问题描述

在上述定义基础上，时序范例预测的问题可以描述为：给定一个时序 T 和容许的阈值 ε，在多层次范例库中寻找范例集 C，使得对于任意的 $C_i\in C$，存在 C_i 的子序列，称为 C_i'，满足 $A^n(T)\approx C_i'$。

由于 C_i' 是 C_i 的一部分，我们使用减号"－"代表取 C_i 后部的操作，从 C_i' 的结束点起到 C_i 的结束点止。

$$T^e=\{T_i^e\mid T_i^e=C_i-C_i'\} \tag{3.5}$$

T^e 为最后预测的结果集。

范例获取算法：范例获取原则很简单，一个序列要与一个具体范例相似，首先要求它在高一层的抽象级别上与之相似。要相似首先要在相对粗糙的时间粒度上相似，然后才是细节上的相似。给定一个多层次的范例库 $H(n,S,A,C)$，从最粗糙的顶点开始，不断往下匹配，直到匹配条件不满足，这过程中匹配到的范例就是我们想要的结果。

3.2 抽象范例的构造

范例构造的方法主要是抽象范例通过概化操作(Abstraction)自动生成，不同层次的概化就形成了整个层次范例库；另外范例的构造方法是手工构造，应用在抽象范例无法自动生成时。

3.3 范例获取

为了匹配目标范例，我们从树根开始向下搜索。位于不同层次的抽象范例可以作为具体和抽象范例层次型的索引。在范例获取过程中，层次范例可以从上往下遍历，只要指向的抽象范例与当前的问题足够相似。我们的方法基于一个假设：一个问题不可能与一个具体范例相似，除非它首先与这个具体范例的相应的高层的抽象范例相似。范例获取有两种方案。

并行方案。序列 T 被概化后，不同层次的概化结果 T^0，$T^1,T^2,\cdots,T^n$ 进入相应的层次进行比较。比较是同步执行，结果也是同步输出的。根据比较的结果，我们就可以决定序列是否相似。

串行方案。序列 T 首先在 n 层上概化，得到结果 T^n。T^n 进入相应的 n 层进行比较，如果比较成功，T 就继续进行 A^{n-1}的概化，得到结果 T^{n-1}，这样不断比较直到到达具体层次 T^0或者相似条件不成立。

串行方案比并行方案耗费的 CPU 时间少，因为并行方案一些结果不一定有用。另外，串行方案遵循人类对相似的思考

方式，一步步地来决定相似性。

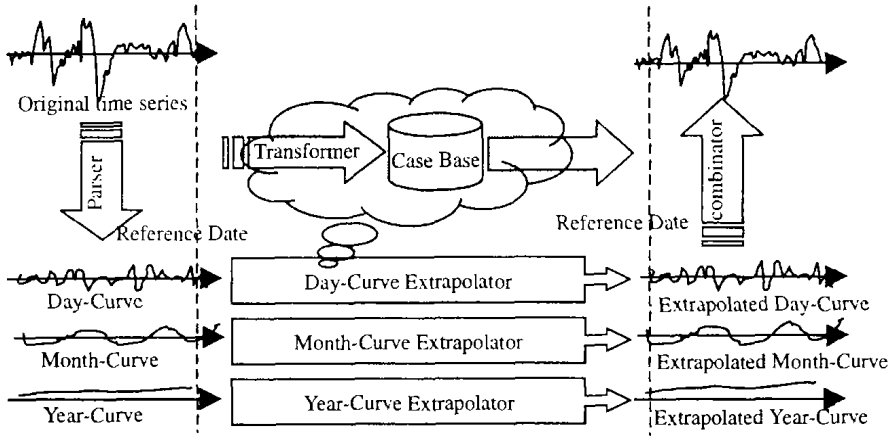


图3　在铜期货 LME copper 03的走热预测中应用我们的框架

3.4　抽象范例重用

利用抽象范例中的信息有很多种方式[2]。抽象范例除了用作具体范例的索引之外，我们还可以充分利用抽象范例的预测结果作为最终结果输出。CBR 系统可以获取并重用抽象范例，不将抽象范例的预测结果细化到具体范例的层面而是直接作为输出。

CBR 系统也可以尽量获取并重用抽象范例并细化到具体范例层面。根据上述讨论，重用的抽象范例层次越高，新生成的预测结果将越不同于初始范例的结果。

3.5　类比转换

类比转换要比别的步骤简单些，因为转换一个曲线趋势的事情仅仅是拷贝获取的范例的预测结果。很多已知的类比转换的方法都可以应用于抽象范例中，提高重用的弹性。

3.6　范例删除

经过一段时间的范例保存，范例库容量太大的时候，就要求一个实现范例删除的功能。在重用抽象范例的时候，范例删除可以以对整个树型组织结构剪枝的形式实现。

这是一个简单的想法，在一个父节点下面的抽象层上，除了那些它们父节点也拥有的特征之外，兄弟范例们不可以太相似。我们这样表达这个思想：给定一个阈值 ε，两个处在同一个父节点 P 下面的时间序列 X,Y,X,Y 之间的欧式距离必须满足：

$$L_p=(\sum_{i=1}^{n}|x_i-y_i|^2)^{1/2}>\varepsilon \qquad (3.6)$$

这里 x_i 是 X 的变量，y_i 是序列 Y 的变量。如果存在一些范例满足3.6式，时间相对较早的范例很可能就要被删除。

3.7　范例学习

预测的结果要求与实际走势做对比并得到评估结果。分析系统未能得到正确曲线(或相对不够相似的曲线)是一件比较困难的事情，需要对许多因素做细致的分析。与其他 CBR 应用系统不同的是，层次型 CBR 总是能找到一个结果，最坏的情况下也是一条相对相似的曲线。当类比转换成功后，新范例生成(见3.4和3.5式)，就必须添入范例库中。

4　框架应用举例

本框架应用在我们的一个期货走势预测的课题 FPS 中，系统在不同的处理点提供了一系列的处理器如：Parser，Combinator，Extrapolator，Transformer 和 Comparor 等等，执行特有的功能，如图3。

4.1　问题描述

本课题是关于期货走势的预测。这里我们假设期货合约间没有关联作用，选择1996年8月到1998年8月的铜期货 LME Copper 03天的500天的数据的前300天作为我们的初始序列，命名为 T_d。我们将初始数据输入系统，希望得到后200天的走势。范例库由1996年到2002年的一些金属期货的走势挖掘而成。

4.2　系统构成

4.2.1　预处理：Parser　为集中在我们的框架本身，我们简单地选择移动平均 MA(Moving Average)作为分解序列的手段。我们将初始序列 T_d 分解为3层，第一次移动平均参数为30，生成月线 T_m。选择参数的原则是，每个月有30天。同样地，以参数12对月线 T_m 做移动平均得到年线 T_y。设：

$$\nabla T_d=T_d-T_m \qquad (4.1)$$

$$\nabla T_m=T_m-T_y \qquad (4.2)$$

我们用 ∇T_m 代替 T_m，用 ∇T_d 代替 T_d，显然有：

$$T_d=T_y+\nabla T_m+\nabla T_d \qquad (4.3)$$

$$T_m=T_y+\nabla T_m \qquad (4.4)$$

与预处理相对应，一个后处理模块 Combinator 安置在处理流程的最后，借以合并三层预测的结果，得到最后的走势。

4.2.2　概化　我们设定3抽象层，定义操作 A^0,A^1,A^2 如下：

A^0 为初始序列(具体范例)T^0，不做任何实质性的运算；

A^1 为对 T^0 的移动平均，参数为30，生成 T^1；

A^2 为对 T^1 的移动平均，参数为12，生成 T^2。

4.3　转换器

转换器 Transformer 用于以指定的协议，将时间序列转换为制定的格式。我们采用 DFT(离散富利叶变换)对每个数据点进行扫描，得到4个系统，则生成的系数将形成4维空间一系列的点，与其他一些信息保存在库中。

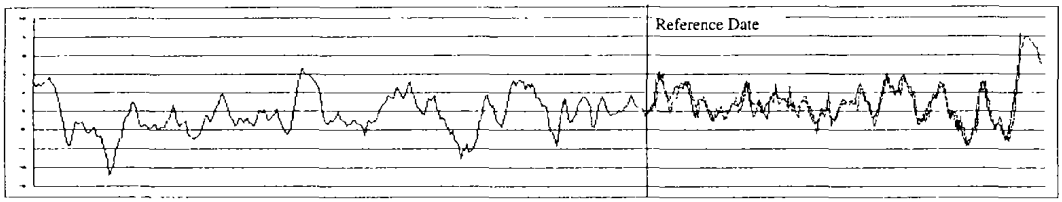
4.4　预测机

预测机 Extrapolator 对给定的时序片断进行前推预测。系统完成转换和对比后，得到若干个范例作为推测的基础。我们的系统可以获取并重用范例，或直接将抽象范例的预测结果返回作为结果，不过我们尽量将抽象范例细化到具体层面才输出(3.4)。图4为一些预测的结果。

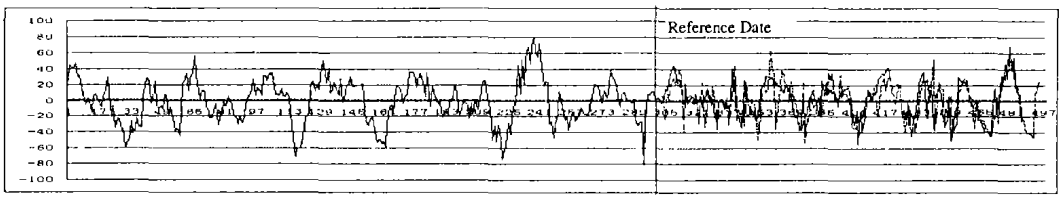
图4中，基于500天中前300天的 LME COPPER 03数据(从1996年8月份到1998年8月份)，预测出约200天的走势。虚线表示预测结果，实线表示实际的走势。

结论　本文提出的框架使得 CBR 能够在多个层次上重用范例，比较适合于时序预测问题。文章中讨论的问题有助于

建立时序预测的CBR模型。我们统一描述时序相似性模型，定义了数据表示和抽象、具体范例，并给出了构造时序范例的一般过程，定义了范例多层概化。最后我们在期货预测中应用了这个思想，取得了较好的效果。



月线 ΔTm 预测结果



日线 ΔTd 预测结果

图4 LME copper 03的预测结果

参 考 文 献

1 Harvey C. Andrew: Forecasting, Structural Time Series Models and the Kalman Filter. Cambridge University Press, Cambridge, 1989

2 Bergmann R, Wilke W. On the Role of Abstraction in Case-Based Reasoning. In EWCBR 1996. 28～43

3 Berndt, Clifford. Using Dynamic Time Warping to Find Patterns in Time Series. In: KDD Workshop, 1994. 359～370

4 Holt C C. Forecasting Seasonal and Trends by Exponentially Weighted Moving Averages. Carnegie Institute of Technology, Pittsburgh, Pennsylvania, 1957

5 Cole L, et al. Representing Cases for CBR in XML. In: UKCBR Workshop, Dec. 2000

6 von Constantin A. Fuzzy Logic and Neuro Fuzzy Applications Explained. Prentice Hall. Englewood Cliffs, 1995

7 Das G, et al. Finding Similar Time Series. In: Proc. of Principles of Data Mining and Knowledge Discovery, 1st European Symposium. Trondheim, Norway, Jun. 88～100

8 Faloutsos C, Ranganathan M, Manolopoulos Y. Fast Subsequence Matching in Time-Series Database. In: Proc. 1994 ACM SIGMOD Conf. Minneapolis, 1994

9 Finn J V. An Introduction to Bayesian Networks. UCL Press, London, 1996

10 Ge, Smyth. Deformable Markov model templates for time-series pattern matching. KDD 2000. 81～90

11 Hayes C, Cunningham P. Shaping a CBR view with XML. In ICCBR 1999. 468～481

12 Hetland M L. A Survey of Recent Methods for Efficient Retrieval of Similar Time Sequences. In: Mark Last, Abraham Kandel, Horst Bunke, eds. Data Mining in Time Series Databases. World Scientific, 2003

13 Rumellhart, Hinton, Williams. Learning representations by back-propagating errors. Nature, 1986. 323

14 Holger K, Schreiber T. Nonlinear Time Series Analysis. Cambridge University Press, Cambridge, 2000

15 Jaczynski M. A Framework for the Management of Past Experiences with Time-Extended Situations. In: 6th ACM Conf. on Information and Knowledge Management (CIKM'97), Las Vegas, Nov. 1997

16 Jagadish, et al. Similarity-Based Queries. PODS 1995. 36～45

17 Lefteri T H, Robert U E. Fuzzy and Neural Approaches in Engineering. John Wiley, New York, 1997

18 Lotfi Z A, et al. Fuzzy Sets and their Applications to Cognitive and Decision Processes. Academic Press, New York, 1975

19 Keogh, Pazzani. Scaling up Dynamic Time Warping for Datamining Applications. in ACM 2000. 1～58

20 Goldin, Kanellakis. On Similarity Queries for Time-Series Data: Constraint Specification and Implementation. CP 1995. 137～153

21 Leake D B. CBR in Context: The Present and Future. AAAI Press/MIT Press, 1996

22 George B E P, Jenkins G M. Time Series Analysis: Forecasting and Control. Holden Day, San Francisco, 1970

23 Pecar B. Case-based Algorighm for Pattern Recognition and Extrapolation. In: 22nd SGAI Int'l Conf. on Knowledge Based Systems and Applied Artificial Intelligence, Cambridge, Dec. 2002

24 Perng, et al. Landmarks: a New Model for Similarity-based Pattern Querying in Time Series Databases. ICDE 2000. 33～42

25 Kalman R E. A new Approach to Linear Filtering and Prediction Problems. Journal of Basic Engineering, D. 1960, 82

26 Rafiei, Mendelson. Querying Time Series Data Based on Similarity. In IEEE TRANS. ON KNOWLEDGE AND DATA ENG., 2000, 12(5)

27 Robert B. Statistical Forecasting for Inventory Control, McGraw Hill Book Co., New York, 1958

28 Sacerdoti E D. Planning in a hierarchy of abstraction space. Artificial Intelligence, 1974, 5: 115～135

29 Smyth B, Keane M T, Cunningham P. Hierarchical Case-Based Reasoning Integrating Case-Based and Decompositional Problem-Solving Techniques for Plant-Control Software Design. TKDE, 2000, 13(5): 793～812

30 Wijsen J. Trends in Databases: Reasoning and Mining. IEEE TRANS. ON KNOWLEDGE AND DATA ENG., 2001, 13(3)

31 Goldberg D E. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, 1989

(上接第70页)

参 考 文 献

1 Extensible Markup Language (XML). World Wide Web Consortium. http://www.w3.org/XML/

2 Concurrent Versions System (CVS). Free Software Foundation. http://www.gnu.org/manual/cvs-1.9

3 Chawathe S, et al. Change Detection in Hierarchically Structured Information. In Proc. of the ACM SIGMOD Intl. Conf. on Management of Data. Montreal, June 1996

4 Douglis F, Ball T, Chen Y F, Koutsofios E. The AT&T Internet Difference Engine: Tracking and Viewing Changes on the Web. World Wide Web, 1998, 1(1): 27～44

5 Berk E. HtmlDiff: A Differencing Tool for HTML Documents. Student Project, Princeton University. http://www.htmldiff.com

6 Barnard D T. Clarke G, Duncan N. Tree-to-tree Correction for Document Trees: [Technical Report 95-372]. 1995

7 Isert C. The Editing Distance Between Trees. 1999

8 Curbera D, Epstein A. Fast Difference and Update of XML Documents. XTech'99, San Jose, March 1999

9 Maruyama H, Tamura K, Uramoto R. Digest values for DOM (DOMHash) proposal. IBM Tokyo Research Laboratory. http://www.trl.ibm.co.jp/projects/xml/domhash.htm, 1998

10 Zhang, Shaha D. Simple Fast Algorithms for the Editing Distance between Trees and Related Problems. SIAM Journal of Computing, 1989, 18(6): 1245～1262

11 Cobéna G, Abiteboul S, Marian A. Detecting Changes in XML Documents. In: The 18th Intl. Conf. on Data Engineering, San Jose, Feb. 2002

12 Wang Y, et al. X-Diff: An Effective Change Detection Algorithm for XML Documents, 2002

13 Nierman A, Jagadish H V. Evaluating Structural Similarity in XML Documents, 2002

14 Zhang K. A New Editing based Distance between Unordered Labeled Trees. Combinatorial Pattern Matching, 1993, 1: 254～265