一种时间序列快速分段及符号化方法*)

任江涛[1] 何 武[1] 印 鉴[1] 张 毅[2]
(中山大学计算机科学系 广州 510275)[1] (清华大学自动化系 北京 100084)[2]

摘 要 作为一类重要的复杂类型数据,时间序列已成为数据挖掘领域的热点研究对象之一。针对时间序列的挖掘通常首先需要将时间序列分段并转变为种类有限的符号序列,以利于进一步进行时间序列模式挖掘。针对当前的时间序列分段方法复杂度较大,效率不高等问题,本文提出了一种简单高效的基于拐点检测的时间序列分段方法,并且采用动态时间弯曲度量计算不等长子序列的相异度,最后运用层次化聚类算法实现子序列的分类及符号化。实验表明,本文所提出的方法切实可行,实验结果具有较为明显的物理意义。

关键词 时间序列,拐点,符号化

A Fast Time Series Segmentation and Symbolization Method

REN Jiang-Tao[1] HE Wu[1] YIN Jian[1] ZHANG Yi[2]
(Department of Computer Science, ZhongShan University, Guangzhou510275)[1]
(Department of Automation, Tsinghua University, Beijing100084)[2]

Abstract As one of the important forms of complex data, time series is a hotspot in data mining area. Sequence pattern mining is based on time series symbolization, which segments the time series into sub-series and labels them. But most current time series segmentation algorithms are with large computation complexity, so the paper introduces a simple but high efficiency time series segmentation method based on change point detection. And dynamic time warping (DTW) method is used to compute the distance of the sub-series, later the hierarchical clustering is used to group the sub-series and label them. The experiments show the proposed method is feasible and the results are meaningful.

Keywords Time series, Change point, Symbolization

1 引言

时间序列是一种在科学研究和商业应用中普遍存在的数据形式,近期人们已开始关注时间序列数据挖掘算法的研究。针对时间序列的数据挖掘,通常要先将数据点密集的长时间序列转换成相对稀疏的具有特定含义的符号序列,即转化为相对简单且易于分析处理的形式,从而可以进一步应用诸如时态关联规则挖掘、序列模式发现等方法进行挖掘及知识发现。

为实现时间序列符号化,通常首先需要采用一定的方法将时间序列分段,然后采用聚类的方法将分段后得到的子序列进行分类,为每类赋予相应的类别标签,最后用类别标签序列代替原始数值序列,从而实现时间序列的符号化。其中,对时间序列进行分段的方法最为重要。文[1]提出了一种基于固定窗口的时间序列分段方法,文[5]提出了一种自底向上的线性化分段近似算法,文[2]提出了通过在一个滑动窗口中跟踪输入数据流概率密度变化来分割时间序列的方法,文[3]提出了一种全局迭代替换算法实现时间序列分段,文[4]提出基于进化计算的方法来实现时间序列的分段。

然而对于一般的时间序列,通常很难确定一个合适的窗口宽度使得所有时间序列分段都具有相对独立的模式[3,5],因此基于固定窗口的时间序列分段方法难以获得理想的结果。基于符号化目的的时间序列分段要求得到的每一个子序列都尽可能反映一定的状态模式,无论是基于线性分段,还是基于概率密度跟踪,以及基于进化计算的方法,都能较好地解决此问题,获得虽然长度可能不等,但确实能反映不同状态模式的分段。但这些方法有一个共同的缺点,就是算法复杂度高,特别是对于长时间序列,算法速度较慢。

针对上述问题,本文提出了一种时间序列快速分段及符号化方法,其基本思想是:首先利用拐点检测方法找到时间序列的拐点,从拐点处将时间序列分段,通过对序列递归地进行拐点查找及分段操作,将原时间序列转换为一系列分段,然后利用动态时间弯曲算法计算分段后得到的子序列的相异度并通过一定的转换构造出相似度矩阵,接着采用层次化聚类算法对各子序列聚类,根据聚类结果为每类子序列赋予相应的类别标签,最后用类别标签序列代替原始数值序列,从而得到原时间序列的符号化表示。

2 基于拐点检测的时间序列分段算法

针对引言中所提到的现有的时间序列分段算法主要采用线性分段、进化计算等等复杂算法,从而导致算法复杂度大、效率不高等特点,本文提出了一种基于拐点检测的时间序列分段算法,采用基于累积和控制图的拐点检测算法,并通过对该算法的递归调用实现时间序列的自动分段。

2.1 累积和控制图查找拐点算法

时间序列拐点是时间序列从一个模式变化到另一个模式的转变点,在时间序列分析中具有重要意义。拐点检测的方法很多,其中基于累积和(CUSUM)控制图的方法只需进行简单的加减运算就可完成,具有较高的效率,因此本研究采用该方法进行时间序列的拐点检测。

累积和(CUSUM)控制图方法的理论基础是序贯分析原理中的序贯概率比检验(Sequential Probability Ratio Test,简称 SPRT),其基本设计思想是通过对数据信息的累积,将过程中小的偏移加以放大,从而提高检测小偏移的灵敏度,该方

*)本研究得到国家自然科学基金项目(60374059)及广东省自然科学基金项目(04300462)资助。任江涛 博士,讲师。

法在工程控制、质量检测等方面有很广泛的应用[8]。基于累积和控制图的拐点发现算法如下：

算法 1 *CUSUM*(X)

输入：待处理的时间序列 $X=\{x_1,x_2,\cdots,x_n\}$

输出：拐点 x_c

步骤：

1)求序列均值：$\bar{x}=\sum_{i=1}^{n}x_i/n$；

2)设定累积和初值 $S_0=0$；

3)计算各点累积和 $S_i=S_{i-1}+(x_i-\bar{x})$，$i=1,2,\cdots,n$；

4)令 $|S_c|=\max\{|S_i|,i=1,2,\cdots n\}$，输出拐点 x_c。

下面以某一时间序列数据为例对 CUSUM 算法进行说明，该时间序列如图 1 所示。

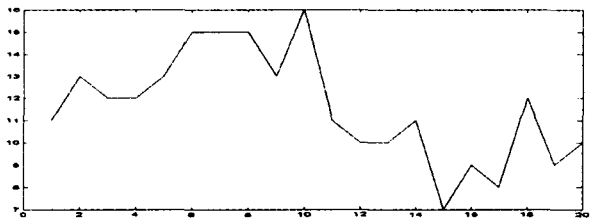

图 1　某时间序列图示

1)求序列均值：$\bar{x}=\sum_{i=1}^{n}x_i/n=11.6$；

2)设定累积和初值 $S_0=0$；

3)计算各点累积和，计算结果如图 2 所示；

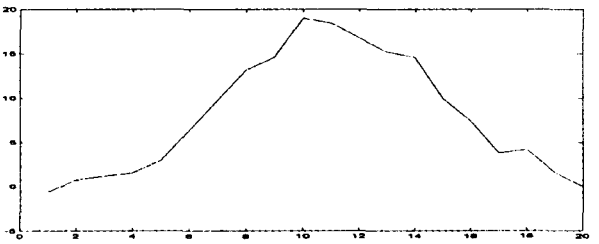

图 2　某时间序列累积和(CUSUM)控制图图示

4)从图 2 中可以看出，累积和在 x_{10} 处达到最大，即 $|S_{10}|=\max\{|S_i|,i=1,2,\cdots 20\}$，因此 x_{10} 正是所寻找的拐点。

2.2　基于拐点检测的时间序列分段算法

从以上分析可知，针对每个时间序列，都能采用 CUSUM 算法获得其拐点，从而将原时间序列分为具有明确物理意义的两部分，即后一部分相对于前一部分发生了转折性的变化。因此可以采用递归与分治的策略，针对一时间序列，不断地基于拐点进行二分，直到每一个分段长度小于一个预先设定的最小分裂长度阈值为止，从而实现对该时间序列的有效分段。

算法 2 *Segment*$(X,MinLength)$

输入：待处理的时间序列 $X=\{x_1,x_2,\cdots,x_n\}$，时间序列最小分裂长度阈值 *MinLength*

输出：时间序列分段

步骤：

1)针对时间序列 $x_1,x_2,\cdots,x_n$，调用 CUSUM 算法，得到拐点 x_c

2)基于 x_c，得到两个时间序列 $X_1=\{x_1,x_2,\cdots,x_{c-1}\}$ 及 $X_2=\{x_c,x_{c+1},\cdots,x_n\}$

3)若 $Length(X_1)\leqslant MinLength$，输出 X_1，否则执行 *Segment*$(X_1,MinLength)$

4)若 $Length(X_2)\leqslant MinLength$，输出 X_2，否则执行 *Segment*$(X_2,MinLength)$

由于本算法在执行时主要调用 CUSUM 算法，而 CUSUM 算法只进行一些简单的加减运算，因此基于本算法进行时间序列分段效率较高，尤其适合处理长时间序列。

3　时间序列符号化

采用上述算法将时间序列分段后，要进一步实现符号化，就需要采用聚类分析方法，对这些分段进行分组，然后对被分到各个组的时间序列进行标记，从而实现时间序列的符号化。

聚类就是根据样本之间的相似度将样本集合分为多个类或簇，要求同一个类中的样本之间有较高的相似度，不同的类中样本差别较大。选取合适的相似度度量标准对聚类的结果十分重要，由于通过上述算法分段后获得的时间序列分段不等长，可采用动态时间弯曲度量[9]作为不等长的时间序列之间的不相似性度量，然后通过一定的方法将其转化为相似性度量。

3.1　动态时间弯曲度量(Dynamic Time Warping, DTW)

动态时间弯曲度量最初应用于文本和字符串匹配及视觉模式识别等领域的不相似性度量方法，研究表明这种基于非线性弯曲技术的算法可以获得较高的识别及匹配精度。其具体定义如下：

设现有二时间序列 P 和 Q，长度分别为 m 和 n，即

$P=\{p_1,p_2,\cdots p_m\}$

$Q=\{q_1,q_2,\cdots q_n\}$

首先给出如下定义：

定义 1(距离矩阵)　矩阵中的元素为不同时间序列数据对象之间的点的欧氏距离，记为 $d(p_i,q_j)$。

矩阵元素 $d(p_i,q_j)$ 是二个时间序列数据点之间的欧式距离，可视为对象 P_i 与对象 q_j 相异性的量化表示。当对象 p_i 与对象 q_j 越相似或越接近，其值越接近 0；两个对象越不相似，其值越大。

定义 2　在二个不同时间序列间的距离矩阵中，定义时间序列间相异性关系的一组连续的矩阵元素的集合，称为弯曲路径，图 3 中的粗线则给出了弯曲路径的一个例子。

$W=w_1,w_2,\cdots,w_k\quad \max(m,n)\leqslant k\leqslant m+n-1$

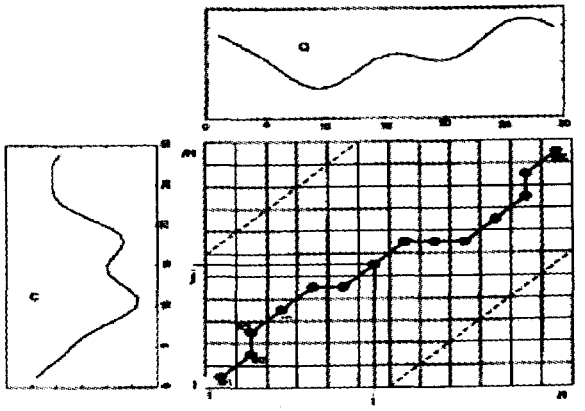

图 3　动态时间弯曲路径示例

对距离矩阵分析可知，弯曲路径存在多解，但是人们关心的仅仅是弯曲总长度最小的路径，即以两序列相似性程度最

大(距离值最小)作为相似搜索的判据,如下式:

$$DTW(P,Q)=\min(\sum_{i=1}^{k}w_i/k)$$

式中分母 k 是为了保证在比较不同长度的路径时有统一的标准。

理论上,可以利用穷举搜索法寻找满足条件的弯曲路径,但是完全穷举在大数据量分析中往往是不切实际的,因为路径的解很多,且与距离矩阵中的元素数呈指数增长关系。由动态规划理论可知,设点(i,j)在最佳路径上,那么从起始点$(1,1)$到点(i,j)的子路径也是局部最优解,即从起始点$(1,1)$到点(i,j)的最优路径可由起点$(1,1)$到终点(i,j)间的局部最优解递归搜索获得,即:

$$S_{1,1}=d(p_1,q_1)$$

$$S_{i,j}=d(p_i,q_j)+\min\{S_{i-1,j},S_{i-1,j-1},S_{i,j-1}\}$$

最终时间序列弯曲路径最小累加值为 $S_{i,j}$,$S_{i,j}$即为两序列的相异度。从点(i,j)起沿弯曲路径按最小累加值回溯至起始点$(1,1)$即可找出整个弯曲路径。

3.2 相似性度量定义

由于动态时间弯曲度量给出的是序列间的相异度,而通常层次化聚类算法基于相似度矩阵,因此本文提出如下相似度定义:

$$SIM_{x_i,x_j}=1-\frac{DTW(X_i,X_j)}{\max(DTW)+1}\in(0,1] \quad (1)$$

$$\max(DTW)=\max\{DTW(X_i,X_j),i,j=1,2,\cdots n\}$$

其中 $DTW(X_i,X_j)$为子序列 X_i 与 X_j 的动态时间弯曲距离,n为子序列个数,当两子序列完全相等,即 $DTW(X_i,X_j)=0$时,$SIM_{X_i,X_j}=1$。

3.3 聚类算法

在聚类分析领域,存在着大量的算法,算法的选择取决于数据的类型、聚类的目的和应用领域等。根据本文所研究问题的特点,选择层次化聚类算法对经过分段且建立起相似度矩阵的时间序列样本对象进行聚类分析。

4 实验结果

本研究采用来源于美国华盛顿州高速公路监控系统的道路占用率时间序列数据来验证所提算法的可行性,具体选用路网中某检测点在 2004 年 2 月 18 日至 3 月 10 号的道路占有率时间序列数据,如图 4 所示。

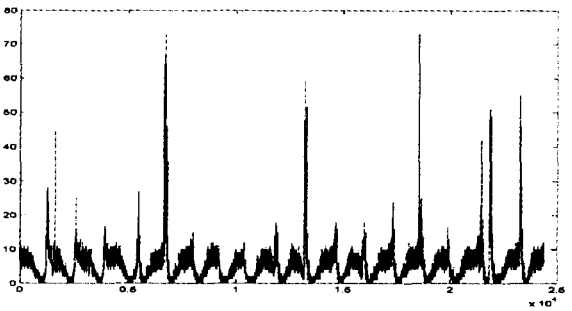

图 4 某交通检测点在 2004 年 2 月 18 日至 3 月 10 号的道路占有率时间序列数据图示

实验过程如下:

1)调用 2.2 节所提出的 Segment 算法,对该时间序列进行分段,得到 409 个子序列(子序列最小分裂长度阈值设为 100);

2)调用动态时间弯曲度量算法,计算两两序列的不相似度;

3)采用式(1)将不相似度转化为相似度,建立相似度矩阵;

4)调用层次化聚类算法,对分段后得到的子序列样本进行分组;

5)分组后得到 7 类样本,并赋予类标志符,形成符号化的时间序列。

取 7 类序列的代表分段,分别示于表 1 及图 5(截取分段的前 20 个点示出)。

表 1 7 类序列代表分段(前 20 个数据点)

0	0	0	2	0	1	1	2	1	0	1	1	0	1	0	1	1	0	0	0	1
1	2	2	2	3	2	1	3	2	2	4	1	3	2	2	1	3	2	2	2	3
2	5	7	7	5	6	5	8	7	8	9	8	7	6	9	8	7	5	7	8	8
3	5	6	4	4	4	4	2	5	4	4	3	3	3	4	2	3	4	3	4	3
4	13	14	16	12	13	14	12	14	14	13	18	25	19	19	20	21	20	19	18	21
5	16	16	20	20	20	20	18	17	20	18	17	21	25	25	28	26	25	18	26	21
6	39	42	51	67	57	65	65	54	38	46	47	33	44	40	44	41	47	55	50	41

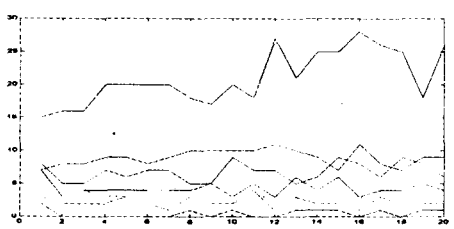

图 5 7 类序列代表分段(前 20 个数据点)

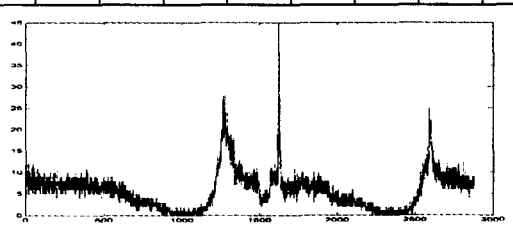

图 6 某交通检测点两天道路占有率时间序列图示

从表 1 中可以看出,各类分段后得到的时间序列样本具有较为明显的物理意义,代表不同的交通拥堵特征。如第 6 类时间序列样本代表突发性交通拥挤情况,第 5 类时间序列代表一般交通拥堵状态,而第 0 类则表示道路基本处于空闲状态。

图 6 代表其中两天的时间序列数据,被分割成 48 个子序列,符号化的结果如图 7 所示。

对比图 6 与图 7,可以发现图 7 中的符号化结果与图 6 中的变化趋势基本上是吻合的,如图 7 中序列加粗表示的第 5、第 6 及第 4 类子序列代表图 6 中的三个交通流量高峰。

结论 针对现有的时间序列分段及符号化方法计算复杂度较高，运行效率较低等问题，本文提出了一种基于拐点检测的时间序列分段及符号化方法。由于基于累积和控制图的拐点检测方法只需进行简单的加减运算，因此大大提高了时间序列分段的效率，而且基于拐点的序列分段具有较为明确的物理意义。在此基础上，运用动态时间弯曲度量及层次化聚类算法，实现了针对分割后得到的子序列的有效分类及符号化。实验分析表明所提出的方法切实可行，分析结果具有较为明显的物理意义。

4 2 2 2 2 2 2 2 2 2 2 2 3 1 1 0 0 0 0 0 0 1 4 5 4 2 2 3 6 2 2 2 2 2 3 1 1 1 0 0 0 0 1 4 4 2 2 2 2

图 7 图 6 中所示时间序列的符号化表示

参 考 文 献

1 David D G, Heikki M, et al. Rule Discovery from time Series. In: Proc. Fourth annual Conference on Knowledge Discovery and Data Mining
2 Kohlmorgen J, StevenLemm. An on-line method for segmentation and identification of non-stationary time series. In NNSP 2001: Neural Networks for Signal Processing XI, 2001. 113～122
3 Himberg J, Korpiaho K, et al. Time series segmentation for context recognition in mobile devices. 2001 IEEE International Conference on Data Mining (ICDM 2001), San José, California, USA, 2001. 203～210
4 Fu T-C, Fulai, Vincent Ng C, Luk R. Evolutionary segmentation of financial time series into subsequences. In: Proc. of the 2001 Congress on Evolutionary Computation CEC2001. 426～430
5 李斌，谭立湘，章劲松，庄镇泉. 面向数据挖掘的时间序列符号化方法研究. 电路与系统学报，2000，5(2)
6 (加)韩家炜，等著，范明，等译，数据挖掘：概念与技术. 机械工业出版社，2001，8
7 Hand D，等著，张银奎，等译，数据挖掘原理. 机械工业出版社，2003，4
8 WAYNE A. TAYLOR Baxter Healthcare Corporation. Round Lake, IL 60073 Change-Point Analysis: A Powerful New Tool For Detecting Changes. http://www.variation.com/cpa/tech/changepoint.html
9 Keogh E J, Pazzani M J. Scaling up dynamic time warping to massive datasets. In: Proc. of the 3rd European Conf. on Principles of Data Mining and Knowledge Discovery (PKDD), 1999. 1～11

(上接第 148 页)

定义 1 设 I 为自然数集，$R=\{r_i\}i\in I$ 是用户角色集，$U=\{u_i\}$ 是用户集，$P=\{p_i\}$ 是许可集。

定义 2 用户--角色赋值关系：$UA\subseteq U\times R$，$(u,r)\in UA$，表示普通用户 u 享有角色 r；角色-许可赋值关系：$PA\subseteq R\times P$，$(r,P)\in PA$ 表示角色 r 包含许可 P。

在系统实现时，可以定义三张表，实现 RBAC 模型，其中许可集表 P 记录所有许可定义，用户表 U 记录用户实例及用户-角色赋值关系，用户角色表 R 记录所有角色的定义及角色-许可赋值关系。这样可以实现一个用户一种角色，一种角色一种访问许可关系。也可以通过增加定义 UA 表和 PA 表，方便地实现一个用户多角色，一种角色多种访问许可的机制。

5 OUKM 应用实例

OKMF 是一个非结构化知识管理系统的核心开发平台，可以应用到不同的具体领域，也可以支持面向不同领域的应用功能扩展开发。我们已将其应用两家高新技术企业。其目的有四个方面：一是管理已有的各种技术知识文档；二是用于管理外部知识资源，三是支持员工的隐性知识向显性知识的转化。四是支持对企业知识的评价。

在应用到高技术企业时，在 OUKM 核心系统上扩充了两个应用子系统：一是面向知识工者的人力资源管理子系统，主要通过考核知识工作者对企业提交的自己创作或发现的有用知识的数量和质量，可以评估个体对企业的知识贡献，鼓励知识工作者主动把自己隐性知识提交给企业，可以建立知识共享的管理机制；二是知识评价子系统，用来评价知识的价值，帮助提高知识工作者检索和使用知识的效率，为进一步实现知识主动推送奠定基础。

系统功能有以下几方面：

OUKM 系统可以解决高技术企业普遍存在的几个方面问题：一是有效管理企业知识资源，改变过去知识共享和重用率低的问题；二是用于支持企业知识积累，解决由于知识工作调动或造成的知识资源力流失问题；三是通过管理外部知识资源，提高企业创新速度；四是可以改善企业文化，作为支持逐步建立学习性组织的基础设施。

该系统具有如下特点：

(1)应用的普遍适应性。系统是按本体论和知识管理的一般原理设计的，对于不同企业或组织，实质上是生成不同的本体实例。只需根据企业领域特点，分析企业领域本体和企业组织结构及管理方法，通过系统定义，生成适应本企业的知识管理体系的应用系统。

(2)应用功能的可扩展性。系统预留面向二次开发的应用程度接口，可以满足基于知识管理的其它功能扩展。

(3)较好的安全性。知识是企业的重要资源。在提供共享机制的同时，需保护企业资源的恶意访问或破坏，本系统采用 RBAC 理论相结合的方法设计并实现系统的安全子系统。

(4)知识检索一定程度的智能性。由于本体支持基于描述逻辑的推理，OUKM 知识管理系统可以借助本体的推理功能，实现检索的一定程度智能化，并可随本体支持系统的发展面提高检索的智能化程度。

(5)知识库具有较好的可重用性、可共享性和互操作能力。首先是知识体描述是基于 DC 基础上进行剪裁和扩展进行设计的，可与任何遵循 DC 的知识管理系统实现数据的共享和互操作。系统知识以本体方式存储，可在后续其它知识相关系统中重用。

结论 本文讨论了非结构化知识管理的意义，并讨论了以本体论为基础建立知识管理系统的方法。根据本文所提出的方法，建立了一个非结构化知识管理平台。系统是知识管理的核心系统，系统可根据用户应用需要，扩展以知识管理为核心的应用与管理功能。系统还有多方面需要改进之处。如何加强知识体的内容管理、如何表现用户的知识需求并实现知识的主动推送、如何将 RBAC 与 MAC 模型结合提高系统安全性是我们未来研究的目标。

参 考 文 献

1 顾芳，曹存根. 知识工程中的本体研究现状与存在的问题. 计算机科学，2004，31(10)：1～10
2 李飞，高济，等. OKMF：一个基于本体论的知识管理系统框架. 计算机辅助设计与图形学学报，2003，15(12)：1538～1548
3 王英林，王卫东，等. 基于本体的可重构知识管理平台. 计算机集成制造系统一 CIMS，2003，9(12)：1136～1144
4 王德禄. 知识管理的 IT 实现. 电子工业出版社，2003. 86
5 王小明. 时态角色委托代理授权图模型及其分析研究：[西北大学博士学位论文]. 2004(4)：6～7
6 Mike Uschold, Martin King. Towards a methodology for building ontologies. http://citeseer.ist.psu.edu/uschold95toward.html
7 Noy F, Deborah L. McGuinness Ontology Development 101: A Guide to Creating Your First Ontology Natalya. (http://www.ksl.stanford.edu/people/dlm/papers/ontology101/ontology101-noy-mcguinness.html).
8 Mizoguchi R. part1 Tutorial on ontological engineering. New Generation Computing, OhmSha&Springer, 2003, 2(14): 365～384
9 Mirzaee V, Iverson L, Hamidzadeh B. Towards Ontological Modelling of Historical Documents. http://protege.stanford.edu/conference/2004/abstracts