Rare Event Detection in a Spatiotemporal Environment

Yu Meng, Margaret H. Dunham, F. Marco Marchetti and Jie Huang

Abstract—**In this paper we explore the use of Extensible Markov Models (EMM) to detect rare events in a spatiotemporal environment. This initial work shows that an EMM is scalable, dynamic, and can detect rare events based on spatial values, temporal values, or transitions between real world states. The core of the EMM approach is a combination of clustering and dynamic Markov Chain.**

Keywords—**Extensible Markov Model (EMM), rare event detection, spatiotemporal data.**

1. INTRODUCTION

THE target of our research is the modeling of spatiotemporal events. In this complex environment, sensors at remote locations collect data at predefined time intervals at each location. An event can be viewed as the collection of all sensor values at a specific point in time. Spatiotemporal applications of this type include weather and geographic data, collections of data from satellites, and data obtained when monitoring the health of a patient. As the sensor data continues to arrive, any associated modeling technique should be constructed dynamically and change as the data is received. Markov Chain (MC) has been extensively used in many applications [5, 6]. However, the static nature of MCs does not fit into this dynamic environment. In this paper we examine the use of our *Extensible Markov Model (EMM)* technique as a modeling tool for these complex spatiotemporal environments. Although EMMs are suitable for many different applications, in this paper we examine their use for rare event detection.

Manuscript received December 29, 2005. This material is based upon work supported by the National Science Foundation under Grant No. IIS-0208741.

Yu Meng is with the Department of Computer Science and Engineering, School of Engineering, Southern Methodist University, Dallas, TX 75275-0122 USA (e-mail: ymeng@engr.smu.edu)

Margaret H. Dunham is with the Department of Computer Science and Engineering, School of Engineering, Southern Methodist University, Dallas, TX 75275-0122 USA (phone: 214-768-3087; fax: 214-768-3085; e-mail: mhd@engr.smu.edu)

F. Marco Marchetti is with the Department of Computer Science and Engineering, School of Engineering, Southern Methodist University, Dallas, TX 75275-0122 USA (e-mail: marco@engr.smu.edu)

Jie Huang was with Southern Methodist University, Dallas, TX 75275 USA. She is now with the University of Texas Southwestern Medical Center at Dallas, Dallas, TX 75390-9041 USA (e-mail: jie.huang@utsouthwestern.edu).

The Markov chain applies a stochastic approach which models the randomness of nature [5]. Definition 1 defines a Markov Chain. For simplicity, here we visualize the MC as a weighted directed graph. The graph has a fixed structure although the labels of the arcs may change. Note that in this paper we will use arc, link and transition interchangeably, and node and state interchangeably. The Markov Property states that the next transition depends only on the current state regardless of the past history, which implies that $P_{12\ldots j} = P(N_1)P(N_2 \mid N_1) \ldots P(N_j \mid N_{j-1})$.

Definition 1: A Markov Chain (MC) is a graph with m vertices or states, S, and directed arcs, A, such that:
1) $S = \{N_1, N_2, \ldots, N_m\}$, and
2) $A = \{L_{ij} \mid i\in 1, 2, \ldots, m, j\in 1, 2, \ldots, m\}$ and each arc, $L_{ij} = \langle N_i, N_j \rangle$ is labeled with a transition probability $P_{ij} = P(N_j \mid N_i)$.

An MC is used to model the transition of real world events. Each state in the graph corresponds to a real world state. An MC can be created by two steps: determine the states in the model and determine the suitable state transition probabilities. Usually the structure is defined by a domain expert to reflect the classification of real world observations and events. Transition probabilities may be determined by learning or assignment from a domain expert. Learning occurs by observing and counting how many times each transition in the model has been taken versus how many times the state actually occurred in the training data. A major problem associated with this straightforward approach to MC creation is that the exact structure of the graph is not known in advance.

We have previously introduced the Extensible Markov Model to overcome the static nature of the MC [1]. An EMM is a time varying MC. At any point in time, when viewed as a static graph it is an MC. However, over time the structure of the graph changes. Both the number of nodes in the graph as well as the transition probabilities vary. Definition 2 provides the definition of an Extensible MC as found in our earlier work [1].

Definition 2: Extensible Markov Model (EMM): At any time t-1, EMM consists of a Markov Chain (MC) with designated current node, N_c, and algorithms to modify it, where algorithms include:
1) *EMMCluster,* which defines a technique for matching between input data at time t and existing states in the MC at

time t-1. Each cluster is a representative granule in the continuous data space.
2) *EMMIncrement* algorithm, which updates MC at time t given the MC at time t-1 and clustering result at time t.
3) *EMMDecrement* algorithm, which removes nodes from the EMM when needed.

EMM interleaves a clustering algorithm with a Markov chain for spatiotemporal data processing. This research continues our study of the EMM model and applies it to the rare event problem. At any point in time, t, one node in the graph is designated as the current node, N_c. This node is the one in the graph which is closest to the actual current input as represented by a vector of n values. We assume that some similarity measure has been defined to determine the similarity between any input state vector and node in the graph. In addition, a minimum similarity threshold value is used to determine when to create a new node. In fact, any similarity or distance technique could be used. In our research we have investigated Jaccard, Cosine, and Overlap similarity measures and Euclidean distance [4]. With each node in the EMM represented as a vector of numeric values and each input state likewise being represented by a vector, a measure such as Cosine could be used. Each cluster (node in the EMM) is represented by either a medoid or centroid for that cluster. The medoid is a representative input event into that cluster, while the centroid is the vector created by finding the mean of all input vector values for each sensor input. Figure 1 shows a simple EMM graph at a snap shot in time.

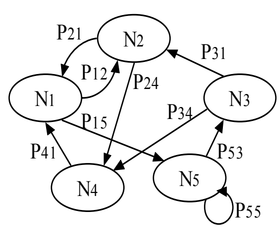

Fig. 1. Sample EMM Graph

The EMM can be used to model many different types of spatiotemporal environments and can be applied to many different spatiotemporal applications. We have previously examined its use for prediction [2] and emerging pattern analysis [3]. In this paper we target rare event detection.

In the remainder of the paper we first examine previous work on rare event detection, we then provide an overview of how EMMs can be used to detect rare events, the results of experiments examining the behavior of EMMs in a spatiotemporal environment are then presented, and finally we conclude the paper.

2. Rare Event Detection

In this section we provide a brief overview or previous work in the area of rare event detection. We conclude with an analysis of requirements for rare event detection in the targeted spatiotemporal environment.

Unsupervised techniques and supervised techniques are the two dominant categories in rare event mining [7]. Supervised techniques are classification-based. This type of approach consists of a machine-learning algorithm being trained over pre-classified data. With data labeled as normal or rare classes, supervised algorithms aim at achieving high recall or precision or both. A nature of rare class problems is the imbalance of data in normal classes and rare classes. A common solution is to use different sampling schemes to alter data distribution so that the data imbalance is alleviated in training data. Over-sampling [8] and under-sampling [9] are the basic methods to balance the classes. SMOTE [10], an optimized combination of both over-sampling and under-sampling, is used for outperforming the above techniques. Boosting techniques [11] are applied to promote performance by optimizing a weighted classifier derived from multiple classifiers. Scoring or cost sensitive algorithms [12, 13, 14, 15] further the endeavor to minimize cost rather than minimize errors.

Unsupervised techniques in rare event mining are capable of detecting events novel to existing data distributions. This type of approaches is clustering-based. The data are not labeled a prior. Clustering is based on measures of similarity or dissimilarity (distance) of the events of interests. The common unsupervised techniques include *distance-based* algorithms, *statistics-based* algorithms and *model-based* algorithms [7].

To function well in a spatiotemporal environment where sensor values arrive in a never ending stream of data, most previous techniques will not function well. We view the following list as a minimum list of requirements of any rare event detection approach in a spatiotemporal environment:

- Supervised and unsupervised: In a target environment domain experts will probably have some a priori ideas concerning what are rare events. These should be supported. However, this is not enough. As the environment changes and even experts can not foresee all rare events that might occur, unsupervised techniques which learn what anomalous behavior is should also be supported.
- Scalability: In this environment, the number of input events is infinite. Thus any modeling technique must be able to learn what normal behavior is, but do so in an efficient manner. This includes the ability to learn developing trend and forget old information as needed.
- Spatial, Temporal, and Transition: Rare events are determined not only by the spatial features (as represented in the vector of sensor values) but also by the temporal aspect. In addition, transition behavior is important.
- Quasi-Real Time: As the stream of sensor data continues to arrive, the identification of rare even must be handled dynamically as it comes.

None of the previously proposed approaches satisfy all of these. The EMM approach, however, does.

3. Use Of EMMs For Rare Event Detection

Extensible Markov Models are well suited to model the spatiotemporal environment and to detect rare events. They support both supervised and unsupervised detection. Even though the basic EMM algorithms assume that learning is used to identify rare events in an unsupervised manner, pre-designated nodes can be added to the EMM which are known to be target events to look for. Scalability is achieved due to the fact that similar real world events are clustered into one EMM node. In addition, nodes can be removed from the EMM or nodes may be merged together if desired. We do not discuss either of these in this paper, but have discussed them in other publications [1, 3]. EMMs can be used to identify events that are rare based on the events themselves (space), time of the events (temporal), or unusual transitions. Finally, the EMM rare event detection algorithm works dynamically in a quasi-real time manner as the data arrives. The time required for each execution of the rare event detection algorithm is dominated by the clustering algorithm which in turn depends on the number of EMM states (not the number of real world events).

EMMs can predict (detect) rare events when a captured real world event is not close enough to an existing node in the graph or when the transition probability to the closest node from the previous node is low. Thus the current captured event has not occurred frequently in the past or has not occurred following the previous state often. We have examined the use of EMMs for rare event detection using network VoIP traffic data as well as automobile traffic data. In these large environments, EMM achieves scalability through a distributed hierarchical approach. Anomalies in Web traffic can be examined in a hierarchically distributed fashion:

- Detect anomalous activity at a specific site (switch).
- Collect rare event information at a higher level by constructing an EMM of more global anomalies from several sites.

To use EMMs for rare event detection, an additional algorithm is needed. The *EMMRare* algorithm indicates if the current input state (event) is rare. A rare event is detected if an input event, E_t, is determined not to belong to any existing cluster (node in EMM), if the cardinality of the associated cluster (CN_n) is small, or if the transition from the current node to the new node is small. Instead of using the actual cluster cardinality, we look at the *Occurrence Frequency* (OF_c) of a node N_c as defined by:

$$OF_c = CN_c \Big/ \sum_i CN_i$$

Likewise when determining what is meant by small for a transition probability, we should look at a normalized rather than actual value. We, thus, define the *Normalized Transition Probability* (NTP_{mn}), from one state, N_m, to another, N_n, as:

$$NTP_{mn} = CL_{mn} \Big/ \sum_i CN_i$$

To identify what we mean by small, two thresholds are introduced. We use th_{of} to denote threshold for occurrence frequency and th_{tp} to denote threshold for normalized transition probability. The EMMRare algorithm is shown below. Here we assume that the starting state is a nonempty EMM.

Algorithm EMMRare:

Input: $V_t = \langle S_1, S_2, \ldots, S_n \rangle$: Observed values at n different locations at time t.

G: EMM with m states at time t-1.

N_c: Current state at time t-1.

th_{of} ; Threshold for occurrence frequency

th_{tp}: Threshold for transition frequency

Output:

R: Boolean value indicating as to whether this event is rare or not.

N_c: Current state at time t.

G: New EMM at time t, reflecting input of event at time t-1.

Algorithm:

1. foreach N_i in G determine sim(V_t, N_i);
2. let N_n be node with largest similarity value;
3. let sim be this largest similarity value;
4. if sim >= threshold then //update matching state
5. $CN_c = CN_c + 1$;
6. if L_{cn} exists
7. $CL_{cn} = CL_{cn} + 1$;
8. else // new node created
9. create new transition $L_{cn} = \langle N_c, N_n \rangle$;
10. $CL_{cn} = 1$;
11. endif;
12. $N_c = N_n$;
13. if $OF_c > th_{of}$ or $NTP_{mn} > th_{tp}$ or if N_c is a designated node then
14. R = true // Rare Event
15. else
16. R = false ; // Not a Rare Event
17. else // create a new state N_{m+1} represented by V_t
18. create new node N_{m+1};
19. $N_{m+1} = V_t$;
20. $CN_{m+1} = 0$;
21. create new transition $L_{c(m+1)} = \langle N_c, N_{m+1} \rangle$;
22. $CL_{c(m+1)} = 1$;
23. $CN_c = CN_c + 1$;
24. $N_c = N_{m+1}$;
25. R = True ; // Rare Event
26. endif;
27. endif;
28. R = R or R_{Nc};
29. R_{Nc} = R.

Notice that in the algorithm, we predict a rare event if a new node is created (step 25) or if either of the thresholds is met (step 14). This is where the unsupervised portion of EMMRare exists. After this, the state is even identified as rare if the node (cluster) to which the input event is place as been previously designated as a targeted event to flag (step 28). This is where the supervised portion of EMMRare exists. Note R_{Nc} is the Boolean indication of whether the current node has been labeled to be rare, and this label may need confirming or modifying by a domain expert. Think of EMMRare as raising a "red flag". Something seems to be wrong, but only a human expert can actually tell if it is. EMMRare actually contains the EMMCluster and EMMIncrement logic within it. We do not show the EMMDecrement here. This statement of the EMMRare algorithm assumes a simple *Nearest Neighbor* clustering technique is used [4]. Notice that all steps in the algorithm can be performed in constant time except for steps 1 and 2 which are *O(m)* where *m* is the number of nodes in the

EMM. Notice that *m* is the number of nodes in the EMM. We will see that it is much smaller than the number of data inputs.

4. EXPERIMENTS

In this section we examine the results of experiments which analyze the general behavior of EMMs in the application of rare event detection. We first introduce the datasets used in these studies, then examine overall EMM performance as a spatiotemporal modeling tool, and finally examine its use for rare event detection. Our performance studies examine the scalability of EMM, analyze its ability to continue learning, compare its use with both Nearest Neighbor and BIRCH clustering techniques, examine two different cluster representations, and finally look at its ability to detect rare events.

4.1. Data

In our experiments we have examined the use of EMMs with four different datasets.

Two datasets contain data of river sensor readings for rivers in England: Ouse Catchment data obtained from: http://www.ccg.leeds.ac.uk/simon/nndown.html and Serwent Catchment data from: http://www.nerc-wallingford.ac.uk/ih/nrfa. Ouse Catchment data were supplied by the Ridings Area Office of the Environment Agency North-East. Three sensors from three river level gauges, namely: Skelton (the nearest to the city of York), the upstream station at Crakehill and the upstream station at Skip Bridge. The target sensor for prediction is Skelton. Total number of data readings is 1920. The sensor readings were taken every 15 minutes. There are three groups of data: 480 data are readings from Aug. 25, 1986 to Aug. 29, 1986; 672 data are readings from Dec. 25, 1994 to Dec. 31, 1994 and 768 data are from Jan. 26, 1995 to Feb. 23, 1995. The first and second groups of data were used for model construction, whereas the third group of data was used for model testing. Serwent Catchment data were provided by the British National River Flow Archive. There are seven sensors in the upper Serwent catchment. Water flow rate (m^3/s) is read from every location once per day. The date range that all seven locations have data available is from Dec. 1971 to Jan. 1977, totally 1574 time points.

MnDot is the traffic data collected from freeways of the Minnesota Twin City metropolis [16]. In our experiments, the traffic volume data collected every 5 minutes in March of 2004 at Monitor Stations 119 and 120 located near I-494/I-35W are used.

VoIP traffic data was provided by Cisco Systems and represents logged VoIP traffic in their Richardson, Texas facility from Mon Sep 22 12:17:32 2003 to Mon Nov 17 11:29:11 2003. We only counted connected calls and discarded unconnected calls. The total number of connected calls was 272,646. We defined 5 phone classes (internal, local, national, international, unknown); therefore we have up to 25 link classes (source class + destination class combinations) in the network.

In the following subsections we report on various experiments performed using these four datasets. Each set of experiments was conducted on each type of data. We only discuss representative results here. In all cases, other tested datasets behaved similarly to those reported. The first experiments examine the scalability of EMM; the second examined the ability of EMMs to continue to learn. These experiments were performed without the decrement operation and without thresholds for rare event detection.

4.2. Scalability

The first set of experiments evaluated the growth rate of EMM. We examined the number of nodes in the graph with respect to different similarity measures and threshold values used. It is important to select the appropriate similarity measure as well as threshold to give accurate prediction. A threshold which is too tight will increase the number of states. Sublinear growth rates of 0.2-0.4% were found with the Serwent and Ouse Catchment data with an optimal threshold. Same conclusion can be seen in Figure 4(a), where the number of states is 38 while the data size is 5422, and the growth rate turns to be 0.7%. The sublinear growth rate is obtained due to the clustering of similar input events into the same node in the MM.

Figure 2 shows the distribution of EMM node sizes using the Minnesota traffic dataset. There were 38 different EMM nodes. The size of the clusters (that is the number of real world events belonging to these nodes) ranged from about 10 to almost 400. The ability of the EMM to dynamically group similar real world events together ensures its scalability.

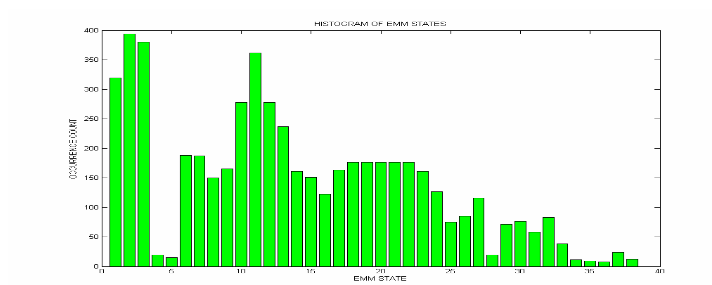

Fig. 2. Node Size Distribution

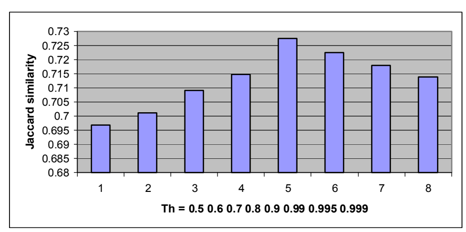

Fig. 3. Jaccard Precision for Prediction in Serwent Data

The clustering threshold has a direct impact on the size of the EMM. In addition, it impacts whether and how the EMM may suffer from overfitting and underfitting. Figure 3 illustrates this when Serwent cachement data was used. With EMMpredict, we are able to predict the numeric values for k^{th} future steps using the maximum expectation concept. Comparing the measured values and predicted values in the same time span with a range of threshold values, we are able to obtain a curve which shows the precisions using the Jaccard similarity between the actual value and predicted value using EMM with eight different threshold values. The turning point represents an empirically optimal value that EMM should use. More experiments show that the optimal value of threshold is least dependent on the performance metrics selected. If the threshold is selected too "tight", overfitting may occur. Otherwise if the threshold is too "loose", underfitting may occur. Both overfitting and underfitting may lead to an inefficiency of the model.

4.3. Continued Learning

One of the most important advantages of EMM is its dynamically adjusted structure, i.e., the states and links in model. Figure 4 illustrates the results of using EMM with Cisco VoIP traffic data. Each point on x axis in Figures 4 a) and b) represents a 15 minute interval. Figure 4 a) shows the growth rate of the EMM, while Figure 4 b) shows the call traffic count. The line on the top of Figure 4 a) shows the growth rate of the EMM based on number of nodes. Each point that intersects any horizontal line represents a reoccurrence of this node in the graph. As can be seen, the growth rate of EMM is sublinear. Out of 5422 time points in this data, we only had 38 EMM nodes. This is less than 1% of what it would be if the growth were linear.

Notice not only the sublinear growth rate (as discussed earlier), but also the fact that when the EMM has learned the model, the growth stops. It starts again when the model changes. This makes the EMM ideal for a dynamically changing environment such as traffic (Web or automobile). Figure 4 b) shows the weekday/weekend pattern for the calls.

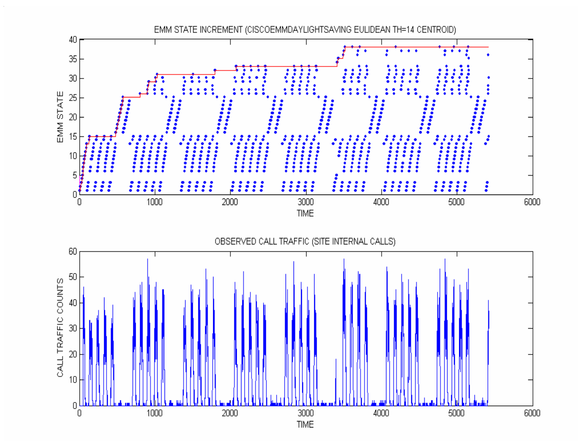

Fig. 4. EMM Applied to Cisco VoIP Traffic Data
a) Growth Rate of EMM States b) Call Traffic Counts

4.4. More Efficient Clustering

The EMMRare algorithm introduced above used a Nearest Neighbor clustering technique which requires *O(m)* time. In this subsection we examine the results of experiments performed to investigate whether a more efficient algorithm, in this case BIRCH [17], can be used as the basis of a more efficient clustering technique. BIRCH uses a hierarchical tree structure similar to a B+ tree. Each step in the BIRCH algorithm requires *O(lg m)* as opposed to *O(m)* time. In fact, due to the dynamic requirements of EMMRare we can only use the first half of the BIRCH technique – namely the creation of the tree to facilitate efficient search of target clusters.

In these experiments we examine the growth rate of EMM using 6 different branching factors for BIRCH. These experiments were performed on the Cisco data, assuming a Euclidean distance measure. As seen in Figure 5, the BIRCH performs very poorly due to the replication of clusters. As the branching factor increases from 3 to 200, both the run time and number of EMM states decreases. In this case the actual number of input events is 62.

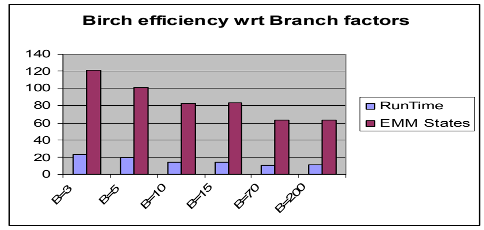

Fig. 5. Impact of Branching Factor with BIRCH

We conclude this subsection with a summarization of the overall experimental results for BIRCH. Although BIRCH can be used to reduce the number of nodes because it performs clustering, it does not behave as well as a simple Nearest Neighbor (NN) clustering. NN is using Euclidean distance, with a centroid cluster representation, and a similarity threshold of 0.1. This is due to the fact that identical events may be placed in two separate clusters. Thus the number of clusters can be much higher with BIRCH than with NN. Recall that the original BIRCH algorithm has a second phase where the first cut clusters are re-clustered. Due to the online dynamic nature of the rare event detection algorithm this second phase can not be performed.

4.5. Cluster Representation

As mentioned earlier, a cluster (EMM node) may be represented by either a medoid or centroid. In this subsection we report on experiments decide to determine which is better. The spatiotemporal environments we are investigating may indeed change over time. For example, traffic patterns vary over time. Similarity and distance functions are performed by examining the input event vector and the cluster representative. With a medoid representation, the cluster representative remains fixed (assuming we choose the first input event placed in this cluster as the medoid). With a centroid representation, the cluster changes over time as does its representative. Figure 6 illustrate the EMM growth rate with both representatives on the Ouse dataset assuming a Jaccard similarity measure and various similarity thresholds. The centroid consistently outperforms the medoid in terms of number of states with about 10% fewer states. We found similar results with the other datasets.

4.6. Rare Event Detection

As mentioned above, some approaches to detect rare events have been based on the creation of signatures, statistical models, or classification schemes created by examining past "normal" behavior. These static approaches are not adequate in environments (such as Web traffic) where behavior patterns may change. In addition, what is normal for one location may not be normal for another. Previous reported experiments showed the ability of EMM to detect rare events in the Minnesota traffic data [1]. Below we look at results with the VoIP traffic data obtained from Cisco Systems.

Looking at Figures 4 a) and b), there appear to be several unusual events. One occurs on Monday October 27, 2003. This area of the graph is enlarged in Figure 7. Notice that the traffic appeared somewhat normal in the morning, but then unusual later in the day. This rare effect would be detected by a cluster with a low occurrence frequency.

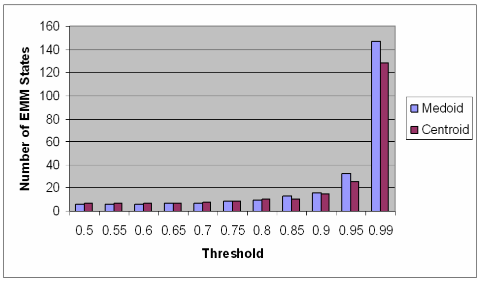

Fig. 6. Growth Rate on Ouse Data with Medoid and Centroid

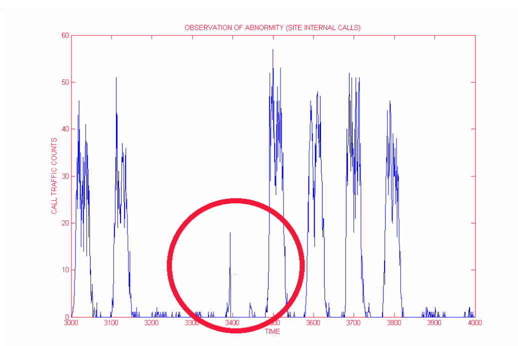

Fig. 7. Anomaly Detected in Cisco Data

5. Summary And Future Work

In this work we have explored the use of EMMs for unsupervised rare event detection in a spatiotemporal environment, adopting representative granules in the data space as states in a dynamic Markov chain. This preliminary study shows that EMMs are a viable approach for rare event detection. Future studies will compare their use with other techniques on datasets with known anomalous behavior. In addition, we continue to explore the use of EMMs for different applications of spatiotemporal data.

Acknowledgment

The authors gratefully acknowledge the Ridings Area Office of the Environment Agency North-East and the British National River Flow Archive for providing the training and testing data, and Cisco Systems, Richardson, TX campus for providing the Voice over IP data.

References

[1] Margaret H. Dunham, Yu Meng, and Jie Huang, "Extensible Markov Model (EMM)," *Proc. IEEE Int'l Conf. Data Mining*, November, 2004.

[2] M. H. Dunham, N. Ayewah, Z. Li, K. Bean, and J. Huang, "Spatiotemporal Prediction Using Data Mining Tools," Chapter XI in *Spatial Databases: Technologies, Techniques and Trends,* Yannis Manolopouos, Apostolos N. Papadopoulos and Michael Gr. Vassilakopoulos, Editors, 2005, Idea Group Publishing, pp 251-271.

[3] Yu Meng and Margaret H. Dunham, "Efficient Mining of Emerging Events in a Dynamic Spatiotemporal Environment", accepted by *PAKDD* 2006.

[4] Margaret H. Dunham, *Data Mining – Introductory and Advanced Topics*, Prentice Hall, New Jersey, 2003.

[5] U. Narayan Bhat and Gregory K. Miller, *Elements of Applied Stochastic Processes Third Edition,* John Wiley & Sons, 2002.

[6] M. J. Black and Y. Yacoob, "Recognizing Facial Expressions in Image Sequences Using Local Parameterized Models of Image Motion", International *Journal of Computer Vision,* Vol. 25, No 1, 1997, pp. 23-48.

[7] Aleksandar Lazarevic, Jaideep Srivastava, Vipin Kumar, "PAKDD 2004 Tutorial: Data Mining for Analysis of Rare Events: A Case Study in Security, Financial and Medical Applications", *Proc. Pacific-Asia Conf. Knowledge Discovery and Data Mining (PAKDD), 2004.*

[8] Ling, C., Li, C. "Data Mining for Direct Marketing: Problems and Solutions", *Proc. ACM Int'l Conf Knowledge Discovery and Data Mining (KDD)*, 1998.

[9] Kubat M., Matwin, S., "Addressing the Curse of Imbalanced Training Sets: One-Sided Selection", *Proc. Int'l Conf. Machine Learning (ICML),* 1997.

[10] N. Chawla, K. Bowyer, L. Hall, P. Kegelmeyer, "SMOTE: Synthetic Minority Over-Sampling Technique", *Journal of Artificial Intelligence Research*, vol. 16, 321-357, 2002.

[11] N. Chawla, A. Lazarevic, et al, "SMOTEBoost: Improving the Prediction of Minority Class in Boosting", *Proc. Pacific-Asia Conf. Knowledge Discovery and Data Mining (PAKDD),* 2003.

[12] P. Domingos, "MetaCost: A General Method for Making Classifiers Cost-Sensitive", *Proc. ACM Int'l Conf Knowledge Discovery and Data Mining (KDD),* 1999.

[13] W. Fan, S. Stolfo, J. Zhang, and P. Chan, "Adacost: Misclassification Cost-Sensitive Boosting", *Proc. Int'l Conf. Machine Learning (ICML),* 1999.

[14] K.M.Ting, "A Comparative Study of Cost-Sensitive Boosting Algorithms", Proc. *Int'l Conf. Machine Learning (ICML),* 2000.

[15] S. Merler, C. Furlanello, B. Larcher, and A. Sboner, "Automatic Model Selection in Cost-Sensitive Boosting", *Information Fusion*, 2003.

[16] Minnesota Department of Transportation, http://tdrl1.d.umn.edu/services.htm, 2005.

[17] T. Zhange, R. Ramakrishnan, and M. Livny, "BIRCH: An Efficient Data Clustering Method for Very Large Databases," *Proceedings of the ACM SIGMOD Conference,* 1996, pp 103-114.