Road Traffic Prediction by Incorporating Online Information

Tian Zhou
Department of Computer Science & Technology,
Xi'an Jiaotong University
tianzhou@stu.xjtu.edu.cn

Lixin Gao
Department of Electrical & Computer Engineering,
University of Massachusetts Amherst
lgao@ecs.umass.edu

Daiheng Ni
Department of Civil & Environmental Engineering,
University of Massachusetts Amherst
ni@ecs.umass.edu

ABSTRACT

Road traffic conditions are typically affected by events such as extreme weather or sport games. With the advance of Web, events and weather conditions can be readily retrieved in real-time. In this paper, we propose a traffic condition prediction system incorporating both online and offline information. RFID-based system has been deployed for monitoring road traffic. By incorporating data from both road traffic monitoring system and online information, we propose a hierarchical Bayesian network to predict road traffic condition. Using historical data, we establish a hierarchical Bayesian network to characterize the relationships among events and road traffic conditions. To evaluate the model, we use the traffic data collected in Western Massachusetts as well as online information about events and weather. Our proposed prediction achieves an accuracy of 93% overall.

Categories and Subject Descriptors

J.m [**Computer Applications**]: Miscellaneous

Keywords

Traffic prediction; RFID; hierarchical Bayesian network

1. INTRODUCTION

Traffic congestion has a significant impact on economics and people's lives throughout much of the world. But if some congestions can be predicted in advance, some measures can be taken to avoid them, or at least people can be informed so that they can reschedule their travel plans or route effectively.

Some works had been done by modeling the traffic flow itself. Early works [2] about traffic condition prediction focused on using precise analytic formulae to model traffic systems. This kind of approach is challenged by the current more complex and busier traffic trends. Other researchers employed some modern models but most of their works [16, 8] just work on highways because the traffic pattern there is relatively self-consistent and their ubiquitous dedicated sensing infrastructure provides large amount of high quality data.

People also want to know the traffic in city arterials and rural highways which are closer their daily life. But for some reasons few sensor is equipped there. Moreover they can hardly be described with typical traffic models because they are more sensitive to random events, like sports games or extreme weather, which are outside of traffic systems.

With the advantages of web, weather forecast, theater's playbills, vacation schedule of schools and big employers and even government emergency notices can be got easily from the Internet real-time. Incorporated monitored traffic information with such online information, traffic conditions can be predicted more accurately.

RFID technology boomed recent years provides a cheap and promising way for traffic tracking out of highways. Radio-frequency identification (RFID) technology aims at non-contact automatically object identification, like cards and vehicles. A large RFID traffic system called E-ZPass [3] had been established in America. Even it is not designed for monitoring, traffic conditions can still be got with some methods.

To incorporate the heterogeneous information, we proposed a hierarchical Bayesian network and a corresponding $O(n^3)$ learning algorithm. This model is designed for modeling complex predicting systems and it is very friendly to heterogeneous information.

We test this model on a local RFID system deployed in western Massachusetts [14]. This system is used to monitor the traffic state on Route 9 and Route 119 of Massachusetts which are the most important two routes connecting University of Massachusetts Amherst to other towns. Our model reaches an overall accuracy around 93% with a 30-minutes granularity and an overall accuracy around 88% with a finer dynamic granularity up to 10-minute of busy times.

2. MODELING TRAFFIC SYSTEM

2.1 Data Source

One kind of offline information and three kinds of online information are collected, the offline traffic information and online weather, online local events, such as big sports event or musical concert, and online local special days like storm days, legal holidays or university closing days.

Traffic information using E-ZPass[3] RFID data in western Massachusetts is collected from January 1, 2010 to June 30,

WWW'14 Companion, April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04.
http://dx.doi.org/10.1145/2567948.2580072 .

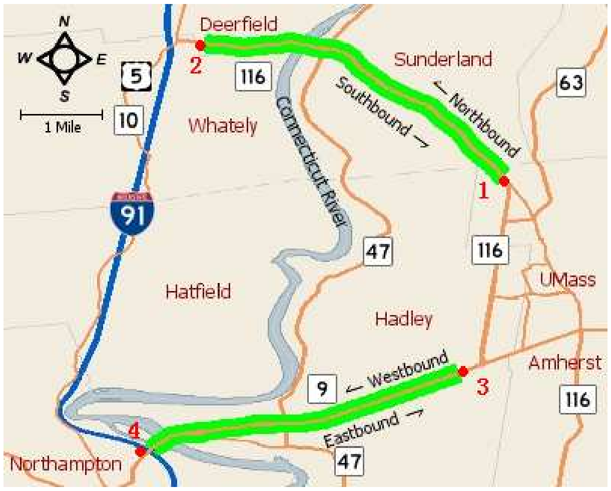


Figure 1: Western Massachusetts RFID monitor system layout ©masstraveler.com

2013. The E-ZPass system is a non-contact automatically identification system which consists of mobile tags and fixed readers. When a vehicle equipped with a E-ZPass tag passes a reader, current time, reader ID and tag ID are recorded. As shown in Figure 1, four sensors shown with red numbers monitor two unconnected rural highways. According to UMass Transit Information Center about 8% vehicles passed by those readers are equipped with E-ZPass tags.

The local weather data is crawled from online website *The Old Farmer's Almanac* [1]. It provides many kinds of meteorological information like temperature, wind speed, visibility, precipitation, snow and so on for each day.

Local events like music concerts, sport events, important lectures, big parties and ceremonies affect the traffic in certain period of time. They are gathered with date and exact time from UMass Amherst Calender of Events [18].

Local special days are the circumstances which may affect the traffic for a whole day. It consists of legal holidays [13], extreme weather and emergency days, local university commencement days and other days when UMass Amherst Campus is closed emergently. They are recognized by the warning notices from the UMass Office of Emergency Management [17]. All special days are stored with categories.

2.2 Modeling the structure

Because our real world is a big causal system, the traffic system can be modeled with causal graphical models. In our model, traffic conditions (detailed defined in 2.3.4) on each road in each time window and other concerned factors are modeled as nodes and the causal relationships among different factors are modeled as direct edges from reason nodes to result nodes. There different states of each factor are represented by the different values of corresponding node.

There is a hierarchical property on this predicting system. According to law of causality, at any time point a current event is impossible to be affect by future events and other events happening simultaneously. So nodes can be separated into several groups according to time. As a simple example in Fig.2 when arranged with temporal ordering, the groups show a hierarchical property.

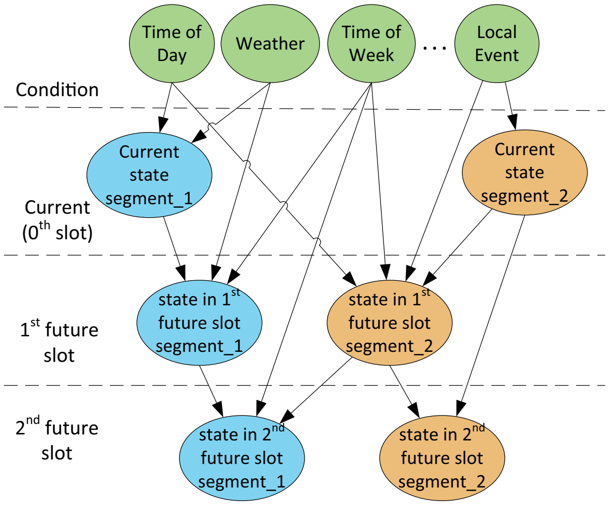


Figure 2: Structure Example of Hierarchical Bayesian Network

Because the causal relationships in the graph is quantified with bayesian theory, we call this kind of model *Hierarchical Bayesian Network*.

2.3 Modeling Nodes

To incorporate heterogeneous information and reduce calculation, values of factors are represented discretely. Continuous values, like time and temperature, are discretized. So do the nodes with too many discrete states.

2.3.1 Weather Nodes

Five kinds of meteorological indexes, visibility, precipitation, snow depth, temperature and wind speed, are adopted.

2.3.2 Event and Special day Nodes

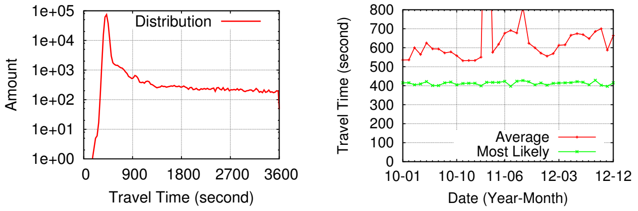
Some nodes describes other local things, like local events and local special days. Events usually affect the traffic for a period of time around them. So its values is according to the holding time of each event. A node "special day" indicates the things can affect the traffic pattern of the whole day. Types of them, like holiday and commencement, mare regarded as the values.

2.3.3 Time Related Nodes

Time-bin is the basic time unit in the model. Each time-bin represents a short period of consecutive time. It is connected with the estimation of traffic conditions discussed later. In simple terms, traffic conditions can only be estimated from data with in certain ranges. And with more samples in a time-bin the estimation of its traffic condition is more accurate. Longer time-bins provides more accurate prediction but the prediction interval gets large, vice versa.

There are several methods of setting time-bins. Intuitively time can be cut evenly with a fixed length. But in fact few people travel in the midnight and roads usually stay clear at that time but in commuting hours there are much more travelers and the traffic condition alters quickly. So the time-bins can be shorter in daytime and longer at night to balance the prediction accuracy and prediction granularity.

Another node "time-of-day", which divides a day coarser, not "time-bin" is used to describe traffic pattern among time



(a) Distribution (b) Travel time

Figure 3: Travel Time Example for northbound Route 116. (a) the travel time distribution with logarithmic coordinate. (b) the average and most-likely travel time for each month.

within a day. Traffic conditions may vary in different time but the traffic trend of that period usual stays the same. Moreover this coarser segmentation lowers the impact of over-fitting.

In the weekend and weekday the traffic pattern is totally different. So a node "time-of-week" indicating part of the week is added. For the same reason, the node "time-of-year" is also modeled. It is set according to the semesters and vacations of schools and larger employers.

2.3.4 Traffic Nodes

Traffic node is that kind of node holding the traffic condition on each road segment on each time-bin. In this model, there are 2 future time-bins and 4 segments named by reader ID pairs for both directions of Route 116 and Route 9. So there are 12 traffic nodes including current conditions.

Additionally some dynamic information is involved. How many time-bins a segment has been falling into abnormal conditions until current time-bin.

The traffic condition describes the average travel time of a traffic node. For generalization it is a discretized value of Travel Time Index (TTI) which is the ratio of current travel time to the normal travel time on that segment. Normal travel time of each segment is defined as the free flow travel time. Supported by the travel time distribution in Fig.3(a) it is represented by the most often travel time in all the data. After discretization the traffic condition values can be referred to as normal, light congestion, substantial congestion etc according to discretization thresholds.

The travel time for the traffic nodes are calculated from RFID records. By tracking the same tag ID, a driver's travel time between two RFID readers is got. The travel time of a traffic node can be estimated from drivers' travel time samples with some methods.

Outliers and the relative low sample density challenges the calculation of nodes' travel time. Without time-consuming thorough analysis, a slower journey itself can hardly be assert in advance to be a outlier or caused by traffic a congestion. As shown in Fig.3(b), speeders are not a big problem because of their poor amount and maybe drivers equipping a trackable tag on their cars are less likely to speeding. While the outliers contributed almost 35% to the average travel time. The number of samples per time-bin is also limited, which makes most statistical methods unreliable.

Two quick methods of getting rid of the road crawler are proposed based on the travel time distribution regulation, in Fig.3. The first method "Discretize-Pick" translates all the travel time samples of a node into conditions and then chooses the most often one as the condition of that node. The other method "Average-Discretize" is inspired from the fact that most normal travelers want to travel faster and the longer cases are usually outliers. So it chooses the samples of the fastest group and consider their average as the travel time of that node.

3. HIERARCHICAL BAYESIAN NETWORK

3.1 Structure

A Hierarchical Bayesian Network (HBN) is a combination of probabilistic graphical theory and hierarchical property. Bayesian Network is a kind of probabilistic graph model for causal relationships. Random events are modeled as nodes and relationships are modeled as directed edges. According to graph theory, source nodes are called *parents* to its destination node. A node may have multiple parent and a node can be many nodes' parent. Probability theory quantifies causal relationships on edges. Conditional Probability Table (CPT) describes the conditional probability distribution of a node's value when its parents' values are under kinds of configuration. Hierarchical property is common in some domains or large systems. Law of causality and some specified domain knowledge restricts some nodes from connecting to some other nodes. According to the equivalence of this feature, nodes in such Bayesian Networks can be grouped. Nodes in the same group are independent and causal relationships only exist among different groups. With some careful ordering, such as topological ordering, all the directed edges are starting from a node in lower ordered clique and ending at a node in a higher ordered clique. Therefore nodes and edges can be viewed as layers and their elevators. So we refer to this kind of special Bayesian Network as *Hierarchical Bayesian Network* (HBN).

Mathematically, nodes in a Hierarchical Bayesian network are grouped into several layers $L_1, L_2, \cdots, L_H$. And for every node $n \in L_i$ any of its parents p, if it exists, must fulfill $p \in L_j$ where $j < i$. It is to say edges are restricted to starting at lower layers and ending at higher layers. In another word if there is an edge between two nodes, the direction has been fixed. All the nodes layered below a node n is referred as *Possible Parent Set* (PPS) of n.

Unlike basic Bayesian networks, the sub-structure property is significant in HBN. A sub-structure represents one elemental causal relationship which consists only one result node and all its parents. So the network can be view as combination of several overlapped sub-structures. All parents are equivalent in a certain sub-structure because there is no edge between then (these edges are considered in other sub-structures). Therefore a sub-structure can be viewed as a small HBN with two layers, the kernel child node in the higher layer and all its parents in the lower layer. In HBNs the maximum number of nodes in a sub-structure is the size of the PPS of its kernel node while in basic Bayesian networks the maximum is the the number of nodes in the whole network! A special kind of sub-structure which contains all the nodes in corresponding PPS is referred to as *Maximum Sub-structure*. And all the sub-structures of the same child node are subsets of its maximum sub-structure.

3.2 Learning Algorithm

The objective of HBN learning is to find the detailed connections among nodes and corresponding CPTs which well supports a given training data.

3.2.1 Scoring

To measure how well a network supports given dataset, Bayesian score[4] is adopted. Based on likelihood function, it is a common measurement for fitness between a dataset and a given Bayesian network. The Bayesian score for a given network G on dataset D is the summation of local scores of each node $X_i \in G$ with its parents Pa_i as equation (1) and (2). In HBNs the local scores are the scores of sub-structures.

$$Score_B = (G : D) = \sum_i FamScore_B(X_i, Pa_i : D) \tag{1}$$

$$FamScore_B(X_i,Pa_i : D) = \log\left(\prod_{v \in Val(P_{a_i})} \frac{\Gamma(\alpha_{(x_i|v)})}{\Gamma(\alpha_{(x_i|v)} + M[v])} \prod_{x_i^j \in Val(X_i)} \frac{\Gamma(\alpha_{(x_i^j|v)} + M[x_i^j, v])}{\Gamma(\alpha_{(x_i^j|v)})} \right) \tag{2}$$

Where $M[x]$ is sufficient statistic of x, which means the occurrence times of x in the dataset D. Hyperparameters $\alpha_{(\dots)}$ represent the Dirichlet priors[6]. They quantify the distribution of the dataset based on prior knowledge about the relationship between relevant nodes. In many real cases, the hyperparameters are assumed to be one. The Bayesian scoring function under this assumption is called *uniform scoring function*.

3.2.2 Structure Learning

Learning the best structure is almost impossible. In general, learning the best structure for a Bayesian network is a hard task because the searching space had been proved NP-complete [11]. Hierarchical property of HBNs reduces the amount of possible structures but the searching space is still exponential to the number of nodes.

So we proposed a heuristic algorithm to find an acceptable structure in polynomial time, as Algorithm 1. The algorithm separated the whole HBN into smaller HBNs using the sub-structure property and hierarchical property and preforms greedy searching on each smaller task.

Learning the whole HBN is separated into learning many small sub-structures. Because the sub-structure scores are the independent local scores in Bayesian score, learning sub-structures separately with Bayesian score and combining them together is equivalent to learning the network as a whole in the perspective of Bayesian score.

On each sub-structure, the algorithm works greedily. In a sub-structure the learning objective is simplified to find an edge set (equivalent to a parent node set) among its PPS to support the given dataset well. This greedy strategy do not enumerate the combinations of parents but just iteratively choses some parents. The algorithm starts from empty chosen set. In each iteration the algorithm lists all unchosen parents until that iteration and try merging each of them into the chosen set to get several candidate sub-structures. These candidate sub-structures are scored by equation (2) and only the one providing highest score is chosen. In most cases the highest score of an iteration is significantly higher than the best score of last iteration. If it happens the candidate sub-structure is accepted. Otherwise when the score improvement is not significant, the learning stops.

Algorithm 1: Hierarchical Greedy Learning

```
Function Hierarchical Greedy Learning
    Input: $G = \langle V, E \rangle$, threshold $\lambda_{improve} > 0$, Data $D$
    foreach node $n \in G$ and $n.level \neq 0$ do
        // PPS of node $n$
        $PPS_n \leftarrow \{x \in V | x.level < n.level\}$
        // learning each sub-structure
        $E' \leftarrow LearningSub(PPS_n, n, \lambda_{improve}, D)$
        $E \leftarrow E \cup E'$
    return $E$
// Learning one sub-structure greedily
Function Learning Sub
    Input: possible parent set $V_p$, node $p$, threshold
           $\lambda_{improve}$, Data $D$
    $V \leftarrow V_p \cup \{p\}$      $E \leftarrow \emptyset$
    $s_l \leftarrow score_B(G_s = \langle V, \emptyset \rangle, D)$
    while $V_p \neq \emptyset$ do
        $p_n \leftarrow \arg\max_p \Big\{$
              $score_B\big(G'_s = \langle V, E \cup (n,p) \rangle, D\big) \Big\}$
        $s \leftarrow score_B(G_s = \langle V, E \cup (n, p_n) \rangle, D)$
        if $s - s_l \geq \lambda_{improve}$ then
            $s_l \leftarrow s$
            $E \leftarrow E \cup \{(n, p_b)\}$
            $V_p \leftarrow V_p - \{p_n\}$
        else // improvement is too small
            break
    return $E$
```

The time complexity of this learning algorithm is $O(n^3)$. At most $m^2/2$ possible sub-structure is scored for a maximum sub-structure with m parents. In the worst cases there are on more than $n^3/2$ possible sub-structure to score in a network with n nodes.

3.2.3 Conditional Probability

After the structure is trained, the *Conditional Probability Table* (CPT) of each causal relationship are calculated from training data. Entries of the table are calculated with the same probability expression as the Bayesian score. Given the parents' value configuration v_p, conditional probability of node x_i taking value x_i^j is calculated as equation (3), where $M[v]$ means the appearance times of value configuration v in training data.

$$\frac{\Gamma(M[x_i^j, v_p] + \alpha_{(x_i^j, v_p)}) \cdot \alpha_{(v_p)}}{\Gamma(M[v_p] + \alpha_{(v_p)}) \cdot \alpha_{(x_i^j, v_p)}} \tag{3}$$

3.3 Prediction

Given the observation values of nodes in the first layer of a HBN, the other nodes' distribution of each value can be calculated with learned structure and CPTs. A Belief Propagation algorithm [15] is adopted. This algorithm calculates the marginal distribution of each unobserved node conditioned on observed nodes.

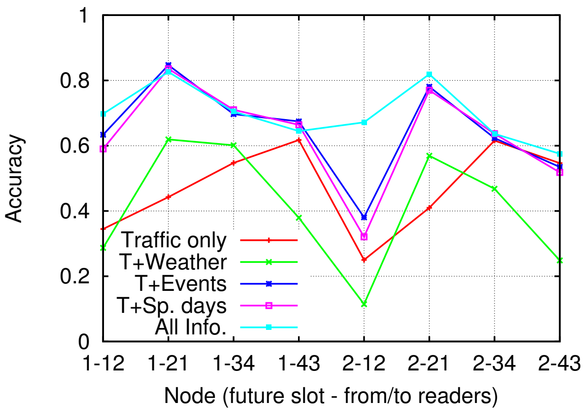


Figure 4: Prediction accuracy on congestion with kinds of information for future nodes

4. EVALUATION

To evaluate the average performance of this model, 150 random days between January 2010 to 30 June 2013 are chosen to be the test set and the data in other days forms the training set. Two future slots are considered in the evaluation. By default a year is divided into 4 parts according to local university academic season and week is divided into 3 parts.

The impact of online information in the prediction is tested. So does the methods of calculating traffic condition from raw RFID data and the methods of dealing with time. To demonstrate the impacts of them separately, a default parameter configuration is adopt unless otherwise stated, which divides a day into 48 time-bins, 4 time-of-days and 2 traffic conditions the ”normal” and ”congested” separated by TTI 1.25 (more than 25% extra travel time leads to congestions) using the ”Average-Discretize” method.

In the following parts more attention is paid to predicting congested conditions because people usually care more about them. Besides because of the big base of normal samples in the dataset the model usually works well in predicting normal condition.

4.1 Impact of Online Information

Online Information is considered the first. Shown in Fig.4 the prediction accuracy via only traffic information and via all the offline and online information is compared. At the same time the respective contributions of all the three kinds of online information, weather, event and special day, are demonstrated separately.

With the traffic information itself, the accuracies on normal and congestion are 93% and 55%. Incorporated with three kind of online information respectively, the overall and congestion accuracies rise to 92% and 60% for just weather, 94% and 69% for just local events and 94% and 68% for just special days. When all three kinds of online information is involved the accuracy rise to 94% overall and 72% on congestion.

The weather information itself contributes little and even jeopardizes some nodes with light traffic. In common sense most weather conditions affect the traffic little. Besides the numerous weather conditions confuse the model learned with limited data by overfitting to the training set.

The influence of local events and special days are very significant. It is consistent with common sense. Apart from life routine most people drive because there is something interested them. And the region this system deployed covers one university and 5 colleges. When there is something like a commencement or a big show, huge number of people drive there.

It is shown that the online information, especially the information about activities, contributes significantly to a local traffic prediction system. For the rest of the evaluation, all the online information is used.

4.2 Impact of Condition Calculating Method

”Discretize-Pick” and ”Average-Discretize” are proposed to get traffic conditions from RFID records in section 2.3.4.

Tested with default configuration, both their overall accuracies reach 90%. But for the accuracy on predicting the congested condition, the ”Discretize-Pick” is only 65.6% while the ”Average-Discretize” method reaches 73.1%. Analysis on the wrongly predicted cases shows that it is because of the shortage of samples in each time bin. For most of the wrongly predicted traffic nodes there are no more than 3 samples so one outlier makes a big influence on the ”Discretize-Pick” method. While the ”Average-Discretize” method ignores some of them actively.

So for a data limited system, the ”Average-Discretize” method is better.

The ”Average-Discretize” method identifies the fastest group by choosing samples whose travel time is less than μ times of the fastest one. When the factor μ grows from 1.5 to 4.0, the accuracy on congested condition keeps decreasing from 71.8% to 62.7% and within the range 1.5 to 2, the accuracy drops little. Intuitively there should be a peek but it is not seen. We think it is because the μ_0 for the peek is smaller than 1.5. But we can’t set μ too small in realty because this will make there are too little samples in each nodes where speeders cannot random errors cannot be handled.

4.3 Impact of Number of Conditions

If the severity of congestion are identified, more detailed traffic condition can be provided. When the number of congested condition is set to 1, 2 and 3, the accuracies on normal condition stay around 93.9% but the average accuracies on congested conditions drop significantly from 73.3% to 61.1% and 51.8%. It is because one original congested condition is split into several conditions while the total number of the congested samples do not change. Less sample makes they are more vulnerable to random errors.

4.4 Impact of Time of day

Three kinds of time-of-day settings (4, 5 and 6) are tested. They are according to the daily traffic pattern like midnight, commuting work, noon, commuting home, night. The different settings differ in the merging and splitting the noon and two commute periods.

With separation getting finer from 4 to 6 the accuracy on normal condition tends to increase from 93.9% to 95.4% while the accuracy on congested condition decreases from 73% to 67% smoothly. Analysis on the learned model and wrongly predicted data shows the model is somewhat overfitted to the training set.

So in-day traffic patterns should not be separated too fine.

4.5 Impact of Time bin

Three kinds of time bin setting are test under the 5 time-of-day setting. The settings are 48-setting (half an hour per bin), 66-setting (half an hour in midnight and 20-minutes for other time) and 89-setting (half an hour for midnight, 20-minutes for normal time and 10-minutes for rush hours).

For the accuracy on all conditions, as imagined, the 48-setting achieves the best at 92.9% and with the finer separation the accuracy drops about 2% and 4% respectively. For the accuracy on congested condition, the decrease is not very significant and all of them are around 68% with a divergence less than 2%. This is because there are more samples in the daytime and shorter time-bins still hold acceptable number of samples in most situations.

So even overall accuracy of the 89-setting is not good as the 48-setting, its prediction has much finer granularity in daytime which make this setting more useful.

5. RELATED WORKS

Herring predicted short term travel time using GPS data [7] in city arterials without other information but this work require huge computations and the result is not very ideal. Horvitz and his colleagues preformed a highway congestion prediction system using some online information like local events and accident reports [8]. For the RFID data, some researchers used it for positioning and tracking vehicles [12] and some other researchers suggested to use it for further traffic management [10].

For the concept of "Hierarchical Bayesian Network" there have been no universal definitions up to now. Researchers defined it on different application environments. Gyftodimos's main idea is abstract nodes [5]. Related nodes are merged as one abstract high level node. Pointing from/to such an abstract node means pointing from/to all its concrete nodes. Hwang [9] separated nodes into layers. But he focused on compression of information and attached many other constraints to the network for this.

6. CONCLUSION

This paper proposed a graphic model called *Hierarchical Bayesian Network* (HBN) and a greedy learning algorithm to train an acceptable HBN within $O(n^3)$ time. Its innate hierarchical and sub-structure properties makes it very friendly to incorporating heterogeneous information for modeling temporal especially prediction systems.

With the model, a traffic condition prediction system around UMass, Amherst is proposed using online and offline information. We evaluated the model under kinds of settings to test the influence to the prediction accuracy of the online information and ways of handling offline information. We also get some methods and conclusions in modeling traffic prediction system with HBN, which might be useful for other researchers.

7. ACKNOWLEDGMENT

This research is partially sponsored by fund of University Transportation Center (UTC) program. The authors would like to thank Dr. Paul W. Shuldiner, Mr. James Schleicher and `masstraveler.com` for their support and help with traffic data. This work is performed when Tian Zhou is a visiting student at University of Massachusetts Amherst.

8. REFERENCES

[1] Amherst weather history,`http://www.almanac.com/weather/history/MA/Amherst%20Center`.

[2] M. Ben-Akiva, M. Bierlaire, H. Koutsopoulos, and R. Mishalani. Dynamit: a simulation-based system for traffic prediction. In *DACCORS Short Term Forecasting Workshop, The Netherlands*. Citeseer, 1998.

[3] `http://www.massdot.state.ma.us/highway/TrafficTravelResources/EZPassMAProgram.aspx`.

[4] J. H. Friedman. On bias, variance, 0/1âĂŤloss, and the curse-of-dimensionality. *Data mining and knowledge discovery*, 1(1):55–77, 1997.

[5] E. Gyftodimos and P. A. Flach. Hierarchical bayesian networks: A probabilistic reasoning model for structured domains. In *Proceedings of the ICML-2002 Workshop on Development of Representations. University of New South Wales*, pages 23–30, 2002.

[6] D. Heckerman, D. Geiger, and D. M. Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243, 1995.

[7] R. Herring, A. Hofleitner, P. Abbeel, and A. Bayen. Estimating arterial traffic conditions using sparse probe data. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pages 929–936. IEEE, 2010.

[8] E. J. Horvitz, J. Apacible, R. Sarin, and L. Liao. Prediction, expectation, and surprise: Methods, designs, and study of a deployed traffic forecasting service. *arXiv preprint arXiv:1207.1352*, 2012.

[9] K.-B. Hwang, B.-H. Kim, and B.-T. Zhang. Learning hierarchical bayesian networks for large-scale data analysis. In *Neural Information Processing*, pages 670–679. Springer, 2006.

[10] Y. A. Kathawala and B. Tueck. The use of rfid for traffic management. *International journal of technology, policy and management*, 8(2):111–125, 2008.

[11] D. Kollar and N. Friedman. *Probabilistic graphical models: principles and techniques.* The MIT Press, 2009.

[12] M. Lu, W. Chen, X. Shen, H.-C. Lam, and J. Liu. Positioning and tracking construction vehicles in highly dense urban areas and building construction sites. *Automation in Construction*, 16(5):647–656, 2007.

[13] The official website of the commonwealth of massachusetts, `http://www.mass.gov/`.

[14] Masstraveler, `http://www.masstraveler.com/`.

[15] J. Pearl. Reverend bayes on inference engines: A distributed hierarchical approach. In *AAAI*, pages 133–136, 1982.

[16] B. L. Smith and M. J. Demetsky. Short-term traffic flow prediction: neural network approach. *Transportation Research Record*, (1453), 1994.

[17] Umass amherst emergency management, `http://www.umass.edu/emergency/`.

[18] Umass calendar of event, `http://www.umass.edu/events/`.