Journal of Web Engineering, Vol. 13, No.5&6 (2014) 483-506

WEB EVENT STATE PREDICTION MODEL: COMBINING PRIOR KNOWLEDGE WITH REAL TIME DATA

XIANGFENG LUO JUNYU XUAN HUIMIN LIU

Shanghai University

luoxf@shu.edu.cn xuanjunyu@shu.edu.cn a60695420@163.com

Received April 22, 2014
Revised July 8, 2014

The state prediction plays a key role in the evolution analysis of web events. There are two issues for the state prediction of web events: one is what factors impact on the state transition of web events; and the other is how the prior knowledge can guide the state transition of web events. For the first issue, we discuss two types of temporal features observed from the real time webpages covering an event, i.e., the statistical ones and the knowledge structural ones. For the second issue, Fuzzy Cognitive Map (FCM) and conditional dependency matrix are mined from the training web events. As the prior knowledge, they represent the relations between the states transition and the relations of unobserved space (i.e., the six states of web events) and observed space (i.e., the two types of features). Based on that, an improved hidden Markov model is developed to predict the state transition of web events. Experimental results show that the model has good performance and robustness because it combines the prior knowledge and the real time data of web events.

Key words: web event, hidden Markov model, topic detection and tracking, multi-factor analysis
Communicated by: B. White & D. Lowe

1 Introduction

As the development of the web, the message publishing is so easy that huge volume of information has been surging on the web. We know that people are more and more willing to know and discuss the social events with each other on the web. For example, they browse webpages to know what happens today and write some blogs or posts to express their opinions about the society on the web. Therefore, the event on the real social activities can be imaged to the web with real-time. Sometimes, a web event even has no corresponding mirror in the real world and only exists on the web, such as the rumour web events. As for the web events, there are some states from their emerging to disappearing, such as the latency state, the outbreak state and the decline state. All the web events may share some common states and they are just different combinations of these states in the evolution process of web events. Prediction on the following state is significantly useful and practical for many real-world tasks, like,

1. For websites, they can increase their hit-rates by attracting users through the web event state prediction;

2. For the government or organizations, they can monitor the development of some specific web events.

For the web events, they have their prior knowledge which may hide in their evolution traces. The prior knowledge is like which state will emerge after a given state, which is not considered in the time series features. The Fuzzy Cognitive Map (FCM) is a graphical model of prior knowledge representation, which comprises concepts and their cause relations. However, FCM cannot acquire knowledge from the real time data because it lacks the self-adaptation ability to the changed environment [10, 11]. The prior knowledge, which is the relation between state transitions and the conditional dependency from the unobserved space (i.e., the six states of web event evolution) to the observed space (i.e., the two types of temporal features), can be mined from the training web events. It can be stored by FCM and conditional dependency matrix, respectively. Therefore, we can use FCM and conditional dependency matrix to store the prior knowledge of web events.

Hidden Markov Model (HMM) has the advantage of adopting the observed features from real time data, and the disadvantage is that it cannot obtain the prior knowledge (i.e., the unobserved states). Meanwhile, the FCM can store the prior knowledge and it is difficult to adopt the real time data. For increasing the accuracy of the web event state prediction, we need to combine the prior knowledge with the real time data. In this paper, we proposed a state prediction model which combines the prior knowledge with the observed features (i.e., the statistical features and the knowledge structural features) extracted from the real time webpages covering with one web event.

The contributions of this paper are summarized as follows,

1. Two categories of features of web events are defined to describe the different states of web events, including statistical features and knowledge structural features;
2. A HMM-based prediction model has been proposed to predict the states of web events based on the defined two categories of features.

The rest of the paper is organized as follows. Section 2 reviews some related work. State space is discussed in Section 3. Observed features including statistical features and the knowledge structural features are developed in Section 4 and 5, respectively. How to build the prior knowledge of web events is introduced in Section 6. The prediction model is proposed in Section 7. Experimental results and their analyses are discussed in Section 8. Conclusion is given in the last Section.

2 Related Work

The study of the evolution of web events mainly belongs to the field of Topic Detection and Tracking (TDT) [12-14]. The main tasks of this research field are the event detection and the evolution process tracking from the huge data[15, 16], like daily news webpages. However, it does not do any further analysis about the detected events. Some existing public opinion analysis systems do the further analysis[16, 17], which mainly focus on the sensitive event identification[18, 19] and sentimental analysis [20, 21]. In this paper, we focus on the state prediction of web events in their evolution process and discuss what factors impact on the state prediction. To our knowledge, this work is not included in the current research of TDT.

The prediction is a very difficult and challenging work. Normally, current methods have two classes: 1) causal forecasting methods and 2) time series methods. The former methods use the causal relationship between the predictive variables and other variables, like Simple Linear Regression Prediction, Multiple Linear Regression Prediction and Nonlinear Regression Prediction [22]. For example, Radinsky [23] uses newsfeeds to generate news story sequence. The later methods use the structure of historical data, like moving average method[a], exponential smoothing method[b], and autoregressive integrated moving average model [2], KALMAN model [3] and neural network prediction method [4]. There are also works which try to combine these two kinds of methods, like hybrid model [24]. But the challenge issues of the above methods are that how to find the causal relationship and how to identify the structural rule from historical data. The states of events are not considered. Prediction has also been widely used in other problems [33, 34, 35].

HMM[25] has been widely used for sequential pattern recognition in many areas, like speech recognition [26], human behaviour analysis [27] and action recognition [37], handwritten word recognition [28], pitch and formant tracking [38] and so on. There are three main factors: hidden states transition matrix, dependency matrix and initial state. It can not only mine the hidden states transition, but connect the observed features with the hidden states. Traditionally, the hidden states transition matrix and dependency matrix need to be learned from the historical data for prediction. A web event can also be seen as a hidden state sequence. The only thing we can directly get is the observed features, like the number of webpages, the number of keywords and so on. HMM can well connect these features with the hidden states of web events.

3 State Space of Web Event

There are many kinds of events in the real world, like natural disasters, security incidents, conflict of nations. Although they have different evolution processes, it is generally believed that the six states are shared by them, including 1) latent state, 2) outbreak state, 3) decline state, 4) increasing transition state, 5) decreasing transition and 6) stable transition[c]. Since the web event is the image of the event in the real world, it can also be believed that web events also share the same states.

The evolution process of a web event is just the composition of different states. At the very beginning of a web event, there are little persons and websites taking care of it, so the webpages covering this web event are seldom. As the evolution of the web event, more and more people, websites and forums are involved in this web event. The number of webpages and posts increase dramatically, the outbreak state emerges. After the outbreak state, the web event gets stable and the discussion about it decreases. It comes to its end, and the decline state is coming. During this evolution process, the states won’t directly transit to other states. There is usually a transition state between states of web events.

Since the six states can represent web event in its evolution process, they are selected as the state space of web event. This space is an unobserved space that needs us to detect from the evolution processes of web events. Next, the basic definitions of the six states are given.

[a] http://en.wikipedia.org/wiki/Moving_average
[b] http://en.wikipedia.org/wiki/Exponential_smoothing
[c] http://en.wikipedia.org/wiki/Emergency

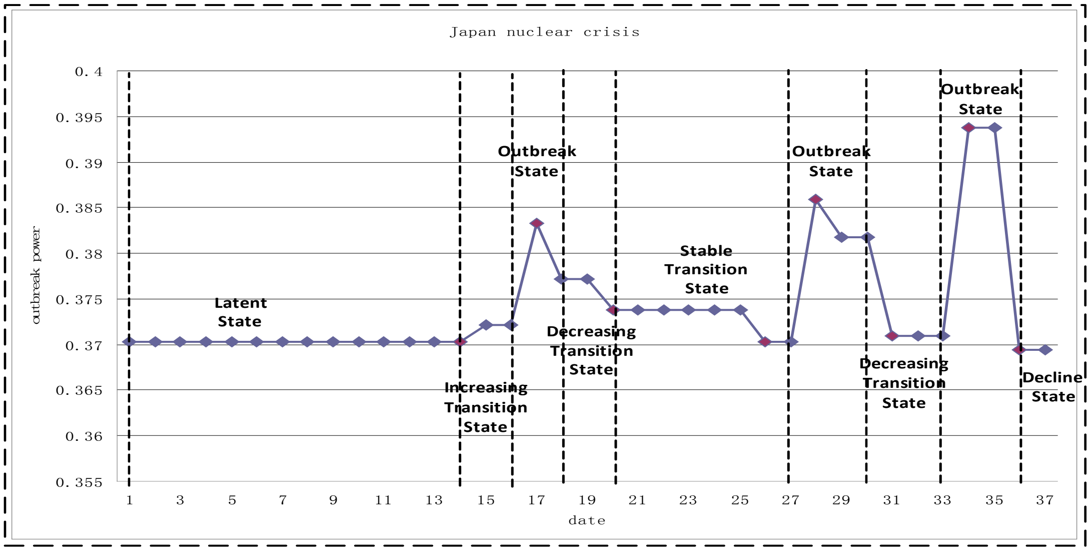

Figure 1 An example of state space of a web event, in which the x- axis shows the timeline of the event and the y- axis describes the urgent degree of this event. The web event 'Japan nuclear crisis' has eight states in its event evolution process.

DEFINITION 1 Latent state ls_e

Latent state is the starting state of a web event. At this state, there are often little webpages, few event attributes[d] *and low update speed of the webpages and attributes.*

In this state, web events are not paid much attention by people. But they do have emerged. As more and more webpages or people involved in, this web event may transfer to other states such as outbreak state.

DEFINITION 2 Outbreak state, os_e

Outbreak state is in the middle of the whole evolution process of a web event. At this state, there are often a lot of webpages and attributes and the update speed of the webpage and attributes is high.

In this state, the discussion about a web event reaches its peak. Based on the huge number of involved people, the webpages of this web event will increasingly emerge to talk different aspects of it. After the outbreak state, the decline state may be appearing.

DEFINITION 3 Decline state, ds_e

Decline state is at the end of the whole evolution process of a web event. At this state, update speed of the webpages and attributes falls off.

[d] A keyword can be seen as an attribute of the web event, and the attribute of a web event can be described by keywords. In this paper, each attribute of a web event is corresponding to a keyword.

In this state, most of attributes of a web event have already been discussed. The number of attributes of the web event, which may catch people's attention, is decreasing. People start to lose their interests on the web event.

DEFINITION 4 Transition state, ts_e

Transition state is in the middle of the whole evolution process of a web event as the connection of different other states. According to the connected states, it is divided into three categories:

1) Increasing transition state, ts_{inc} *;*

2) Decreasing transition state, ts_{dec} *;*

3) Stable transition state, ts_{sta} .

Generally, increasing transition state is seen as the connection of ls_e and os_e; decreasing transition state is seen as the connection of os_e and ds_e; stable transition state is the connection of ts_{inc} and ts_{dec} . These transition states could be seen as the 'buffer' of other states. An example of a web event's state space is shown in Fig. 1.

All of these states together constitute the unobserved state space of web event. The evolution process of a web event is composited by the above six states.

DEFINITION 5 State space of web event, ρ

State space of web events is the set of all the possible states in the evolution process of a web event,

$$\rho = \{\omega_1, \omega_2, \mathrm{L} \ , \omega_n\} \tag{1}$$

and

$$\omega_i \in \{ls_e, os_e, ds_e, ts_{inc}, ts_{dec}, ts_{sta}\} \tag{2}$$

where the element ω_i *in* ρ *is a possible state of web event; n is the number of the live state of a web event.*

The state space of web event is not directly observable. What and how we can get the observable features of these states are the key issues for the state prediction of web events. In the next two sections, two categories of features, statistical ones and knowledge structural ones, are developed to predict the state of web event in its evolution process.

4 Statistical Features Obtained From the Real-time Data

Statistical features detect the changes of the real time web event in a fixed time interval, such as one day or one hour. For example, the average number of the increased webpages, the attribute transmission speed and the attribute update speed, etc. In this section, we will describe the above features. All introduced features are summarized in Table 1.

Table 1: *All* Feature Symbols

symbol	*Statistical features*	*symbol*	*Structural features*
$\left\|\overline{\varphi}\left(t_i,t_j\right)\right\|$	Average number of the increased webpages	*alp*	Attributes link power
$\Delta\overline{K}\left(t_i,t_j\right)$	Average number of the increased attributes	*mla*	Maximum length between attributes
$\Delta K\left(t_i,t_j\right)$	Average number of the attributes	*ac*	Attribute clustering coefficient
$\overline{\vartheta}\left(t_i,t_j\right)$	Attribute transmission speed	*ace*	Attribute centrality
$\mu\left(t_i,t_j\right)$	Attribute update speed	*ad*	Attribute density
		acc	Attribute community clustering

It should be noticed that the value of this feature could be big or small at stable transition state. Relative to other states, the values of features do not have much change during this state.

Statistical feature 1: The average number of the increased webpages from time t_i to t_j.

$$f_1^{st}:\left|\overline{\varphi}\left(t_i,t_j\right)\right|=\frac{\left|\varphi\left(t_i,t_j\right)\right|}{t_j-t_i} \tag{3}$$

where $\varphi\left(t_i,t_j\right)$ is the set of increased webpages from time t_i to t_j; $\left|\overline{\varphi}\left(t_i,t_j\right)\right|$ is the average number of increased webpage.

It is easy to understand the meaning of this feature. We know that the probability of a web event being at outbreak state or increasing transition state increase with the growing of the number of increased webpages. Therefore, we can get the conclusion: the probabilities of os_e states ts_{inc} and are proportion to $\left|\overline{\varphi}\left(t_i,t_j\right)\right|$; the probabilities of ds_e states ts_{dec} and are inverse proportion to $\left|\overline{\varphi}\left(t_i,t_j\right)\right|$.

With the reduction of the average number of increased webpages, the probability of web events being on decline state or decreasing transition state increases.

The increased attributes are the new content during this time evolution of web event. Sometimes, there are many new increased webpages but they are just the reprint of former webpages. Therefore, we need to compute the average number of increased attribute that can reflect the evolution speed and direction of a web event.

Statistical feature 2: The average number of the increased attributes from t_i to t_j

$$f_2^{st} : \Delta \overline{K}\left(t_i,t_j\right) = \frac{\left|\overline{K}\left(t_i,t_j\right)\right|}{t_j - t_i} \tag{4}$$

where f_2^{st} is the average number of increased attributes of web event; $\overline{K}\left(t_i,t_j\right)$ is the set of increased attributes from t_i to t_j.

Same with feature 1, the probability of a web event being on outbreak state or increasing transition state increases with f_2^{st}. So, we can get the following conclusion: the probabilities of os_e states ts_{inc} and are proportion to $\left|\Delta\overline{K}\left(t_i,t_j\right)\right|$; the probabilities of ds_e states ts_{dec} and are inverse proportion to $\left|\Delta\overline{K}\left(t_i,t_j\right)\right|$.

According to Eq.4, as the increased attributes of a web event decrease, the probability of web events being on decline state or decreasing transition state increases.

Statistical feature 3: The average number of the attributes of web event from time t_i to t_j.

$$f_3^{st} : \Delta K\left(t_i,t_j\right) = \frac{\left|K\left(t_i,t_j\right)\right|}{t_j - t_i} \tag{5}$$

where f_3^{st} is the average number of the attributes of web event from time t_i to t_j; $K\left(t_i,t_j\right)$ is the set of the attributes of web event from t_i to t_j.

Comparing to $\overline{K}\left(t_i,t_j\right)$, $K\left(t_i,t_j\right)$ reflects not only the new emerging attributes of a web event, but also the remaining part included the former attributes of a web event.

At the outbreak state, the more attributes a web event has, the more content and subtopics it has. The probability of a web event being on outbreak state or increasing transition state increases with the growing of the average number of the increased attributes, and vice versa. So, we can get the conclusion: the probabilities of os_e states ts_{inc} and are proportion to $\Delta K\left(t_i,t_j\right)$; the probabilities of ds_e states ts_{dec} and are inverse proportion to $\Delta K\left(t_i,t_j\right)$.

Statistical feature 4: Attribute transmission speed.

$$f_4^{st} : \overline{\vartheta}\left(t_i,t_j\right) = \frac{\left|\varphi\left(t_i,t_j\right)\right|}{\left|K\left(t_i,t_j\right)\right|} \tag{6}$$

where $\overline{\vartheta}\left(t_i,t_j\right)$ is the average number of the webpages that each increased attribute belongs to.

Eq. 6 reflects the average transmission power of all the event attributes. The more webpages an attribute belongs to, the bigger this attribute's transmission power is. The summation of transmission powers of all attributes reflects the transmission power of the whole web event.

We know that the bigger attribute transmission speed is, the bigger probability of this web event on the outbreak state or increasing the transition state has, and vice versa. So, we can get the conclusion: the probabilities of os_e states ts_{inc} and are proportion to $\overline{\vartheta}\left(t_i,t_j\right)$; the probabilities of ds_e states ts_{dec} and are inverse proportion to $\overline{\vartheta}\left(t_i,t_j\right)$.

Statistical feature 5: Attribute update speed.

$$f_5^{st}:\mu\left(t_i,t_j\right)=\frac{\left|K\left(t_i,t_j\right)\right|}{\left|\overline{K}\left(t_i,t_j\right)\right|} \tag{7}$$

where $\mu\left(t_i,t_j\right)$ is the ratio of the increased attributes and all the attributes.

Attribute update speed reflects the average update speed of attributes. Considering the two extreme situations of Eq. 7: if $\mu\left(t_i,t_j\right)$ is equal to 1, it means all the attributes are new born and the web event is evolving fiercely; if $\mu\left(t_i,t_j\right)$ is equal to 0, that means there is no new born attributes and the web event do not evolve during this time interval.

We know that the probability of a web event being at outbreak state or increasing transition state increases with the growing of the attribute update speed of a web event. So, we can get the following conclusion: the probabilities of os_e states ts_{inc} and are proportion to $\mu\left(t_i,t_j\right)$; the probabilities of ds_e states ts_{dec} and are inverse proportion to $\mu\left(t_i,t_j\right)$.

According to Eq.7, the probability of the web event being at decline state or decreasing transition state increases, as the decrease of the attribute update speed of a web event,.

In the above discussion, we leave a question that how to discriminate os_e from increasing transition state ts_{inc}, and decline state ds_e from decreasing transition state ts_{dec}. Usually, if the value of the statistical feature is more larger, then the state appearing probability $p(os_e)$ is more larger than $p(ts_{inc})$; if the values of the statistical features is more lower, then the state appearing probability $p(ds_e)$ is more lower than $p(ts_{dec})$.

5 Knowledge Structural Features Obtained From the Real-time Data

Since the statistical features are only the external expression of web events' states, they are not enough to do state prediction alone. We need to get the underlying knowledge structure of web events, which comes from the content of web events and can represent the semantics of web events. Therefore, the knowledge structural features of different states can also contribute to the state prediction. For the evolution of a web event is normally determined by the content of this web event, this knowledge structure is also even more important than statistical ones. In this section, attribute association link network (A2LN) [29] is used to model this knowledge structure and some complex network features of this network are selected as knowledge structural features for web events.

Since A2LN is the basic of all the knowledge structural features, we give a brief introduction of the construction of A2LN here. With all the webpages of a web event in hand, we firstly extract the keywords from these webpages. Then, traditional association rules between these keywords are mined. Finally, the keywords are linked together by their association rules, and then A2LN has been constructed.

Different from the statistical features, the knowledge structural features of web event more focus on the association relations of event's attributes. Furthermore, the whole knowledge structure is also determined by attributes' interaction with each other. To some extent, it is the association relations of attributes that determine the evolution process of a web event. Therefore, A2LN, a complex network consisted by the association relations of web event attributes, is constructed [30]. It reflects the underlying knowledge structure of a web event. Then the features of A2LN reflecting the knowledge structure features of web event are given, including attributes link power, maximum length between attributes, attribute clustering, etc. Before the discussion of the knowledge structural features, we need to define A2LN first, as shown in Fig. 2.

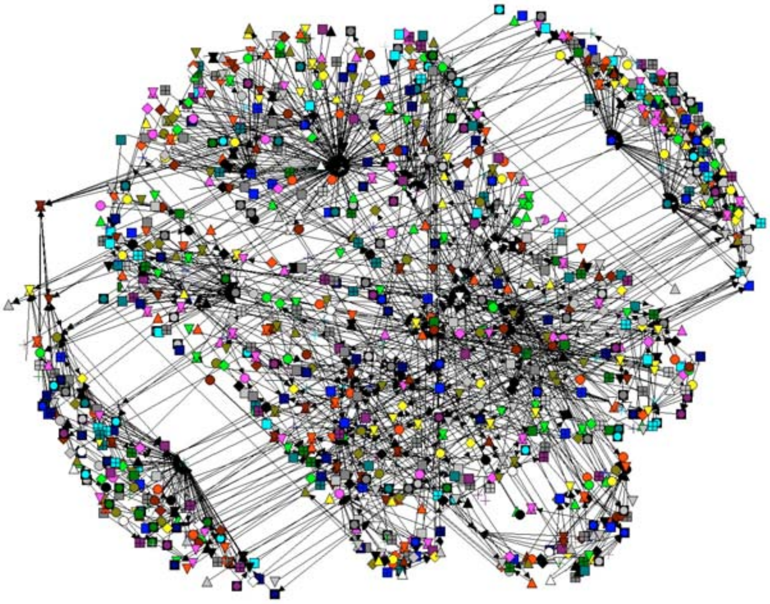

Figure 2 An attributes network of a web event. The nodes represent attributes and the arcs represent association rules between attributes.

DEFINITION 6 Attribute Association Link Network (A2LN), $G_{\langle V,E\rangle}\left(t_i,t_j\right)$

$$G_{\langle V,E\rangle}\left(t_i,t_j\right)=\begin{pmatrix} r_{11} & \mathrm{K} & r_{1n} \\ \mathrm{M} & \mathrm{O} & \mathrm{M} \\ r_{n1} & \mathrm{L} & r_{nn} \end{pmatrix} \tag{8}$$

where V is the attribute set of a web event and E is the association relation set; r_{ij} is the association relation between attribute i and j, which are extracted from the webpages covering one web event and range from [0, 1].

After the definition, some important knowledge structural features of this complex network are selected to describe web events.

Structural Feature 1: Attributes link power, alp

$$s_1^{ks}: alp = \frac{1}{N}\sum_i k_i \tag{9}$$

where k_i is the degree of attribute i (the number of nodes connecting this node i) in the attribute association link network of a web event; N is the number of attributes.

This knowledge structural feature reflects the associated link status of a web event's attributes. When alp is small, that means the attributes of a web event have little association relations with each other. They have not formed a whole and closed web event. At this time, the web event will have a larger probability in latent state or decline state; and lower probability in outbreak state or increasing transition state. So, we can get the following conclusion: the probabilities of os_e states ts_{inc} and are proportion to alp; the probabilities of ds_e states ts_{dec} and are inverse proportion to alp.

Structural Feature 2: Maximum length between attributes, mla

$$s_2^{ks}: mla = Max\left\{L_{i,j}\right\}, \left(i, j \in V, i \neq j\right) \tag{10}$$

where $L_{i,j}$ is the length between attribute i and j in a web event attribute association link network (the minimum number of links that connect two nodes).

This knowledge structural feature shows the maximum distance between two attributes of a web event. It reflects the diversity of a web event. The bigger this distance is; the more diverse evolution this web event is. The sub-events of this web event are very different from each other. This web event has big evolution power and more various states. So, we can get the following conclusion: the probabilities of os_e states ts_{inc} and are proportion to mla; the probabilities of ds_e states ts_{dec} and are inverse proportion to mla.

Structural Feature 3: Attribute clustering coefficient, ac

$$s_3^{ks}: ac = \frac{1}{N\left(N-1\right)}\sum_{i,j\in V, i\neq j} L_{i,j} \tag{11}$$

where $L_{i,j}$ is the length between attribute i and j in a web event attribute network.

This feature describes the clustering of all the attributes in the web event attribute association link network. The big of the ac means that the attributes have more association relations. The power of the evolution process is weak and the states are stable or the event focuses on a specific sub-event. So, we can get the following conclusion: the probabilities of os_e states ts_{inc} and are proportion to ac; the probabilities of ds_e states ts_{dec} and are inverse proportion to ac.

Structural Feature 4: Attribute centrality, ace

$$s_4^{ks}: ace = \frac{1}{N}\sum_{i=1}^{N}\frac{N-1}{\sum_{j=1}^{N} L_{i,j}} \tag{12}$$

where ace reflects the average centrality of all attributes; $L_{i,j}$ is the length between attribute i and j in a web event attribute network; N is the number of attributes.

All the attributes have the probability to be the main sub-event of a web event. This feature reflects the trend of all the attributes to be centres. The low ace means the event has various sub-events that might lead to a big evolution. So, we can get the following conclusion: the probabilities of os_e states ts_{inc} and are inverse proportion to ace; the probabilities of ds_e states ts_{dec} and are proportion to ace.

Structural Feature 5: Attribute density, ad

$$s_5^{ks}: ad = \frac{2\cdot|E|}{N(N-1)} \tag{13}$$

where $|E|$ is the number of association rule of web event attribute association link network; N is the number of attributes.

This feature reflects the tightness of all attributes. When it equals 1, it means the different sub-events of a web event all have the association relations with each other. The lower the ad is, the more relaxing of the attributes are. The live state of the web event is maybe in the latent state or transition state. So, we can get the following conclusion: the probabilities of os_e states ts_{inc} and are inverse proportion to ad; the probabilities of ds_e states ts_{dec} and are proportion to ad.

Structural Feature 6: Attribute community clustering, acc

$$s_6^{ks}: acc = \frac{1}{N}\sum_{i\in V}\frac{2\varpi_i}{k_i(k_i-1)} \tag{14}$$

where k_i is the degree of attribute i and ϖ_i is the number of arcs between neighbours of attribute i in an attribute association link network.

The high value of this feature shows that the similar or very associated attributes have trend to get together in the evolution of a web event. This feature is a good metric to measure this kind of trend of getting attributes together. There may be some overlap with other structural features. But they all have their own perspectives. For example, density, average clustering coefficient and average path length all have the “density” meaning in describing the network structure. However, density focuses on the whole network’s density, and average clustering coefficient focuses on the local clustering phenomenon, and average path length focuses on the distance of attributes. So, we can get the following conclusion: the probabilities of os_e states ts_{inc} and are inverse proportion to acc; the probabilities of ds_e states ts_{dec} and are proportion to acc.

During the evolution process of a web event, different states consist of the whole evolution process of a web event. Although these states are not observable, there are some observable features with the real time webpages. The different compositions of these features constitute the observed space of a web event, which can be used to predict the unobserved state space.

DEFINITION 7 Observed space of web event obtained from the real time data, O

Observed space of a web event is the set of all the possible feature compositions of a web event, including knowledge structural features and statistical features.

$$O = \{o_1, o_2, \ldots, o_m\} \tag{15}$$

and

$$o_i \ \mathrm{B} \left\langle f_1^{st}, f_2^{st}, f_3^{st}, f_4^{st}, f_5^{st}, s_1^{ks}, s_2^{ks}, s_3^{ks}, s_4^{ks}, s_5^{ks}, s_6^{ks} \right\rangle$$

and

$$o_i \ \mathrm{B} \left\langle \begin{array}{l} \left|\overline{\varphi}\left(t_i, t_j\right)\right|, \Delta \overline{K}\left(t_i, t_j\right), \Delta K\left(t_i, t_j\right), \overline{\vartheta}\left(t_i, t_j\right), \mu\left(t_i, t_j\right), \\ alp, mla, ac, ace, ad, acc \end{array} \right\rangle$$

where o_i is the composition of all the observable features including statistical features and knowledge structural features; and m is the number of the observable features.

It can be seen from the definition that the observed space describing a web event from the different perspectives. The statistical features more focus on the ‘volume change’ of the web events. On the contrary, the knowledge structural features more focus on the ‘nature change’. There may be some overlap between different features, but they can be the complementary of each other as the following experiments shows. Next, we simply see these features are independent from each other.

6 Prior Knowledge Mined from the Training Web Events

There are two types of prior knowledge in the evolution process of a web event. One is the association relations between web event state transitions, and the other is the conditional dependency between the unobserved (state) space to the observed (feature) space. Here, we use State Transition Matrix (STM) to represent and store the first type of prior knowledge, and use the Conditional Dependency Matrix (CDM) to represent and store the later type of prior knowledge, respectively.

6.1. Mining State Transition Matrix

For a web event, it has its own prior knowledge which may hide in its evolution trace. The prior knowledge of state transition is like which state will emerge after a given state. It is not considered by time series features. Herein we use State Transition Matrix to store this kind of prior knowledge, which comprises states and their association relations. It can be mined from the training web events.

As discussed in above section, we know that to get the prior knowledge about the association relations between the states is the key issue for the state prediction of a web event. The state space of web events is defined in Section 2. A whole evolution process of a web event is just the transitions between the six states in the state space. In this section, we will propose a technique for the mining of transition rules from the state time series of the training web events by Apriori algorithm [31]. Transition rules can be recorded by STM.

DEFINITION 8 State transition matrix of web event, K_n

$$K_n = \begin{pmatrix} c_{1,1} & \mathrm{K} & c_{1,n} \\ \mathrm{M} & \mathrm{O} & \mathrm{M} \\ c_{n,1} & \mathrm{L} & c_{n,n} \end{pmatrix} \tag{16}$$

and

$$c_{ij} = q\left(\omega_j \mid \omega_i\right)$$

where $q\left(\omega_j \mid \omega_i\right)$ *shows the transition degree from state* ω_i *to state* ω_j *and* n *is the number of states.*

Eq. 16 represents the state transition prior knowledge of web events. It is obtained from many training web events, so it expresses a general characteristic of the state transitions of web events.

After the above discussion, we define the evolution process of a web event is a state time series, which can be denoted as,

$$Life_e = < \omega_1, \omega_2, ..., \omega_l > \tag{17}$$

where $\omega_l \in \rho$ is the evolution state and l is the state number of a web event during its evolution process.

For example, the evolution process of a web event, 'Japan Earthquake Crisis', is divided into eight states as shown in Fig. 1, in which the first state is latent state and the second one is the outbreak state etc. Herein, the problem we concern has referred to the mining of transition rules from the states of the training events' webpages.

Here, Apriori algorithm is adopted to mine the frequent temporal patterns and to generate transition rules. We obtain these rules from the time series of web events' states. We seek for the identification of frequent patterns in these state time series. A temporal pattern is defined as a set of states together with their transition relationships. A simple example of the transition rule format is: If ω_i occurs, then ω_j occurs with transition degree q.

where $\omega_i, \omega_j \in \rho$ are two states of a web event in its evolution process. We denote the above rule as $r : \omega_i \xrightarrow{q(\omega_j|\omega_i)} \omega_j$.

Given a state time series of a web event, the frequency $q(r)$ of a transition rule r is the number of occurrences of $r \in Life_e$; the support $supp(r)$ of rule r is $Q(r)/k$. The confidence $conf(r)$ of the rule r is defined as,

$$q(\omega_j \mid \omega_i) = conf(\omega_i \to \omega_j) = \frac{Q(\omega_i, \omega_j)}{Q(\omega_i)} \tag{18}$$

where $q(\omega_j \mid \omega_i)$ means that the appearance of state ω_i leads to the degree of appearance of state ω_j; $Q(\omega_i)$ is the occurrence times of ω_i in the training events' state time series; and $Q(\omega_i, \omega_j)$ is the co-occurrence times of ω_i and ω_j in the training events' state time series.

If $q(\omega_j \mid \omega_i)$ exceeds a threshold δ, we regard that appearing of ω_i will lead to the appearing of state ω_j with degree $q(\omega_j \mid \omega_i)$. Our task is to find all the frequent temporal patterns in the training web events, from which we can obtain the transition rule set which can be used to build the STM.

The mined frequent transition patterns can be stored by State Transition Matrix as the prior knowledge of the web event's state transition in its evolution process.

Although the State Transition Matrix-based prior knowledge representation method can describe the general state transitions of web events, but it has low accuracy to predict the following state since it miss the real time data. So the State Transition Matrix alone is not enough, we further introduce the observed features which have the emission probability with the real time evolution states of web events. In the experiments, we will compare State Transition Matrix alone with our method discussed in section 8.2.

6.2. Mining Conditional Dependency Matrix

The first part of prior knowledge has been discussed in section 6.1, which are the state transition rules represented by STM. In this section, the second part of prior knowledge will be constructed, which is the prior knowledge that connects the web event state space with the feature space. This kind of prior knowledge plays a key role in the state prediction of web events.

DEFINITION 9 Conditional Dependency Matrix, Z_{emi}

The dependency relation is defined as the emission probability between the feature space and the state space, which can be stored by,

$$Z_{emi} = \begin{pmatrix} b_{11} & \cdots & b_{1m} \\ \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{nm} \end{pmatrix} \tag{19}$$

and

$$b_{ij} = p\left(o_j \mid \omega_i\right)$$

where b_{ij} *is the probability of state* ω_i *having the observed features* o_j *;* $n = |\rho|$ *is the state number of the feature space and* $m = |O|$ *is the number of the possible composition of features in observed space.*

The emission probability exists between the web event state space and the feature space, which means that a certain observed feature may be a result of an evolution state. For example, "if the attribute update speed $\overline{\vartheta}\left(t_i, t_j\right)$ is higher, then the web event has higher probability under the outbreak state". It is obvious that the former observed feature is the result of the latter unobserved state.

The prior knowledge of web events composed by the emission probabilities between the two spaces is stored by conditional dependency matrix. In addition, each element of the conditional dependency matrix indicates the emission probability between the observed feature space and the unobserved state space. Suppose $\partial_j(\omega_i)$ denotes the number of the occurrences of observation result o_j under the state ω_i, then the element b_{ij} is obtained by ,

$$b_{ij} = p(o_j \mid \omega_i) = \frac{\partial_j(\omega_i)}{\sum_j \partial_j(\omega_i)} \tag{20}$$

The conditional dependency matrix contains the emission probability that an observation result can be observed when the event is under a specific state, which represents and stores the emission probability between the unobserved state and the observation space. In the matrix, the i^{th} row vector indicates the emission probabilities between unobserved space and the observed space when the state is ω_i; the j^{th} column vector indicates the weights of the emission probabilities when the observation is o_j.

Through the classified statistics of the occurrences of observation results under different states on our training web events, we can discover an observation features may occur when the events are under state ω_i. Therefore, given an effect observation, we can find all the possible states or all the pairs of the emission probability mapping $\omega_i \to o_j$ from the conditional dependency matrix Z_{emi}, which is the basis of the prediction of the evolution of web events. Then the conditional dependency matrix can be obtained from the training web events as the second part prior knowledge in our prediction model.

7 Prediction Models

7.1. Use Prior Knowledge Alone

Herein, State Transition Matrix (STM) is used to predict the evolution state of web event alone. This model includes two parts: the current state vector and the state transition matrix. Then the next time state of web event can be predicted as:

The current state at time t will influence the state at time $t+1$. According to the characteristic of State Transition Matrix, this influence can be computed by,

$$\begin{aligned} \lambda_{t+1} &= \left| h(\omega_1) \quad h(\omega_2) \quad \ldots \quad h(\omega_n) \right|_{t+1} \\ &= \left| h(\omega_1) \quad h(\omega_2) \quad \ldots \quad h(\omega_n) \right|_{t} \times K_n \end{aligned} \tag{21}$$

where λ_{t+1} is the state vector of a web event at time $t+1$, in which $h(\omega_i)_t$ is the transition degree of a web event being in current state ω_i and K_n is the transition matrix,

By iteratively using STM, it can be applied to long-term prediction, that is:

$$\begin{aligned} &\lambda_1 = K_n \cdot \lambda_0 \\ &\lambda_2 = K_n \cdot \lambda_1 = K_n^2 \cdot \lambda_0 \\ &\ldots\ldots \\ &\lambda_k = K_n^k \cdot \lambda_0 \end{aligned} \tag{23}$$

According to Eq.23, the prediction of state is determined by the current state and the state transition matrix. By the way, the evolution process may approach the steady-state, cycle state or chaos state after a certain period of time. The limitation of this kind of prediction is that it does not use the real time data. This limitation may lead to a low accurate of prediction since the web event is a dynamic event.

7.2. Combined Prior Knowledge with Real-time Data

The observed features, the statistical ones and the knowledge structural ones, are defined as the observed space of HMM. Different states have different features obtained from the real time data of web event. In this section, STM-represented prior knowledge of web event's state transition will be embedded into HMM as the hidden state transition matrix.

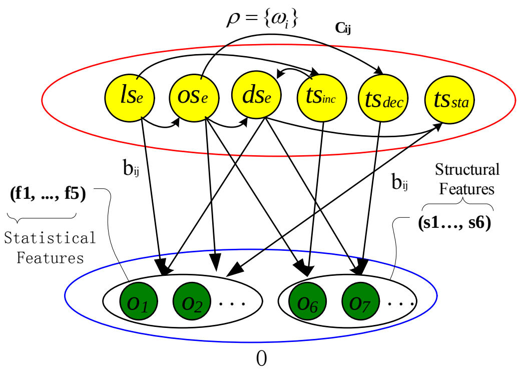

Figure 3 The framework of the prediction model that combining prior knowledge with real time data. The conditional dependency matrix (b_{ij}) and the states transition matrix (c_{ij}) are combined to reflect the prior knowledge of all the web events and the real time data of a web event.

Hidden Markov Model (HMM) can build the connection between the observed features and the unobserved states of an object. When we do state prediction, what we can get is just the observed features of a web event. In order to predict the states of a web event, the connection needs to be built between the real-time observed features and the unobserved states (i.e., the state transition space) of a web event. So, HMM is a good choice to do that. In this paper, the observed features of web event reflect its real time state, which is represented by the real time observed space of HMM.

The conditional dependency matrix can be regarded as the emission probability in HMM, which is a kind of representation of association relation between ω_i and o_j. Based on that, the HMM-based model, combining the prior knowledge with the real time data, is developed to predict the state transition of web event as shown in Fig. 3.

When the state of time $t+1$ needs to be predicted, we have the features at time $t+1$ (green balls in Fig.3) observed and the state of time t (yellow balls in Fig.3). Firstly, we can predict by state transition prior knowledge (link between yellow balls shown in Fig.3). Secondly, we predict by observed features (link between yellow ball and green ball in Fig.3). At last, two prediction results are multiplied together as the final result. Next, the more concrete process will be given.

At first, the next time state will be influenced by the current state. This influence can be computed by,

$$\lambda_{t+1} = K_n \cdot \lambda_t \tag{24}$$

where λ_{t+1} is the state vector of a web event at time $t+1$, which describes the probability of web events being in each state at time $t+1$. It can be transformed as,

$$p(\omega_i)_{t+1} = \sum_{j=1}^{n} p(\omega_j)_t \, p(\omega_i \mid \omega_j)^{\text{e}} \tag{25}$$

Secondly, the next time state λ_{t+1} will also be influenced by current observed feature o_k^{t+1}. It can be computed by,

$$\lambda_{t+1} = Z_k^{\mathrm{T}} \cdot o_k^{t+1} \tag{26}$$

where λ_{t+1} is state vector of a web event at time $t+1$, o_k^{t+1} is the observed features at time $t+1$; Z_k is the conditional dependency matrix. It can be transformed as,

$$P(\omega_i)_{t+1} = P(\omega_i \mid o_k) \cdot o_k^{t+1} \tag{27}$$

The states at time $t+1$, which are predicted by Eq.24 and Eq.26, can be combined as,

$$\lambda_{t+1} = K_n \cdot \lambda_t \cdot Z_k \cdot o_k^{t+1} \tag{28}$$

or

$$P(\omega_i)_{t+1} = \sum_{j=1}^{n} P(\omega_j)_t \, P(\omega_i \mid \omega_j) P(\omega_i \mid o_k) \cdot o_k^{t+1} \tag{29}$$

where $P(\omega_i)_{t+1}$ is the probability of a web event being in state ω_i.

According to the above discussion, the state prediction of web event has two steps:

1) Initialize the observed space O and train conditional dependency matrix Z_{emi} and the state transition matrix K_n from training web events;

2) At the time $t+1$, the state will be predicted by Eq.29.

8 Experiments and Discussions

We will do two experiments. One is the prior knowledge alone based state prediction; and the other is the prior knowledge combined with the real time data.

In the former discussion, the observed space representing the real time data is introduced, including two kinds of features: knowledge structural features and statistical features. Different features have their own ability to describe the real time statue of web events from various sides. Therefore, in the HMM based experiments, three strategies are made to predict the state of web events as following,

S1: Knowledge structural features alone;

S2: Statistical features alone;

[e] When the training data is enough, $q(\omega_j \mid \omega_i)$ shown in Eq.16 can be regard as $p(\omega_i \mid \omega_j)$.

S3: Both the knowledge structural features and statistical features.

Through the comparison of three different strategies, we can see the different performance of the two kinds of features. By the way, the latent state is the start of web events, which cannot be predicted, which is the web event identification task not the task of this paper. So this state is removed from the prediction task.

8.1. Data Set

The data used in this experiment is from the Chinese biggest search engine - Baidu News[f]. We selected 50 web events including 450,000 webpages. The detailed information is listed in Table 2. Because the whole evolution process of web events will be used in the experiments, the start time and the end time of web events need to be identified first. The start time is detected by [32]. The end time is the maximum time shown in the downloaded webpages.

Table 2: Date Set Description

Average number of webpages per web event	1763
Average number of attributes per web event	6118
Average number of days per web event	42
Average number of webpages per day	59
Average number of states per event	10
Average number of days per state	5

8.2. Prediction Experiments

Herein, the results of two experiments are reported. First, the effectiveness of the prediction model using ALN alone is presented. Then, the efficiency of applying the HMM which combines the prior knowledge with the real time data to the states prediction is demonstrated, along with three different strategies.

In order to verify our state prediction model, we need to build a comparison standard. For the 50 web events in the data set, their different states during their own evolution process are identified. For example, the evolution process of event, ‘Japan Earthquake Crisis’, is divided into eight states as shown in Fig. 1. If our state prediction for the third state of this web event is ‘decreasing transition state’, it means our prediction succeeds. For this web event, there are eight states. If the number of successful prediction is 6, the success rate is 6/8. Next, we do the same prediction task to all 50 web events which contain about 500 states in all.

8.3. Results and Analysis

Table 3: Results used only prior knowledge represented by ALN

ALN	os_e	ts_{inc}	ts_{dec}	ts_{sta}	all
Accuracy	90%	23%	71%	8%	47%

[f] http://news.baidu.com

The prediction result only using State Transition Matrix is shown in Table 3, from which we can see that the accuracy of different evolution states differ greatly from each other and the whole prediction accuracy is also relatively low. It can be concluded that the State Transition Matrix based model is difficult to predict the certain states. The reason for that the evolution details of real time data of web events have been ignored. So, we can see from Table 3, the stable transition state is very lower than other states by the State Transition Matrix-based prediction model.

Table 4: Results used combine prior knowledge with real time data

		os_e	ts_{inc}	ts_{dec}	ts_{sta}	all
S1	Num	40	49	22	14	125
	Accuracy	72%	70%	65%	64%	69%
S2	Num	43	54	22	18	137
	Accuracy	80%	82%	65%	67%	73%
S3	Num	49	60	29	24	162
	Accuracy	91%	91%	77%	75%	81%

The result that the HMM based model combining prior knowledge and the real time data is used to predict the state is shown in Table 4. From the results, we can see that the accuracy of this prediction model has been improved significantly since the observed features make great contributions to the prediction of web events. Moreover, we can see that the strategy S1 is more accurate than the strategy S2. It can be conclude that the knowledge structural features are more powerful in expressing the states of web events than the statistical features.

In all the states prediction, the strategy S1 is better than the strategy S2. The reason is that the knowledge structural features can reflect the nature and underlying characteristic of web events. So the prediction based on that has better accuracy. At the same time, we can see from Table 3, the strategy S3 is the best one. The accuracy of the third strategy is 81%, which is more accuracy than strategy S2 (73%) and strategy S1 (69%). That result suggests that the statistical features can offer the complement to the knowledge structural features in the prediction task despite having worse accuracy than it.

The stable transition state is the hardest state to predict than others as shown in Table 3. We think it depends on the different properties of states. Transition state is in the middle of evolution process of web events, including three different trends. It takes the function to lead the turning of web events. The other states have their clear and distinctive properties but the transition state lies in different states. The property of it is so obscure that it is very hard to predict through the observed features.

Comparing Table 3 and Table 4, regardless strategies of the prediction model, the accuracy of the model combining prior knowledge with the real time data is higher the one that only using prior knowledge (the accuracy is 47%). The experimental results show our proposed model has a very good performance than the one which is only using prior knowledge.

8.4. Multiple-factor analysis

Some features are introduced in Section 3-4 to constitute the observed space. Are they all or part used in the state prediction experiments? Which one is the most important in the perdition of web events?

Are they all independent with each other? What will happen if some of them are selected to do the prediction of state? In order to answer these questions, we will do the multiple-factor analysis (F-test) on the knowledge structural features and the statistical features respectively.

8.4.1. Statistical Features Analysis

Table 5: The Multi-factor analysis result of all features in observed space

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
degree	.001	1	.001	.001	.975
diameter	.112	1	.112	.115	.734
AverLen	.392	1	.392	.405	.525
Topology	.188	1	.188	.194	.660
density	.977	1	.977	1.010	.316
cluster	5.111	1	5.111	5.282	***.023***
page	1.357	1	1.357	1.402	.238
newword	.496	1	.496	.513	.475
word	.482	1	.482	.498	.481
diffuse	2.068	1	2.068	2.137	.145
content	.041	1	.041	.042	.837

At First, the influence to the state prediction model from each knowledge structural feature is analysed, as shown in Table 5. For the values of all Sig. of the knowledge structural features are bigger than 0.05, there is no remarkable influence to the prediction. After the single factor analysis, the compositions of them are analysed, which contains the two-order compositions. The <increased webpages, increased attributes> has the remarkable influence to the state prediction, because of its Sig. (0.037) < 0.05. When we do higher order composition of structural features, there is no remarkable one. So we believe that <increased webpages, increased attributes> is the simplest composition with the best prediction performance. We also do the real prediction experiment to confirm this. These two features are selected to do the state prediction task alone and the strategy 1, which contains all the structural features, is selected as the comparison. The accuracy is 66% and 69%, respectively.

8.4.2. Knowledge Structural Features Analysis

One-order factor analysis shows cluster has the remarkable influence to the state prediction due to its Sig. (0.023) < 0.05. It means that when only one feature selected to do the state prediction, cluster will have the best performance.

As for the compositions of them, for example, there are two compositions, <AverLen, Topology> and <degree, cluster>, which have the remarkable influences. Among the three-order compositions, <diameter, AverLen, density>, <Topology, density, cluster> and <degree, Topology, density> have the remarkable influences, shown in Table 6. In the higher order of compositions, only all the features together has the remarkable influence.

The same with structural features analysis, we also do the real state prediction experiment on these compositions to confirm their remarkable influence. The result shows that the accuracy of <AverLen, Topology> and <degree, cluster> are 68% and 66%, respectively. The accuracies of three-order compositions, <diameter, AverLen, density>, <Topology, density, cluster> and <degree, Topology, density>, are 64.5%, 71.5% and 66%. The overall accuracy of Strategy 2 is 73%.

From above analysis, it can be concluded that all the statistical features have the remarkable influence to the state prediction. And we can give the best compositions according this analysis if the number of features is limited, which can also get better performances.

Table 6: The Multi-factor analysis result of the three-order compositions of Structural features in observed space

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
AverLen * Topology * cluster	.056	1	.056	.071	.791
degree * AverLen * Topology	.494	1	.494	.626	.430
AverLen * Topology * density	1.327	1	1.327	1.681	.197
diameter * AverLen * Topology	.568	1	.568	.719	.398
degree * AverLen * cluster	.327	1	.327	.414	.521
AverLen * density * cluster	2.621	1	2.621	3.320	.070
diameter * AverLen * cluster	2.196	1	2.196	2.782	.097
degree * AverLen * density	1.390	1	1.390	1.761	.186
degree * diameter * AverLen	.093	1	.093	.118	.732
diameter * AverLen * density	3.119	1	3.119	3.950	***.049***
degree * Topology * cluster	.290	1	.290	.368	.545
Topology * density * cluster	3.275	1	3.275	4.148	***.043***
diameter * Topology * cluster	.062	1	.062	.079	.779
degree * Topology * density	4.496	1	4.496	5.695	***.018***
degree * diameter * Topology	1.425	1	1.425	1.805	.181
diameter * Topology * density	.002	1	.002	.002	.963
degree * density * cluster	2.071	1	2.071	2.624	.107
degree * diameter * cluster	.874	1	.874	1.108	.294
diameter * density * cluster	.848	1	.848	1.074	.302
degree * diameter * density	.205	1	.205	.260	.611

9 Conclusions and Future Work

In this paper, we have proposed a real time state prediction model for web events. The main contributions of this paper are summarized as follows.

1). The observed space is introduced, including two kinds of features: knowledge structural features and statistical features. They have their own ability to describe the real time statue of web events from different perspectives, which can guarantee the prediction model is more robustness and more accuracy.

2). State transition matrix and conditional dependency matrix are mined from the training web events to store their prior knowledge of state transitions and the conditional dependency, which can guide the state transition of web event effectively.

3). The prior knowledge, represented by state transition matrix and conditional dependency matrix, is embedded in HMM. The real time observed features are considered as the observed space of HMM. Through combining with the prior knowledge and the real time observed features into HMM, our model has good performance and robustness in the state prediction of web events.

4). We analyse what and how much features impact on the state prediction of web event by multi-factors analysis. By this analysis, we know which feature or composition of features can give the maximum performance with less observation cost. This gives us a way to balance the performance and the complexity on state prediction of web events.

The state of web events is determined by many factors. In this paper, we just consider the observed features and the prior knowledge. Overall accuracy is only about 80%. In the future, we will consider the categories of web events and the initial source of web events. We will also consider applying the results to e-learning [36].

Acknowledgements

This work was supported in part by the National Science Foundation of China under Grant nos. 91024012, 91324005 and the Key Innovation Program of Shanghai Municipal Education Commission under grant no.13ZZ064.

References

1. Dixon, W.J. and F.J. Massey Jr, *Introduction to statistical analysis.* 1957.
2. Valenzuela, O., et al., *Hybridization of intelligent techniques and ARIMA models for time series prediction.* Fuzzy Sets and Systems, 2008. **159**(7): p. 821-845.
3. Haykin, S.S., *Kalman filtering and neural networks*2001: Wiley Online Library.
4. de Oliveira, F.A., et al. *The use of artificial neural networks in the analysis and prediction of stock prices*. in *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*. 2011. IEEE.
5. Baum, L.E. and T. Petrie, *Statistical inference for probabilistic functions of finite state Markov chains.* The Annals of Mathematical Statistics, 1966. **37**(6): p. 1554-1563.
6. Rabiner, L.R., *A tutorial on hidden Markov models and selected applications in speech recognition.* Proceedings of the IEEE, 1989. **77**(2): p. 257-286.
7. Choi, H. and R.G. Baraniuk, *Multiscale image segmentation using wavelet-domain hidden Markov models.* Image Processing, IEEE Transactions on, 2001. **10**(9): p. 1309-1321.
8. Do, M.N. and M. Vetterli, *Rotation invariant texture characterization and retrieval using steerable wavelet-domain hidden Markov models.* Multimedia, IEEE Transactions on, 2002. **4**(4): p. 517-527.

9. Rossi, A. and G.M. Gallo, *Volatility estimation via hidden Markov models.* Journal of Empirical Finance, 2006. **13**(2): p. 203-230.

10. Mannila, H., H. Toivonen, and A. Inkeri Verkamo, *Discovery of frequent episodes in event sequences.* Data Mining and Knowledge Discovery, 1997. **1**(3): p. 259-289.

11. Perusich, K. and M.D. McNeese, *Using fuzzy cognitive maps for knowledge management in a conflict environment.* IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 2006. **36**(6): p. 810-821.

12. Allan, J., et al., *Topic detection and tracking pilot study: final report*, in *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*1998: Lansdowne, VA, USA. p. 194-218.

13. Allan, J., R. Papka, and V. Lavrenko, *On-line new event detection and tracking*, in *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*1998, ACM: Melbourne, Australia. p. 37-45.

14. Makkonen, J., *Investigations on event evolution in TDT*, in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: Proceedings of the HLT-NAACL 2003 student research workshop - Volume 3*2003, Association for Computational Linguistics: Edmonton, Canada. p. 43-48.

15. Golshani, M.A., A.M. Zarehbidoki, and V. Derhami, *Slash-based relevance propagation model for topic distillation.* Journal of Web Engineering, 2013. **12**(3-4): p. 265-290.

16. Liu, Y. and A. Agah, *Topical crawling on the web through local site-searches.* Journal of Web Engineering, 2013. **12**(3-4): p. 203-214.

17. Neumann, G. and S. Schmeier, *Interactive Topic Graph Extraction and Exploration of Web Content*, in *Multi-source, Multilingual Information Extraction and Summarization*, T. Poibeau, et al., Editors. 2013, Springer Berlin Heidelberg. p. 137-161.

18. Suhara, Y., et al., *Automatically generated spam detection based on sentence-level topic information*, in *Proceedings of the 22nd international conference on World Wide Web companion*2013, International World Wide Web Conferences Steering Committee: Rio de Janeiro, Brazil. p. 1157-1160.

19. Lin, T., et al., *The dual-sparse topic model: mining focused topics and focused terms in short text*, in *Proceedings of the 23rd international conference on World wide web*2014, International World Wide Web Conferences Steering Committee: Seoul, Korea. p. 539-550.

20. Guan, Q., et al., *Research and design of internet public opinion analysis system*, in *Proceedings of the 2009 IITA International Conference on Services Science, Management and Engineering*2009, IEEE Computer Society. p. 173-177.

21. Li, X., *The design and implementation of internet public opinion monitoring and analyzing system*, in *2nd International Conference e-Business and Information System Security (EBISS)*2010, IEEE: Wuhan, China. p. 1-5.

22. Quinn, C.J., T.P. Coleman, and N. Kiyavash, *A generalized prediction framework for granger causality*, in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*2011, IEEE. p. 906-911.

23. Radinsky, K. and E. Horvitz. *Mining the web to predict future events*. in *Proceedings of the 6th ACM International Conference on Web Search and Data Mining*. 2013. ACM.

24. Amodeo, G., R. Blanco, and U. Brefeld. *Hybrid models for future event prediction*. in *Proceedings of the 20th ACM international conference on Information and knowledge management*. 2011. ACM.

25. Joo, R., et al., *Hidden markov models: the best models for forager movements?* PloS one, 2013. **8**(8): p. e71246.

26. Cui, X., H. Jing, and C. Jen-Tzung, *Multi-view and multi-objective semi-supervised learning for HMM-based automatic speech recognition.* IEEE Transactions on Audio, Speech, and Language Processing, 2012. **20**(7): p. 1923-1935.

27. Chin-De, L., C. Yi-Nung, and C. Pau-Choo, *An interaction-embedded HMM framework for human behavior understanding: with nursing environments as examples.* IEEE Transactions on Information Technology in Biomedicine, 2010. **14**(5): p. 1236-1246.

28. Bharath, A. and S. Madhvanath, *HMM-based lexicon-driven and lexicon-free word recognition for online handwritten indic scripts.* IEEE Transactions onPattern Analysis and Machine Intelligence, 2012. **34**(4): p. 670-682.

29. Xuan, J., et al. *Building hierarchical keyword level association link networks for web events semantic analysis*. in *IEEE 9th International Conference on Dependable, Autonomic and Secure Computing (DASC)*. 2011. IEEE.

30. Luo, X., et al., *Building association link network for semantic link on web resources.* Automation Science and Engineering, IEEE Transactions on, 2011. **8**(3): p. 482-494.

31. Agrawal, R., T. Imieliński, and A. Swami. *Mining association rules between sets of items in large databases*. in *ACM SIGMOD Record*. 1993. ACM.

32. Jin, X., et al. *Topic initiator detection on the world wide web*. in *Proceedings of the 19th International Conference on World Wide Web*. 2010. ACM.

33. Deng, Y. and R. Lau, *On delay adjustment for dynamic load balancing in distributed virtual environments*, IEEE Trans. on Visualization and Computer Graphics, **18**(4):529-537, 2012.

34. Fan, J., X. Lin, X. Jia, and R. Lau, *Edge-pancyclickity of twisted cubes*, LNCS 3827, Springer, pp. 1090-1099, Dec. 2005.

35. To, D., R. Lau, and M. Green, *An adaptive multi-resolution method for progressive model transmission*, Presence, MIT Press, **10**(1):62-74, Feb. 2001.

36. Li, Q, R. Lau, E. Leung, F. Li, V. Lee, B. Wah, and H. Ashman, *Emerging Internet technologies for e-learning*, IEEE Internet Computing, **13**(4):11-17, July 2009.

37. Borzeshi, E.Z., Perez Concha, O., Xu, R.Y.D., Piccardi, M., Joint Action Segmentation and Classification by an Extended Hidden Markov Model, Signal Processing Letters, 2013. 20(12): p.1207-1210, IEEE.

38. Durrieu, J.-L.; Thiran, J.-P., Source/Filter Factorial Hidden Markov Model, With Application to Pitch and Formant Tracking, Audio, Speech, and Language Processing, IEEE Transactions on , 2013, 21(12): p.2541-2553, IEEE.