

Extensible Markov Model¹

Margaret H. Dunham and Yu Meng
*Department of Computer Science and
Engineering
Southern Methodist University
Dallas, Texas 75275-0122
mhd(ymeng)@enr.smu.edu*

Jie Huang
*The University of Texas Southwestern
Medical Center at Dallas,
5323 Harry Hines Blvd.
Dallas, Texas 75390-9041
jie.huang@utsouthwestern.edu*

Abstract

A Markov Chain is a popular data modeling tool. This paper presents a variation of Markov Chain, namely Extensible Markov Model (EMM). By providing a dynamically adjustable structure, EMM overcomes the problems caused by the static nature of the traditional Markov Chain. Therefore, EMMs are particularly well suited to model spatiotemporal data such as network traffic, environmental data, weather data, and automobile traffic. Performance studies using EMMs for spatiotemporal prediction problems show the advantages of this approach.

1. Introduction

In this paper we investigate a new modeling tool targeting spatiotemporal problems. An interesting, yet crucial, aspect of this type of data is that any modeling approach must be able to handle the fact that the data changes over time. For example, traffic data changes as a city grows. *Markov Chain (MC)* and its variations are some of the most powerful tools available to engineers and scientists for analyzing complex systems. In many real applications, such as spatiotemporal event prediction, problems with the use of MCs include:

1. The required structure of the MC may not be certain at the model construction time.
2. As the real world being modeled by the MC changes, so should the structure of the MC.

This paper presents a new data modeling technique, named *Extensible Markov Model (EMM)*. EMM is essentially a time varying Markov Chain. It has the advantage of learning and adjusting its structure (number of states) as well as state transition probabilities based on the input data seen. Applications initially examined using the EMM approach include prediction of river flow rate/water level, prediction of traffic volumes for both networks and roadways, identification of rare events in roadways, and identification of rare events for network traffic.

In the next section we provide an overview of previous related work. Section 3 introduces EMM,

Section 4 provides preliminary results of performance experiments, and we conclude the paper in Section 5.

2. Related Work

A first order Markov Chain is a finite or countably infinite sequence of events $\{E_1, E_2, \dots\}$ over discrete time points, where $P_{ij} = P(E_j | E_i)$, and at any time the future behavior of the process is based solely on the current state [1]. Throughout this paper, we use a view of Markov Chain as depicted by a directed graph. The graph has a fixed structure although the labels of the arcs may change. (Note that in this paper we use arc, link and transition interchangeably; and use node, to specifically refer to a vertex in the MC (EMM).) The Markov Property states that the next transition depends only on the current state regardless of the past history, which implies that $P_{12\dots j} = P(E_1)P(E_2 | E_1) \dots P(E_j | E_{j-1})$. Although higher order Markov Chains can be studied, in this paper we only look at first order Markov Chains.

An MC is used to model the transition of real world events (states). Each node in the MC corresponds to a real world state. An MC is typically 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.

There have been several proposals to extend the static MC which we briefly examine in the following paragraphs.

One approach is to start with a single state model and create new states by splitting an existing state until a good model is found for the training data. This model construction measure is employed in several applications when using a *Hidden Markov model (HMM)* [2]. The determination of appropriate state to split is complicated and requires an exhaustive search of the model space.

Another approach of dynamically constructing an MC was proposed in [4], where certain states and their

¹ This material is based upon work supported by the National Science Foundation under Grant No. IIS-0208741.

transitions were cloned (duplicated) due to satisfaction of certain criteria. The number of states could grow with the input data, whereas the number of state representatives is still the same, no new state representatives are identified with the input data coming in.

Dani Goldberg and Maja Mataric came up with a so called *Augmented Markov Model (AMM)* [6], which created new states if the input data has never been seen in the model, and transition probabilities are adjusted according to the traverse times kept with the link versus the number of times in the transit from state. After the model is constructed in a dynamic manner, it will not be updated in prediction phase.

3. Extensible Markov Model Overview

Our proposed EMM model is similar to AMM, but is more flexible:

- 1) EMM continues to learn during the application (prediction, etc.) phase.
- 2) The EMM is a generic incremental model whose nodes can have any kind of representatives.
- 3) State matching is determined using a clustering technique.
- 4) EMM not only allows the creation of new nodes, but deletion (or merging) of existing nodes. This allows the EMM model to “forget” old information which may not be relevant in the future. It also allows the EMM to adapt to any main memory constraints for large scale datasets.
- 5) EMM performs one scan of data and therefore is suitable for online data processing.

Definition 1: Extensible Markov Model (EMM): at any time t , EMM consists of an MC with designated current node, N_n , and algorithms to modify it, where algorithms include:

- 1) *EMMCluster*, which defines a technique for matching between input data at time $t + 1$ and existing states in the MC at time t .
- 2) *EMMIncrement* algorithm, which updates MC at time $t + 1$ given the MC at time t and clustering measure result at time $t + 1$.
- 3) *EMMDecrement* algorithm, which removes nodes from the EMM when needed.

It is crucial to keep in mind that each node in the EMM represents a cluster of a set of real world events/states which belong to that cluster. The labeling of a node actually depends on the clustering approach used as well as the representative identified for that cluster. We propose the use of a centroid or medoid for this labeling. Future work will examine the use of Birch [9] and in that case the labeling of a node would be the CF-feature for that cluster.

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 represents the cluster to which

the the previous input state was determined to belong. *EMMCluster* is used to place a given event at time t , E_t , into one of the clusters represented by the nodes in the EMM, G . If the current event is not sufficiently “close enough” to any current cluster, a new node (and thus a new cluster) is created. For simplicity, we assume a nearest neighbor algorithm [5]. However any incremental algorithm can be used *EMMIncrement* is called by *EMMCluster* to update and add nodes incrementally to an EMM.

EMMIncrement calculates transition probabilities by keeping counts that indicate the number of times a node has been found to be the current node. Note that we use CN_i to represent the size of the cluster represented by node N_i and CL_{ij} to be the number of times the transition from N_i to N_j has occurred. We initially define the state transition probability to be $P_{ij} = (CL_{ij} / CN_i)$. Example 1 illustrates the use of *EMMCluster* and *EMMIncrement*.

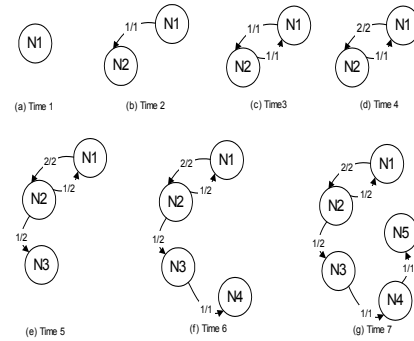


Figure 1. EMMIncrement example

Example 1. This example uses hypothetical automobile traffic data shown below:

- Time 1: < 20 50 100 30 25 4 10 >
- Time 2: < 20 80 50 20 10 10 10 >
- Time 3: < 40 30 75 20 30 20 25 >
- Time 4: < 15 60 30 30 10 10 15 >
- Time 5: < 40 15 25 10 35 40 9 >.
- Time 6: < 5 5 40 35 10 5 4 >
- Time 7: < 0 35 55 2 1 3 5 >

Each real world state can be viewed as a vector of seven values obtained from sensors located at specific points on roads. Each sensor collects a count of the number of vehicles which have crossed this sensor in the preceding time interval. In the following we show the steps involved in the construction of the EMM for the above data. Figure 1, shows the creation of the EMM. We assume the use of Jaccard similarity with a threshold of 0.8 and a nearest neighbor clustering technique

Time 1: Initialize $N_1 = V_1 = < 20 50 100 30 25 4 10 >$, $CN_1 = 0$.

Time2: $Sim_{jaccard}(V_2, N_1) = 0.7342$. A new state N_2 is created and represented by vector. $N_2 = V_2 = < 20 80 50 20 10 10 10 >$. A state transition from N_1 to N_2 has been made. Update $CL_{12} = 1$ and $CN_1 = 1$.

Time 3: $\text{Sim}(V_3, N_1) = 0.8497$, $\text{Sim}(V_3, N_2) = 0.6559$. Therefore V_3 is mapped to state N_1 . The state transition is from N_2 to N_1 . Update the parameters $\text{CN}_2 = 1$ and $\text{CL}_{21} = 1$.

Processing continues in this manner.

If the EMM is determined to be too large, *EMMDecrement* can be executed to decrease the size of the EMM. Our initial version of this algorithm is assumed to be executed manually, although it could be merged with *EMMIncrement* to develop a complete EMM construction algorithm. We also assume that with each execution of *EMMDecrement*, one node in the graph is removed. The node to be decremented, N_d , is input. When the node is removed, each incoming arc is replaced with the same number of arcs as N_d initially had as outgoing. The associated transition probabilities are the products of the probabilities found in the incoming-outgoing arc pairs.

If multiple nodes are to be removed, this can be performed in an incremental manner. Each connected subgraph to be removed can be compressed into one node and *EMMDecrement* applied to each. *EMMDecrement* can be executed at any time, but will probably be used only when memory space is an issue or if it is known that certain portions of the graph are no longer applicable.

Example 2. To illustrate the *EMMDecrement* process, we describe the removal of node N_2 from Figure 2(a). Initially this node has two incoming arcs and three outgoing arcs. When it is removed, each of the incoming arcs will be replaced with three outgoing arcs. Figure 2(b) illustrates this.

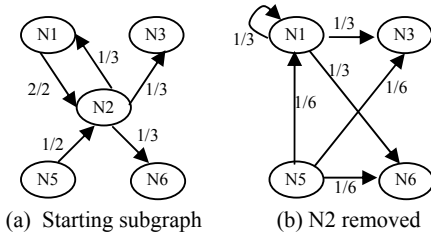


Figure 2. *EMMDecrement* example

As indicated earlier, EMM is a modeling tool. It can be used by many different types of applications. We have initially investigated its use with prediction (forecasting) and rare event detection. The prediction application predicts a future node (or cluster) value at some time point in the future. Prediction is made using the outgoing link with the highest transition probability from the current node. Prediction for multiple time points into the future is made by taking the subsequent links with highest probability. The likelihood of this future event is determined to be the product of the probabilities along the associated path. Example 3 describes the use of EMM for prediction.

Example 3 – Prediction. At time 3 in Figure 1, the current state is N_1 . Based on the EMM at this state, Figure 2 (c), we predict the state at the next time to be state N_2 . If we were to predict two time points into the future, we would predict N_1 .

Rare event detection is somewhat more complicated and space limitations prevent us from discussing this application in detail. The basic idea is that 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 small.

4. Performance

Experiments were performed with actual implementations using Matlab. Three different datasets were used for testing. MnDot is traffic data collected from freeways of the Minnesota Twin City metropolis [7]. 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. Ouse and Serwent are river sensor readings at two different catchments in the United Kingdom [8] [3]. Ouse sensors are located at three river level gauges. The sensors read every 15 minutes. There are three groups of data: 480 data are readings from August 25, 1986 to August 29, 1986; 672 data are readings from December 25, 1994 to December 31, 1994 and 768 data are from January 26, 1995 to February 2, 1995. Serwent data is obtained from seven sensors in the upper Serwent catchment where 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 December 1971 to January 1977.

4.1. Clustering Impact

Table 1. Clustering impact on number of EMM nodes

Data	Sim	Threshold				
		0.99	0.992	0.994	0.996	0.998
Serwent	Jaccrd	156	190	268	389	667
	Dice	72	92	123	191	389
	Cosine	11	14	19	31	61
	Ovrlap	2	2	3	3	4
Ouse	Jaccrd	56	66	81	105	162
	Dice	40	43	52	66	105
	Cosine	6	8	10	13	24
	Ovrlap	1	1	1	1	1

Initial experiments examined the use of different similarity measures [5] using the nearest neighbor clustering algorithm. Table1 shows the number of states created in EMM using the Ouse and Serwent datasets, four different similarity measures, and five thresholds for clustering. As can be seen, there is a dramatic difference in the size of the respective EMM

graphs. Ongoing work is examining the use of more sophisticated clustering techniques such as Birch.

4.2. Size

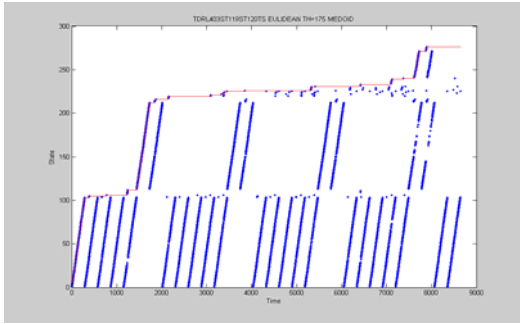


Figure 3. State growth of EMM for MnDot dataset

We now examine the growth rate of EMM using MnDot traffic data. In Figure 3, each entry on the y-axis represents a unique node in the EMM. Over time, as events occur, these nodes may be reused (as the event is placed into an already existing cluster). Thus the real benefit of the EMM model is illustrated when multiple real world events are placed into the same cluster (node). This is shown by repeated y values for multiple x values. Figure 3 shows that of the MnDot. If a horizontal line is drawn on the graph, the number of intersections which occur indicate the actual size of the cluster at that point. Of course, this depends on the data. In fact in this figure not only the unique clustering of weekday vs. weekend traffic is captured, but also the occurrence of some anomalous behavior is identified on the upper right hand corner of this graph. During this last weekend, there was an extremely low amount of traffic at these locations.

4.3. Prediction Accuracy

Performances of predictions are evaluated by two metrics, Normalized Absolute Ratio Error (NARE) and Root Means Square (RMS), as defined below:

$$NARE = \frac{\sum_{t=1}^N |O(t) - P(t)|}{\sum_{t=1}^N O(t)} \quad RMS = \sqrt{\frac{\sum_{t=1}^N (O(t) - P(t))^2}{N}}$$

Where $O(t)$ is the observed profile and $P(t)$ is the predicted profile, N is the length of the dataset and t is the time variable or the t^{th} tuple in the input dataset.

Table 2. Water level prediction of EMM and RLF

		NARE	RMS	No of States
RLF		0.321423	1.5389	
EMM	Th=0.95	0.068443	0.43774	20
	Th=0.99	0.046379	0.4496	56
	Th=0.995	0.055184	0.57785	92

Using Ouse data, Table 2 compares the water level prediction accuracy of EMM to that of a neural network solution provided with the RLF system that is available on-line [8]. The data clearly shows that for this experiment the prediction accuracy of EMM is higher than that of the neural network approach, a typical non-linear regression method.

5. Summary and Future Work

We have introduced a new spatiotemporal modeling tool, Extensible Markov Model (EMM). We have very briefly reported our ongoing EMM performance experiments. To summarize, we have found that the size of EMM grows at a sublinear rate being able to take advantage of the clustering aspect of nodes. The degree of clustering (and thus the EMM size) depends on the clustering technique, as well as the dataset. Prediction accuracy is good, and at least as good as one available neural network approach specifically designed for the dataset studied. Future work will look at EMMs and rare event detection in more detail. Birch and other more sophisticated clustering algorithms will be examined.

6. Acknowledgements

The authors gratefully acknowledge the Transportation Data Research Laboratory of the Minnesota Department of Transportation, 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 Nathaniel Ayewah and Zhigang Li for their helpful information.

7. References

- [1] U. Narayan Bhat and Gregory K. Miller, *Elements of Applied Stochastic Processes Third Edition*, John Wiley & Sons, 2002.
- [2] M.J. Black and Y. Yacoob, "Recognizing facial expressions in image sequences using local parameterized models of image motion", *Int. Journal of Computer Vision*, 25(1), 1997, 23-48.
- [3] British National River Flow Archive, <http://www.nercwallingford.ac.uk/ih/nrfa/index.html>.
- [4] G. V. Cormack, R. N. S. Horspool. "Data compression using dynamic Markov Modeling," *The Computer Journal*, Vol. 30, No. 6, 1987.
- [5] Margaret H. Dunham. *Data Mining Introductory and Advanced Topics*. Prentice Hall, 2002.
- [6] Dani Goldberg, Maja J Mataric. "Coordinating mobile robot group behavior using a model of interaction dynamics," *Proceedings, the Third International Conference on Autonomous Agents (agents '99)*, Seattle, Washington, May.
- [7] Minnesota Department of Transportation (MnDot), <http://tdrl1.d.umn.edu/services.htm>.
- [8] River Level Forecaster (RLF/1), <http://www.ccg.leeds.ac.uk/simon/intro.html>.
- [9] Tian Zhang, Raghu Ramakrishnan, and Miron Livny, "BIRCH: An Efficient Data Clustering Method for Very Large Databases," *Proceedings of the ACM SIGMOD Conference*, 1996, pages 103-114.