Mining Developing Trends of Dynamic Spatiotemporal Data Streams

Yu Meng and Margaret H. Dunham
Department of Computer Science and Engineering, Southern Methodist University, Dallas, USA
ymeng(mhd)@engr.smu.edu

Abstract—**This paper presents an efficient modeling technique for data streams in a dynamic spatiotemporal environment and its suitability for mining developing trends. The streaming data are modeled using a data structure that interleaves a semi-unsupervised clustering algorithm with a dynamic Markov chain. The granularity of the clusters is calibrated using global constraints inherent to the data streams. Novel operations are proposed for identifying developing trends. These operations include deleting obsolete events using a sliding window scheme and identifying emerging events based on a scoring scheme derived from the synopsis obtained from the modeling process. The proposed technique is incremental, scalable, adaptive, and suitable for online processing. Algorithm analysis and experiments demonstrate the efficiency and effectiveness of the proposed technique.**

Index Terms—**data mining, data stream, clustering, Markov chain, developing trend**

I. Introduction

Nowadays a growing number of applications generate streams of data. The data of this type include computer network monitoring data, highway traffic data, call detail records in telecomm industry, online purchase logs, credit card transaction records and data collected by other sensor networks [2]. Due to the rapidly growing demand, data stream has been an emerging research area in recent years [2, 9, 16].

A common characteristic of data stream is its high volume of data; moreover the data continuously arrive at a rapid rate. It is not feasible to store all data from the streams and take random accesses to the data as we do in traditional database. This implies a *single pass* restriction for all data in the streams. Therefore, the data stream must be modeled in order to obtain synopsis of global profile of the data. Data mining is a key technique in modeling stream data [2].

Another prominent characteristic of data is its time-variant nature. The profiles of the streams change over time and the change of the profiles may cause *concept drifts* [10, 17]. Concept drifts result in changes of responses to the queries by users. Knowing the developing trends provide valuable information about concept drifts. Note that by saying developing trends, we refer to the process that identifies the emerging events and obsolete events. A modeling technique of such is able to update the corresponding components of the synopsis without rebuilding the whole synopsis.

Mining of emerging patterns has recently received attention [8, 9, 10]. Given an ordered time series or a data stream that is composed of a (large) set of data points collected by a real-world application, we are interested in many cases in finding those events that are relatively new but potentially have significant impact on the system. The utility of identifying emerging events is obvious. Consider a managed highway network. Due to continuous municipal construction and population growth, the traffic volume distribution of the highway network changes gradually. New patterns keep emerging and some patterns disappear. The knowledge of the traffic distribution helps to predict demand, detect accidents, and provide usage analysis. Another example is traffic anomaly detection on computer networks. Traffic anomaly has been rated as a major indicator of system risk exposure. Juniper Networks has proposed a combination of traffic anomaly detection, protocol anomaly detection and stateful signatures to identify a variety of types of attacks in the computer networks [5]. Cisco has physically delivered the Cisco Traffic Anomaly Detector XT 5600 for detection of distributed denial of servers (DDoS), worms, and other attacks [6]. Applications can be intuitively extended to credit card fraud detection, web purchase mining, insurance risk modeling, and electric power demand management.

The significance of mining emerging events rests on detecting them dynamically at an early stage. Thus we aim at finding them when they are rare but new in occurrence in a soft real time manner. The rarity of emerging events makes it related to identifying patterns of rarity [7]. However previous work does not address this problem in a dynamic spatiotemporal environment. First, existing algorithms require that the entire dataset be accessed at one time [12, 13, 16] or mine within a data window [10, 11, 17]. Mining with the entire dataset

Part of this paper has been published in Proceedings of 10th Pacific-Asia Conference on Knowledge Discovery and Data Mining 2006, ©Springer LNAI Vol. 3918. This paper includes major modification and enhancement of the conference paper.

implicitly assumes stationarity and therefore losses the dynamically changing nature of the dataset. On the other hand, mining within a time window has made an assumption that the history prior to the window does not influence current behavior and is totally forgettable. The second issue is that existing algorithms either keep temporal information of the datasets without examining spatial relationship among data points [11, 12] or otherwise focus on spatial clustering but ignore temporal dependency of data [10, 13]. In the practical examples such as computer network traffic, highway traffic and electric power demand management, both the spatial relationship of data points and their temporal dependency are important.

Therefore previous related techniques can be viewed at three different levels. The first level work (outlier detection, anomaly detection, and rare event detection) is to detect those events which our deviate from the majority in the whole dataset. The second level work (surprising patterns, concept drifting) takes a time-variant statistical distribution of the data profile into consideration. The third level work (emerging events) seeks those events which are rare but with a support larger than a threshold. Moreover, mining of rarity can be either spatial or temporal or both. Our work represents this new fourth level.

The proposed technique is built based on the Extensible Markov model (EMM), a spatiotemporal modeling technique proposed by the authors [1]. EMM interleaves a clustering algorithm with a dynamic Markov chain to model spatiotemporal data. In this paper, modules for adding and deleting states of Markov chain are used in modeling. To extract emerging events, an *aging score of occurrences* is proposed to reflect decay of importance. Emerging events are judged using functions of the score and thus the proposed technique is able to continuously model the change of the data profile. The proposed technique inherits the traits of EMM and therefore is efficient, scalable, incremental and thus suitable for unsupervised online processing.

The rest of the paper is organized as follows. Section II discusses related researches. Sections III and IV present the EMM framework and the techniques to use it to mine developing trends. Section V provides performance experiments. We conclude the paper in Section VI.

II. Related work

A. Mining with Rarity

Mining emerging events is a problem of mining with rarity. Unsupervised techniques and supervised techniques are the two dominant categories in rare event mining [7].

Unsupervised techniques in rare event mining are capable of detecting events novel to the majority of the data distributions and are relevant to the target problem. These approaches are usually clustering-based. Clustering is based on measures of similarity or dissimilarity of the events of interests. The basic steps of distance-based rare event algorithms include feature construction, selection of distance measures, clustering of data objects to identify rare events. Clustering algorithms may be classified as partitioning (nearest neighbor) or hierarchical (BIRCH) [13]. The common unsupervised techniques include *distance-based* algorithms, *statistics-based* algorithms and *model-based* algorithms.

The mixture models [19, 21] and information theory [20] are used to build statistics based models that reflect the stochastic distributions of data. This type of algorithms is hard to estimate stochastic distributions of high dimensional data. Most statistics based algorithms assume that all dimensions are independent and loss dependency among different attributes.

Neural Networks and Support Vector Machines (SVM) are two of the popular non-linear model based algorithms [22, 23]. Generally to say the model-based algorithms require fewer parameters and computation is fairly fast. However the model based algorithms are sensitive to initial selection of the models. Because of number of parameters is predefined, they can be vulnerable to underfitting and overfitting.

Distance-based clustering does neither assume independence among different dimensions of data as statistics based algorithms do, nor is as sensitive to the initial selection of the model as model based algorithms do. The meaning of the model is easy to interpret and is suitable for spatial data mining.

B. Concept Drifting

In ordered time variant datasets, current behaviors do not rely on all historical data. Outdated data is considered less relevant. Keeping outdated data may negatively impact the ability of the model to identify emerging events and will certainly cause a loss of space and probably time efficiency. The data modeling approach used must capture the changing behavior of the data. Simple clustering techniques alone are not effective in these situations. This type of problem is called *concept drift* in machining learning. The common approaches include using sliding window, decay of importance and insertion of subtrees [9, 10, 11, 17].

C. Temporality

Modeling of temporality is another issue to be addressed in identifying emerging events. Markov chains and suffix trees have been used [11, 12] to store temporal profiles. The benefit of a Markov chain is its concise representation in mathematics. Some variants of the Markov chain with dynamic structures have been proposed for modeling dynamical changing data [1, 24]. The Suffix tree stores all suffixes of a sequence and is linearly efficient in string matching with the suffixes.

D. Extensible Markov Model

We have previously proposed the *Extensible Markov Model (EMM)* [1] and its applications in prediction [1], rare event detection [4] and risk assessment [3]. The technique combines clustering and temporal analysis into one framework. Each cluster represents a representative granular in the data space and is uniquely mapped to a state of a Markov chain and consequently a Markov chain

models the temporal profile of a dataset. Moreover the structure of the chain dynamically changes via increment and decrement operations which dynamically add and remove states. Thus EMM algorithms are able to capture the concept drift on the fly. As we will see in subsequent sections of this paper, EMM also provides a suitable technique to mine emerging events due to its scalability, incremental ability, adaptation, and online processing.

III. Modeling Data Streams Using Extensible Markov Model

EMM [1] models stream data with the following assumptions: Data are collected in discrete time points; Each of the data points is multidimensional and is represented by a vector as it represents measurements of a sensor network at certain time; Data has a good approximation of the Markov property. The raw data points use the format

$$M_t = < m_{1t}, m_{2t}, ..., m_{lt}, ...>,$$

Where m_{lt} denotes the measurement at sensor at location l at time t and M_t denotes a data point representing measurements at the time.

KDD has defined preprocessing procedures of data [14] to convert the format of raw data to the format that is appropriate for data mining. The preprocessed data for EMM use a format which combines the time stamp and spatial attributes in one vector:

$$V_t = <D_t, T_t, S_{1t}, S_{2t}, ..., S_{kt}, ...>,$$

where D_t denotes type of day, T_t time of the day, and S_{kt} the value of attributes found at a virtual spatial location k, at time t. Note that this virtual spatial location is a logical location which may be different from the physical location. This format defines an input real world event in the multidimensional data space.

At any time t, EMM consists of algorithms [1] including:

1) *EMMCluster* defines a technique for matching between input data at time $t + 1$ and existing states in the MC at time t. This is a clustering algorithm which determines if the new state should be added to an existing cluster (MC node) or whether a new cluster (MC node) should be created. In fact, any clustering technique could be used based on the structure on the input data.
2) *EMMBuild* updates (as well as adds and deletes) MC at time $t + 1$ given the MC at time t and output of EMMCluster at time $t + 1$. The insertion of new nodes to the MC is based on the similarity of the input data to the nodes in the MC. Deletion of nodes is triggered based on the target application. In the next section we propose a technique for deleting (and merging nodes) when the target application is emerging pattern detection.
3) Additional algorithms are needed based on the target application. In previous work we have discussed algorithms for prediction, anomaly detection and risk assessment. In the next sections we introduce algorithms for emerging and obsolete events.

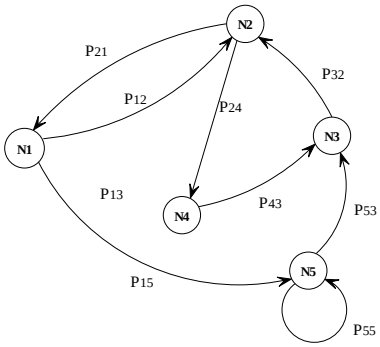


Figure 1. Sample EMM Graph

EMM can be viewed as a directed graph, as illustrated in Figure 1, similar to a Markov chain. Throughout this paper we assume that both the data input to the EMM algorithms as well as the labeling of each node in the EMM is viewed as a vector of n values. However, this is not a restriction of the EMM approach, rather it facilitates discussion. At any point in time, t, one node in the EMM graph is designated as the current node, N_c. This node is the one in the graph which is closest, based on some similarity measure, to the actual current data input. Each cluster (node or state in the EMM) is represented by a *cluster feature* [13], which includes at least a count of occurrence, CN_i (count on the node) and either a medoid or centroid for that cluster, LS_i. Figure 1 shows a simple EMM graph at a snapshot in time. Each node (or state of EMM, or cluster) and each link (or arc, or transition of Markov chain) in the graph are associated with cluster feature CF_i or CL_{ij} (count on the link). In Figure 1, the links are labeled with transition probabilities (rather than counts) which are obtained from the counts.

EMM algorithms can be accommodated into two major modules, EMM modeling and EMM pattern finding, as illustrated in Figure 2. The input stream data can be slightly preprocessed. Before modeling, hypothesis of the data profile must be input so that the modeling component can determine what modeling algorithms to use. EMM models the streams as the data are input and outputs synopsis incrementally. The EMM pattern finding algorithms are application specific and use the synopsis obtained in the modeling process to yield its results. As the user make a query, the application module outputs responses. This technique is flexible, modularized, and learns while performing applications.

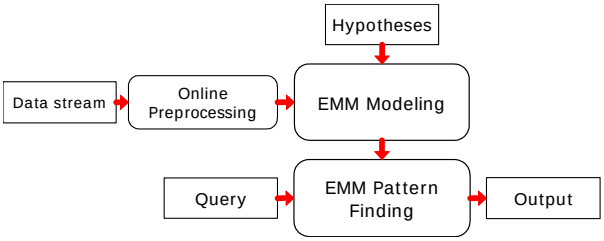


Figure 2. Work flow of EMM modeling and pattern finding

IV. MINING DEVELOPING TRENDS

In this section we present EMM techniques to be used in the identification of obsolete and emerging events. We first present the *aging score of occurrence*, investigate its properties and present how the score scheme is applied to identification of emerging events. Then we present how to identify obsolete events using a sliding window scheme.

As we have seen in Figure 1, an EMM is composed of two types of graph components: node and link. Each time a data point is input, a current node and consequently a current link are visited and updated by EMM. To reflect the decay of importance by time, we define the *aging scores of occurrence* (or simply the score) for all nodes and links. Each node or link is labeled with a score, in addition to previous introduced labeling elements: cluster features and count of links.

The score is built using an *indicator function*:

$$I_{Ek}(\xi_t) = \begin{cases} 1 & E_k = \xi_t \\ 0 & E_k \neq \xi_t \end{cases}$$

Here E_k is an EMM component and ξ_t is the current component (either node or link) of the same type at time t. We may eliminate the subscript k for simplicity.

With this indicator function, we present the definition of the aging score of occurrence in Definition 1. At each time instance only the current node and link are examined and updated rather than all graph components for the purpose of efficiency. Example 1 gives an example of calculation of the scores. An advantage of the score is its additive property as given in Lemma 1. This property is particularly useful when two EMM nodes are merged. The merge operation is needed, for example, to eliminate replications of clusters in BIRCH. Another important property of the score is its incrementality, as given in Lemma 2. With its incrementality, a score at a time instance can be derived from a score at a previous time instance and the data point at current time instance. The additivity and incrementality of the score gives efficiency of computations.

Definition 1: *(Aging Score of Occurrence for an EMM component)* At time *t*, the aging score of occurrence *for an EMM component E* is defined by

$$S_t^{(E)} = \sum_{i=1}^{t} (I_E(\xi_i) \cdot \alpha^{t-i}),$$

where $0<\alpha<1$ is an aging coefficient, t is current time.

Example 1: Assume that $\alpha = 0.9$ and that from time t=1 to t = 12 the current EMM nodes visited corresponding to a data series are:

$N_1\, N_3\, N_2\, N_1\, N_1\, N_1\, N_1\, N_2\, N_1\, N_3\, N_2\, N_2$

Then the scores for the current node and link at time $t=12$ are

$$S_{t=12}^{(N2)} = 0*0.9^{11} + 0*0.9^{10} + 1*0.9^{9} + 0*0.9^{8} + 0*0.9^{7} + 0*0.9^{6} + 0*0.9^{5} + 1*0.9^{4} + 0*0.9^{3} + 0*0.9^{2} + 1*0.9^{1} + 1*0.9^{0} = 2.9435$$

$$S_{t=12}^{(L22)} = 0*0.9^{11} + 0*0.9^{10} + 0*0.9^{9} + 0*0.9^{8} + 0*0.9^{7} + 0*0.9^{6} + 0*0.9^{5} + 0*0.9^{4} + 0*0.9^{3} + 0*0.9^{2} + 0*0.9^{1} + 1*0.9^{0} = 1$$

The scores of current nodes and links from t=1 to t=12 can be calculated as listed in Tables 1 and 2.

Lemma 1: Additivity of Score of Occurrence S_t. Let $S_t^{(E1)}$ and $S_t^{(E2)}$ be two scores where E_1 and E_2 are mutually exclusive events. When E_1 and E_2 are merged, i.e. $E = \{E_1, E_2\}$, the score of occurrence of the new EMM component is

$$S_t^{(E)} = S_t^{(E1)} + S_t^{(E2)}.$$

Proof: $S_t^{(E)} = \sum_{i=1}^{t} (I_E(i, \xi \in E) \cdot \alpha^{t-i})$

$$= \sum_{i=1}^{t} (I_{E1}(i, \xi_1 \in E_1) \cdot \alpha^{t-i}) + \sum_{i=1}^{t} (I_{E2}(i, \xi_2 \in E_2) \cdot \alpha^{t-i})$$

$$= S_t^{(E1)} + S_t^{(E2)}.$$

Lemma 2: Incrementality of Score of Occurrence S_t. Let $S_t^{(E)}$ and $S_{t-1}^{(E)}$ are the scores of the same EMM components at time t and $t-1$, respectively. Then

$$S_t^{(E)} = \alpha\, S_{t-1}^{(E)} + I_E(t, \xi_t).$$

Proof: $S_t^{(E)} = \sum_{i=1}^{t} (I_E(i, \xi \in E) \cdot \alpha^{t-i})$

$$= \alpha \cdot \sum_{i=1}^{t-1} (I_E(i, \xi \in E) \cdot \alpha^{(t-1)-i}) + I_E(t, \zeta)$$

$$= \alpha\, S_{t-1}^{(E)} + I_E(t, \xi_t).$$

To use the score of the EMM components, we extend the labeling of both with a score S_t and a time t, as defined in Definitions 2 and 3. It is easy to see that the CF_t, $CL_{ij,\,t}$, and S_t are additive and incremental, and thus computations of e-CF and e-CL are efficient.

Definition 2: Extended Cluster Feature (e-CF) is a vector with five attributes to summarize the information of a cluster or a node at time t, defined by:

$$e\text{-}CF_t = < CN_t, \vec{LS}_t, SS_t, S_t^N, t >.$$

Definition 3: Extended EMM Transition Labeling e-CL is defined by:

$$e\text{-}CL_{ij,t} = < CL_{ij,t}, S_t^L, t >.$$

We use a sliding window to achieve the stationary approximation. The idea is to examine whether a node of EMM has been visited in the window w. If a node is visited, then it is in active use; otherwise the node is considered an obsolete node and will drop from EMM along with associated links. Note that when a node is in active use, all of their associated links may not have occurred in the target window. This is a difference between our algorithm and previous work. We keep transitions which themselves may not have occurred within the window. Keeping these transitions allows us

to capture a more accurate temporal profile of the active states. Obsolete events and emerging events are given in the Definitions 4 and 5. The emerging events can be interpreted as data inputs that correspond to newly occurring nodes and links at the time of examination. With both nodes and transitions labeled using the Definitions 2 and 3, obsolete events and emerging events can be detected on the fly using the score scheme. Our approach to determining emerging patterns based on scoring has the following benefits:

- The Score is additive and incremental. The computations upon the score are efficient.
- Obsolete events are deleted to achieve an approximate stationarity of temporality.
- Emerging events are examined not only by the distribution of input data points, but also by first order of dependency of the input data points.
- By capturing 0^{th} and 1^{st} order of dependency of dataset, the emerging patterns of any kind is left for offline or distributed work, based on user's need.

Scores of a node and a link are computed at one time. In addition, two comparisons are needed to determine an emerging event. Thus for an EMM with *m* nodes, the time for these computations has *O(1)* complexity. Recall that in EMM, the clustering takes *O(m)* for nearest neighbor algorithm and *O(log m)* for BIRCH clustering, and *O(1)* for updating the Markov chain.

Definition 4: Obsolete Events. Assume current time is *t*. If an EMM node N_o is not seen in a window *[t-w, t]*, it is recognized as an obsolete node. All transitions (links) in and out of obsolete node N_o are obsolete and are removed from EMM.

Definition 5: Emerging Event. Assume a transition L_{ij} between two EMM nodes N_i and N_j occurs at time *t*. The transition L_{ij} and the absorbing node N_j is considered as an emerging events if

$$R_t^{(L)} = S_t^{(L)} / CL_t > 1 - \varepsilon, \text{ or}$$

$$R_t^{(N)} = S_t^{(N)} / CN_t > 1 - \varepsilon,$$

where ε is a predefined threshold.

Example 2: Assume in Example 1 $\varepsilon = 0.1$ and window w = 10. The score/count ratio at time t=12 would be

$$R_{12}^{(L22)} = 1/1 = 1$$

$$R_{12}^{(N2)} = 2.9435/4 = 0.7359$$

For simplicity we can call $R_t^{(N)}$ the S/C ratio. The S/C ratios at other time points are listed in Tables 1 and 2. Intuitively the criteria in Definition 5 gives a flag when the link and the node in EMM associated with the current input data point are both rare and new. Either rare or new does not count to be emerging. We really see this in calculation results given in Tables 1 and 2.

Table 1. Calculations of scores and S/C ratios for nodes

t	N_1 /$R^{(1)}$	N_2 /$R^{(2)}$	N_3 /$R^{(3)}$
1	1.000/1.000	-	-
2	-		1.0000/1.0000
3	-	1.0000/1.000	-
4	1.7290/0.8645	-	-
5	2.5561/0.8520	-	-
6	3.3005/0.8251	-	-
7	3.9704/0.7941	-	-
8	-	1.5905/0.7953	-
9	4.2161/0.7027	-	-
10	-	-	1.4305/0.7153
11	-	2.1595/0.7198	-
12	-	2.9435/0.7359	-

Table 2. Calculation of scores and S/C ratios for links

t	L_{11} /$R^{(11)}$	L_{12} /$R^{(12)}$	L_{13} /$R^{(13)}$	L_{21} /$R^{(21)}$	L_{22} /$R^{(22)}$	L_{32} /$R^{(32)}$
1	-	-	-	-	-	-
2	-	-	1.0000 /1.0000	-	-	-
3	-	-	-	-	-	1.0000 /1.0000
4	-	-	-	1.0000 /1.0000	-	-
5	1.0000 /1.0000	-	-	-	-	-
6	1.9000 /0.9250	-	-	-	-	-
7	2.7100 /0.9033	-	-	-	-	-
8	-	1.0000 /1.0000	-	-	-	-
9	-	-	-	1.5905 /0.7953	-	-
10	-		1.4305 /0.7153	-	-	-
11	-	-	-	-	-	1.4305 /0.7153
12	-	-	-	-	1.0000 /1.0000	-

V. EXPERIMENTS

In this section, we examine the efficiency in terms of time and space, and then effectiveness of the proposed technique.

A. Data and Experiment Setup

Experiments were performed on VoIP traffic data provided by Cisco Systems and represents logged VoIP CDR traffic in their Richardson, Texas facility from Mon Sep 22 12:17:32 2003 to Mon Nov 17 11:29:11 2003. Over 1.5 million call trials were logged. The logs contain 66 attributes. After data preprocessing, the size of dataset is 5422 and the format used is:

$$V_t = <D_t, T_t, S_{1t}, S_{2t}, ...>,$$

where D_t denotes type of day, T_t time of the day, and S_{it} the value of statistic volume found at that link *i*, at time *t*. In the experiments, traffic from and to the internal site is selected for investigations. Specifically the site related traffic comprises the links of internal to internal (Cisco to Cisco), internal to local (Cisco to outside Cisco but a local call), internal to national (Cisco to long distance but within US), internal to international (Cisco to outside US). as well as those in opposite directions. Combined

with the day of the week and the time of the day, there are 9 attributes in each data point. BIRCH and nearest neighbor clustering algorithms are used, Euclidean distance is used as the similarity measure, and each MC node is labeled using a medoid or centroid. Four parameters, namely clustering threshold *th*, window size *w*, decay coefficient α and threshold of score/count ratio *r*, are used throughout the experiments. A default value is used (*th = 30*, *w = 1000*, α *=0.01* and *r = 0.9)* unless it is a variable of investigation.

B. Calibration of Clustering Threshold

The EMM paper [1] indicates it possible to calibrate an empirically optimal threshold value. This threshold value is estimated by testing with a range of threshold values against the EMM prediction algorithm [1]. The optimal value of the threshold is picked at the place where the best performance of prediction is found. Figure 3 illustrates that the empirically optimal threshold has a value of 35. A threshold of **30** is selected throughout the experiments.

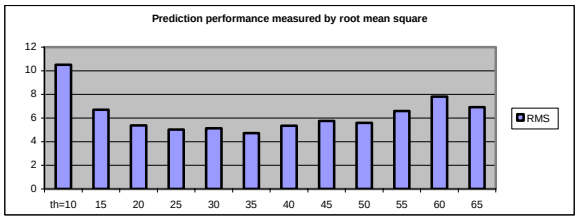


Figure 3: Determination of an empirical threshold value

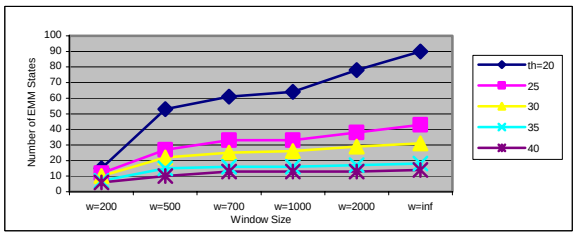


Figure 4. Number of EMM states created vs. window size and clustering threshold

C. Memory Efficiency

The growth rate of EMM nodes is the ratio of the number of EMM nodes to the size of the dataset. The growth rate is a measure of memory efficiency. Two variables, clustering threshold *th* and window size *w*, are used to examine the memory efficiency of the model. Figure 4 shows that more memory is needed when the clustering threshold is selected to be tighter or when window size is selected to be wider. For the selected threshold 30, the growth rate is less than 1% for all window sizes. A plateau is seen in window size ranges from 700 to 1000. This is particularly interesting and it reflects the weekly period of the traffic statistics.

D. Time Efficiency

Although the effectiveness of EMM is independent of clustering algorithm selected, the time efficiency of it is determined by clustering, Markov chain, and operations in mining emerging events. Clustering has approximately a *log(m)* time complexity in hierarchical BIRCH algorithm and *O(m)* for nearest neighbor clustering algorithm. Update on Markov chain is *O(1)*. Score computation and operations on the scores all have *O(1)* complexity. Figure 5 shows that the number of deletions dramatically changes with different window sizes. However the total run time of the application at hand does not change much as illustrated in Figure 6. Taking a division of total run time over size of the dataset, we see that the execution takes 1ms for each data point.

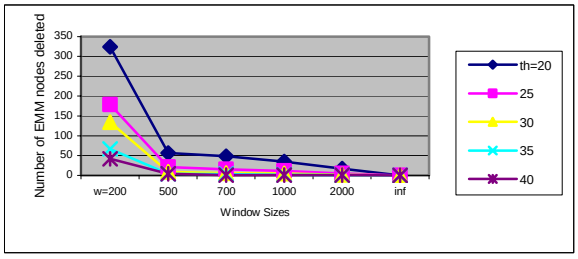


Figure 5. Number of EMM Nodes deleted

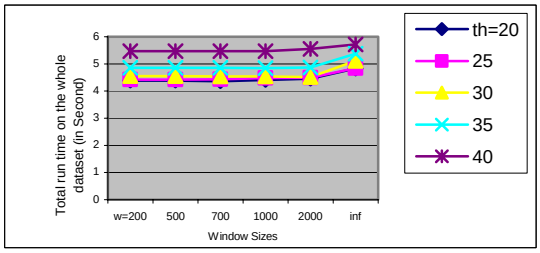


Figure 6. Run time to process the dataset

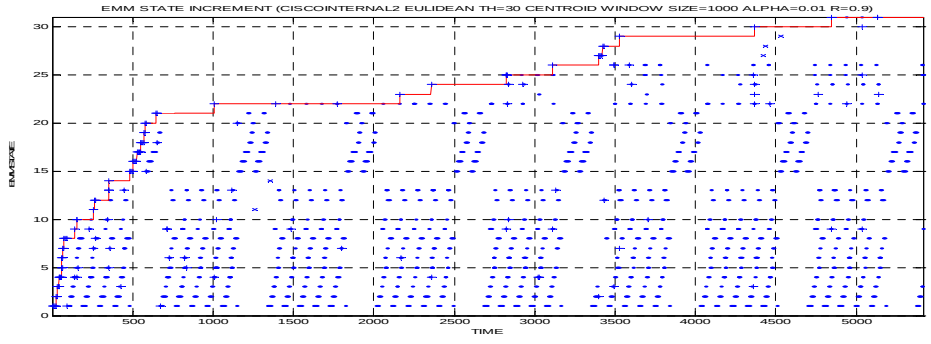


Figure 7. Node increment, node deletion, and emerging event finding with EMM

E. Effectiveness

The effectiveness of the proposed technique lies on the capability to find input data points corresponding to emerging nodes and links in EMM. The technique should be able to model the data, remove obsolete nodes and links, and finding emerging events as designed. Figure 7 illustrate increments of nodes, deletions of obsolete nodes (x) and detections of emerging nodes (+) in EMM as the time goes. Analysis is needed to interpret the meaning of emerging events. Figure 8 gives two day traffic data on Oct. 27 and 28, 2003 and those corresponding to emerging events. It was able to detect a shutdown of the system when an absence of any call logs in Cisco dataset occurred on Monday, October 27, 2003. Initially when the system was down, our technique was able to set alarms on the inputs corresponding to emerging events. After a while, it got used to it and shut down the alarms until it detected new emerging events were found. We can see the alarms cover most of the shutdown period. Also in Figure 7, the technique was able to forget the states and links that were judged to be emerging with 'x' signs.

F. Effectiveness and Parameters

Now that we have briefly examined the effectiveness of EMM, we evaluate the impacts of clustering threshold, window size, decay coefficient, and score/count ratio threshold on number of emerging events detected. This investigation helps to give parameter selection criteria.

For each of the parameters as the variable, the number of emerging events determined by score/count ratio of nodes is denoted by *SN_CN*. In the same way, the number determined by score/count ratio of links is demoted by *SL_CL*. The number of emerging events determined by OR relationship of the two is denoted by *OR*. The results are illustrated in Figures 9-12. First the emerging events are detected by link criteria more than node criteria. This shows the importance of detection criteria using nodes and links, rather than the nodes alone. Secondly the number of emerging events detected has a low slope plateau with respect to the window size, decay coefficient, and threshold of score/count ratio. We feel the selection of parameters on the plateau results in a good capture of the profile characteristics.

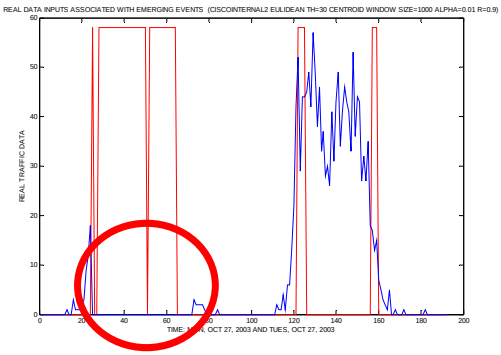


Figure 8. Identifying emerging events within real data

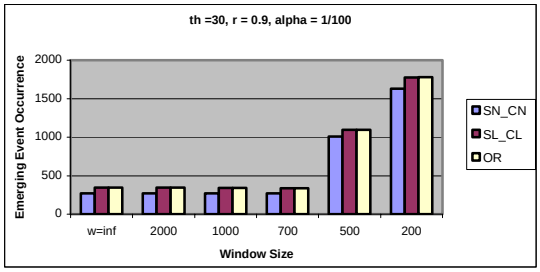


Figure 9. Emerging events vs. window size

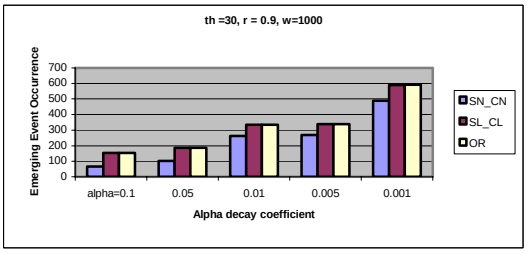
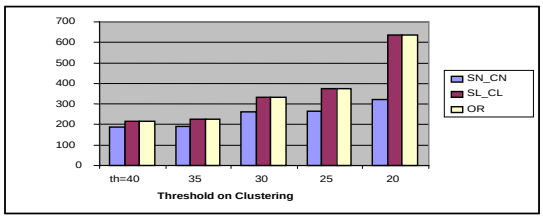


Figure 10. Emerging events vs. decay coefficient alpha.



w =1000, r = 0.9, alpha = 1/100

Figure 11: Emerging events vs. clustering threshold

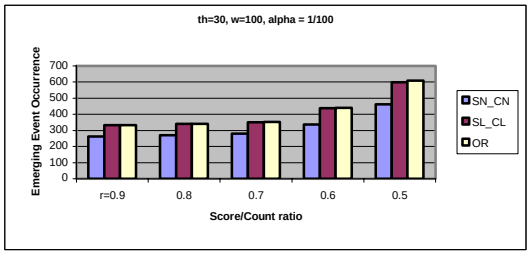


Figure 12: Emerging events vs. score/count ratio

G. Summary

In terms of efficiency and effectiveness, we can conclude the following from the experiments.

- An algorithm is able to delete obsolete events and finding emerging events as expected.
- The memory requirement with this model is efficient. The state growth rate of EMM is less than 1% with an optimal clustering threshold determined by EMM.
- All operations introduced for emerging event detection require an *O(1)* time. Experiments show deletions do not obviously affect total run time needed.
- Examination of both spatial and temporal relationships is important. Temporal dependency gives more conservative judgments.

Plateaus exist for parameters such as window size, score/count ratio and decay coefficient, with which appropriate ranges of parameters that reflect the dynamic profile of the data can be determined.

VI. CONCLUSIONS

In this paper, we have presented a data mining modeling technique which is suitable for finding emerging events in a spatiotemporal environment. The model accommodates anomaly, concept drifting, support, and temporality in one framework without losing time and space efficiency. Novel operations are proposed for deleting obsolete states, and finding emerging events based on a scoring scheme. All computations introduced in this paper are with *O(1)* time complexity. Selection of parameters for appropriate capturing the dynamic data profile is found to have a range. Offline mining of the results generated by this model is possible for more complex patterns.

ACKNOWLEDGMENT

The authors gratefully acknowledge Cisco System for providing the training and testing data, and Dr. Fei Wu at Nortel Networks and Dr. Marco Marchetti at Southern Methodist University for his helpful information. This research is supported by NSF Grant IIS-0208741.

REFERENCES

[1] Margaret Dunham, Yu Meng, and Jie Huang, “Extensible Markov Model”, ICDM, 2004, pp. 371-374.
[2]. Haixun Wang, Jian Pei, Philip S. Yu: Online Mining of Data Streams: Applications, Techniques and Progress. *ICDE* 2005.
[3]. Yu Meng and Margaret H. Dunham, “Online Mining of Risk Level of Traffic Anomalies with User's Feedbacks”, in *Proceedings of the Second IEEE International Conference on Granular Computing (GrC'06)*, Atlanta, GA, May 10-12, 2006.
[4]. Y. Meng, M.H. Dunham, F.M. Marchetti, and J. Huang, ”Rare Event Detection in A Spatiotemporal Environment”, in *Proceedings of the Second IEEE International Conference on Granular Computing (GrC'06)*, Atlanta, GA, May 10-12, 2006.
[5] Sarah Sorensen, “Competitive Overview of Statistical Anomaly Detection”, *Intrusion Prevention Literature*, Juniper Networks, 2004.
[6] Cisco Systems, *Cisco Traffic Anomaly Detection and Mitigation Solutions*, Product Bulletin, 2004
[7]. Aleksandar Lazarevic et al., “Tutorial: Data Mining for Analysis of Rare Events: A Case Study in Security, Financial and Medical Applications”, PAKDD, 2004.
[8]. G. Dong and J. Li, “Efficient mining of emerging patterns: Discovering trends and differences”, SIGKDD 1999: 43-52.
[9]. Aggarwal, Han et al., "A Framework for Clustering Evolving Data Streams", VLDB 2003.
[10]. Kevin B. Pratt and Gleb Tschapek, "Visualizing Concept Drift", SIGKDD 2003.
[11]. N. Ye and X. Li, “A markov chain model of temporal behavior for anomaly detection”, *Proc. IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop*, 2000, pp.171-174.
[12]. E. Keogh et al., “Finding Surprising Patterns in a Time Series Database in Linear Time and Space”, SIGKDD, 2002, pp. 550-556.
[13]. T. Zhang, R. Ramakrishnan, and M. Livny, “BIRCH: A new data clustering algorithm and its applications”, *Data Mining and Knowledge Discovery*, 1(2): 141-182, 1997.
[14]. Usama M. Fayyad et al., “Knowledge discovery and data mining: A Overview”, *Advances in Knowledge Discovery and Data Mining*, pages 1-34, AAI/MIT Press, 1996.
[15]. M. Mahoney and P. Chan, “Learning Non-stationary Models of Normal Network Traffic for Detecting Novel Attacks”, SIGKDD 2002, pp. 376-385.
[16]. P. Domingos and G. Hulten, "Mining high-speed data streams”, *Knowledge Discovery and Data Mining*, pages 71-80, 2000.
[17]. Gerhard Widmer and Miroslav Kubat, “Learning in the Presence of Concept Drift and Hidden Contexts”, *Machine Learning*, 23, 69-101 (1996).
[18]. J. Chen et al., “Temporal Sequence Associations for Rare Events”, PAKDD, 2004.
[19]. K. Yamanishi et al., “On-line Unsupervised Outlier Detection Using Finite Mixtures with Discounting Learning Algorithm”, SIGKDD, 2000, pp. 320–324.
[20]. W. Lee and D. Xiang, “Information-Theoretic Measures for Anomaly Detection”, *Proc. IEEE Symp. Security*, 2001, pp. 130-143.
[21]. E. Eskin, “Anomaly Detection over Noisy Data using Learned Probability Distributions”, ICML 2000, pp. 255-262.
[22]. Simon Hawkins et al., “Outlier Detection Using Replicator Neural Networks”, DaWaK, 2002, pp. 170–180.
[23]. Aleksander Lazarevic et al., “A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection”, *Proc. SIAM Int’l Conf. Data Mining*, 2003.
[24]. Dani Goldberg, Maja J Mataric. “Coordinating mobile robot group behavior using a model of interaction dynamics,” Proc. *3rd Int’l Conf. Autonomous Agents*, 1999.

Yu Meng received an MS degree from Jilin University, China in 1990 and his MSEE and MSCS from Louisiana State University in New Orleans in 1998 and 2000 respectively. He has 10 year professional experience including his services in Nortel Networks. He is currently a Ph.D. candidate at Southern Methodist University, affiliated with the Databases Lab. His research interests include data mining, stream data modeling, VoIP, network security, intelligent Web, and mobile databases.

Margaret H. Dunham received the BA and MS degrees in mathematics from Miami University, Oxford, Ohio, and the PhD degree in computer science from Southern Methodist University in 1970, 1972, and 1984, respectively. She is currently a professor in the Department of Computer Science and Engineering at Southern Methodist University. Professor Dunham’s research interests encompass data mining, mobile computing, and bioinformatics. Dr. Dunham served as an editor of the ACM SIGMOD Record from 1986 to 1988. She has served on the program and organizing committees for many ACM and IEEE conferences. She served as guest editor for a special section of *IEEE Transactions on Knowledge and Data Engineering* devoted to Main Memory Databases as well as a special issue of the ACM SIGMOD Record devoted to Mobile Computing in databases. She served as the general conference chair for the ACM SIGMOD/PODS held in Dallas in May 2000 and was an associate editor for *IEEE Transactions on Knowledge and Data Engineering*. She has published over eighty technical papers.