Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 723260, 10 pages
http://dx.doi.org/10.1155/2013/723260

Research Article

Predictive Analytics by Using Bayesian Model Averaging for Large-Scale Internet of Things

Xinghui Zhu,[1] Fang Kui,[1] and Yongheng Wang[2]

[1] *College of Information Science & Technology, Hunan Agricultural University, Changsha 410128, China*
[2] *College of Information Science and Engineering, Hunan University, Changsha 410082, China*

Correspondence should be addressed to Xinghui Zhu; 152062475@qq.com

Received 7 August 2013; Accepted 18 September 2013

Academic Editor: Gelan Yang

Massive events can be produced today because of the rapid development of the Internet of Things (IoT). Complex event processing, which can be used to extract high-level patterns from raw data, has become an essential part of the IoT middleware. Prediction analytics is an important technology in supporting proactive complex event processing. In this paper, we propose the use of dynamic Bayesian model averaging to develop a high-accuracy prediction analytic method for large-scale IoT application. This method, which is based on a new multilayered adaptive dynamic Bayesian network model, uses Gaussian mixture models and expectation-maximization inference for basic Bayesian prediction. Bayesian model averaging is implemented by using Markov chain Monte Carlo approximation, and a novel dynamic Bayesian model averaging method is proposed based on event context clustering. Simulation experiments show that the proposed prediction analytic method has better accuracy compared to traditional methods. Moreover, the proposed method exhibits acceptable performance when implemented in large-scale IoT applications.

1. Introduction

The Internet of Things (IoT) is a novel paradigm that aims to bridge the gap between the physical world and its representation in the digital world. IoT is expected to become an integrated part of the Internet in the future. IoT is defined as a dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols. Bandwidth and storage are no longer considered restricting factors in IoT applications because of the rapid development of novel information and communication technologies. The main issue is how to process the massive events produced by IoT applications, such as the possibility of processing incomplete data streams and historical data from various data sources.

In a large-scale IoT application, the system must process events that arrive from various sources. Such sources include sensors, which constitute the wireless sensor networks, radio-frequency identification (RFID) readers, global positioning systems (GPS), and social media. The events that are directly generated by RFID readers or sensors are primitive events. The semantic information inside primitive events is quite limited. Thus, only simple information can be obtained from primitive events. In real-life applications, people give more attention to higher-level information, such as business logic and rules. For example, each reading operation of an RFID reader at a garage generates a primitive event. However, the user is actually concerned with a complex event such as "the car leaves the garage." Numerous primitive events have to be combined according to certain rules to obtain such complex event. An IoT application system converts business logic into complex events and then detects business logic on the basis of the detected complex events. Complex event processing (CEP) [1] is used to process huge primitive events to obtain valuable information. As an example, in logistics industry, CEP is used to track the goods and trigger some actions when an exception is found.

CEP in active databases has been studied extensively and has recently become a popular research area because of the rapid development of IoT. Most CEP methods assume that the data are deterministic. However, events are imprecise in several real-time IoT applications because of a number of factors,

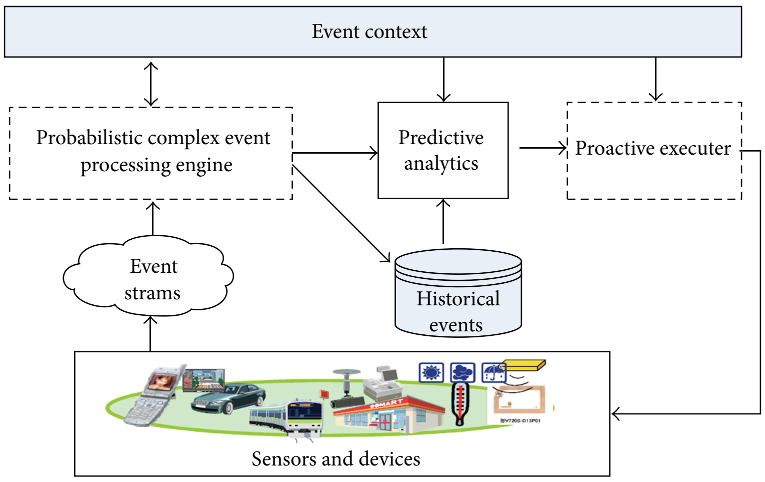

Figure 1: Proactive event processing in IoT.

such as limitation of measuring accuracy, signal disturbance, or privacy protection. The uncertainty is usually treated as probabilities. Therefore, the event processing engine must be capable of supporting probability.

In certain IoT applications, actions can be executed to change the state of the system. Most of these event processing methods are reactive, which indicates that the action is triggered by the state change of the system. A proactive event processing system has the ability to mitigate or eliminate undesired future events or to identify and take advantage of future opportunities by applying prediction and automated decision-making technologies [2]. For example, in a transportation IoT, we can predict a number of congestion states and then perform a number of actions to avoid the congestion states. Figure 1 shows a proactive event processing system in the IoT. The most important part of proactive event processing is predictive analytics (PA), which analyzes historical data to generate predictions regarding a future event.

PA applies several statistical and data mining techniques, such as clustering, classification, and regression. The Bayesian network (BN) [3] and its variations, such as dynamic Bayesian network (DBN) [4] and adaptive Bayesian network (ABN) [5], are extensively used in PA. Bayesian model averaging (BMA) [6] technology, which can average a great number of different competing models to help account for the uncertainty inherent in the model selection process, is also used in PA. When processing massive events in large-scale IoT applications, the performance of PA with BN and BMA is restricted by the challenges posed by big data. In 2012, “Big data” was recognized as one of the three leading edge technology trends that a CEO cannot overlook [7]. Big data can be characterized by volume, velocity, variety, and veracity (“data in doubt”). IoT is one of the most important areas in which big data must be processed. However, most of the existing PA algorithms cannot directly process the big data from IoT. Furthermore, IoT applications, particularly proactive event processing systems, require high-performance PA because such applications have to perform an action at an exact time. To date, few papers have reported on how to integrate CEP and PA to support proactive event-driven systems.

In this paper, we propose a predictive analytics method based on Bayesian model averaging (PABMA). This method can support proactive event processing in large-scale IoT applications. DBN can be created by employing probabilistic CEP. Consequently, basic predictive analytic method is proposed. A model comparison method for BMA is proposed to address the model uncertainty issue. A novel context clustering-based data partitioning and model selecting method is also proposed. The remainder of this paper is organized as follows. In Section 2, we provide a general review of related works on CEP, PA, and BMA technologies. In Section 3, we describe the event context model of our work. The PABMA method is described in detail in Section 4. In Section 5, we present the experiments we conducted to test PABMA. Finally, the discussion and conclusion are presented in Section 6.

2. Related Work

2.1. Complex Event Processing. CEP detects complex events on the basis of a set/sequence of occurrences of single events by continuously monitoring the event stream and then reacting to detected situations. Etzion and Niblett defined the basic concept and architecture of CEP [8] in their book. Event processing agent (EPA) is a component that applies logic on a set of input events to generate a set of complex events as output. Event processing network (EPN) is a network of a collection of EPAs, event producers, and event consumers linked by channels. The network is used to describe the event processing flow execution. Luckman first introduced EPN in the field of modeling [1]. The main idea of complex event detection consists of four steps: (1) primitive events are extracted from large volume data; (2) event correlation or event aggregation is detected to create a business event with event operators according to specific rules; (3) primitive or composite events are processed to extract their time, causal, hierarchical, and other semantic relationships; and

(4) response is sent to the actionable business information because of the guaranteed delivery of events to the subscribers.

The CEP engine must process streams of events with timestamps. Thus, numerous event pattern recognition methods are based on sequential variants of probabilistic graphical models, such as hidden Markov models [9], DBNs [4], and conditional random fields [10]. Methods for detecting complex events in probabilistic event streams based on nondeterministic finite automaton (NFA) have been recently proposed. Xu et al. proposed a data structure called chain instance queues to detect complex events that satisfy query requirements with single scanning probabilistic stream [11]. Conditional probability indexing tree is defined to store conditional probabilities of BN to improve its performance. Kawashima et al. proposed an optimized method for calculating the probability of outputs of compound events and for obtaining the value of confidence of the complex pattern given by the user against uncertain raw input data stream generated by distrustful network devices [12]. The present work is based on CEP and probabilistic CEP.

Proactive applications have been continuously developed in recent years. A number of examples include proactive security systems [13], proactive routing in mobile ad-hoc wireless networks [14], and proactive service level agreement negotiation in service-oriented systems [15]. Engel et al. proposed a proactive event-driven computing framework based on CEP, PA, and Markov decision processes [2, 16]. Engel at al. extended the event processing agent model to include two more types of agents, namely, predictive agents, which may derive future uncertain events based on prediction models, and proactive agents, which compute the best proactive action that should be performed. Proactive CEP requires the prediction of future events or states; such prediction is the main focus of this paper.

2.2. Predictive Analytics with Bayesian Networks. For PA methods based on complex event data, certain attributes of the monitored system can be predicted according to previously monitored events. Such prediction process can be divided into four steps: (1) collect and preprocess raw data; (2) transform preprocessed data into a form that can be easily handled by the (selected) machine learning method; (3) use the transformed data to create the learning model (training); and (4) use the previously created learning model to report predictions to the user. Thus, future events can be predicted by using the recent data on the basis of the learning model trained for previously monitored events.

Bayesian methods are becoming increasingly popular as frameworks for model selection and forecasting tools. Castillo et al. used BN, which considers the random character of the level of the total mean flow and the variability of origin-destination pair flows [17]. Pascale and Nicoli proposed an adaptive BN in which the network topology changes according to the nonstationary characteristics of traffic [18]. In this study, two major stationary areas were recognized as principal phases of traffic flows. Sun et al. modeled traffic flows within adjacent road links in a transportation network as a BN. The joint probability distribution between the cause nodes and the effect node in a constructed BN is described as a Gaussian mixture model (GMM) [19]. Hofleitner et al. used DBN and introduced a model based on hydrodynamic traffic theory to study the density of vehicles on arterial road segments and to illustrate the distribution of delay within a road segment [20]. In contrast to our work, these methods use single BN models and do not consider the massive data from IoT applications.

2.3. Bayesian Model Averaging for Predictive Analytics. Standard data analysts ignore the uncertainty in model selection, resulting in overconfident inferences and decisions that are riskier than they actually seem. The uncertainty inherent in the model selection process is often neglected by traditional statistical analyses. BMA is a technique designed to help account for this uncertainty. By averaging over several different competing models, BMA incorporates the model uncertainty into the conclusions regarding parameters and prediction. The traditional method for BMA is the mixture of experts model [21] proposed by Jacobs et al. In this model, the predictive distribution of various submodels is composed with weight to obtain the predictive distribution of the composed model. The main issue is how to compare and select the models to determine the coefficient of the mixture of expert models.

Model comparison and selection have been recently proposed. Zhou et al. constructed posterior probabilistic properties and model parameters on the basis of sequential Monte Carlo sampling and used these properties to compare different models [22]. With regard to the appropriateness of different genic models to different biological systems, Milias-Argeitis et al. compared and selected Bayesian genic models according to numerous methods, such as annealed importance sampling and approximate Bayesian computation [23]. Karabatsos and Walker proposed a mixed multinomial logit model and presented a Markov chain Monte Carlo (MCMC) algorithm to sample and estimate the posterior distribution of the model parameters [24]. Tawara et al. investigated the effect of the differences of the optimization methods for the multiscale GMM. MCMC-based method was compared with variational Bayesian method in a speaker clustering experiment [25]. Compared with our work, these methods were not optimized for big data from IoT.

3. Event and Context Model for IoT

We present an example of a transporting system based on the IoT to illustrate the event model and the function of the system. The system can obtain information, such as ID, location, and speed, from vehicles by using RFID, radar, GPS, and camera. Other information such as temperature and brightness can be obtained through a wireless sensor network. Through the use of complex events, the system can predict a number of future states (e.g., the congestion state of roads) and then perform actions to support proactive event processing.

Definition 1 (probabilistic primitive event). A primitive event in a stream indicates an atomic occurrence of interest in time.

A probabilistic primitive event is represented by $\langle A, T, Pr\rangle$, where A is the set of attributes, T is the timestamp when the event occurs, and Pr is the concrete probability value used to represent the occurrence probability of the event. The probability value represents the possibility that an event is converted accurately from truthful data to digital data used for computing in electronic devices.

In the transporting system example, each read operation of the devices generates a primitive event. Sometimes, the primitive event is not certain; for example, two RFID readers may find the same object at the same time. The primitive event may also be uncertain when a car accident is detected through a camera. A probability value is used to represent such uncertainty.

Definition 2 (probabilistic complex event). A complex event is a combination of primitive events or complex events according to certain rules. A probabilistic complex event is represented by $\langle E, R, Ts, Pr\rangle$, where E represents the elements that compose the complex event, R represents the rule of the combination, Ts represents the time span of the complex event, and Pr is the probability value.

Definition 3 (event type). The event type is a specification for a set of event objects that have the same semantic intent and the same structure. Every event object is considered an instance of an event type. An event type can represent either primitive events derived from a producer or complex events produced by an event processing agent.

The main complex event patterns in our work include ALL, ANY, COUNT, and SEQ. In this paper, the COUNT event can be used to represent the number of objects in a specified area during a specified time span. The SEQ event can be used to represent the moving path of an object. The detailed meanings of the patterns can be found in [8]. Such patterns can be employed to create hierarchical complex patterns.

Definition 4 (event context). An event context is a specification of conditions which groups event instances so that these instances can be processed in a related manner. The event context assigns each event instance to one or more context partitions.

The context types in our work include "event interval," "fixed location," and "event distance." The detailed definitions of these contexts can be found in [8]. As an example, assume context C_1 means "within 2 km from the motel M_1," context C_2 means "within 10 km from accident A_1," and context C_3 means "traffic status of highway is traffic slow" (Traffic in a certain highway has several status values: traffic flowing, traffic slow, and traffic stationary). In this example, C_1 is an entity distance context, C_2 is an event distance context, and C_3 is a state-oriented context. The context representation in our work is based on the fuzzy ontology framework of [16] and optimized for event processing.

Definition 5 (fuzzy ontology). A fuzzy ontology O in a particular domain Δ is $O_\Delta = (C, R, P, I, A)$, where C is a set of fuzzy concepts, P is a set of fuzzy properties of concepts, I is a set of objects, R is a set of fuzzy roles that denote the relations between two objects, and A is a set of axioms expressed in a logical language.

Definition 6 (fuzzy concept). A fuzzy concept C is defined as $C = \{a_1^{v1}, a_2^{v2}, \ldots, a_n^{vn}\}$, where a_i is an object and v_i is the membership degree of object i in concept C. The degree of object a belonging to a fuzzy concept C is given by the fuzzy membership function $\mu_C : A \to [0,1]$, where A is the set of objects.

Definition 7 (concept subsumption). For two fuzzy concepts $X = \{a_1^{w1}, a_2^{w2}, \ldots, a_n^{wn}\}$ and $Y = \{a_1^{v1}, a_2^{v2}, \ldots, a_n^{vn}\}$, where a_i is an object and w_i and v_i are the membership degrees of the objects. If for all $a_i^{wi} \in X$, $a_i^{vi} \in Y$, and $v_i \geq w_i$, then X is subsumed by Y, which is denoted by $X \subseteq Y$.

Definition 8 (fuzzy role). A fuzzy role R is a fuzzy set of binary relations between two objects in the domain. The fuzzy role is interpreted as a set of pairs of objects from the domain denoted by $R = \{\langle a_1, b_1\rangle^{w1}, \langle a_2, b_2\rangle^{w2}, \ldots, \langle a_n, b_n\rangle^{wn}\}$, where a_i and b_i are two objects and w_i is the degree of the strength of the relation and is given by the fuzzy membership function $\mu_R : A \times B \to [0,1]$, where A and B are sets of objects. The set of objects A is regarded as the domain of the role, whereas the set of objects B is regarded as the range of the role.

Definition 9 (fuzzy role subsumption). For two fuzzy roles $S = \{\langle a_1, b_1\rangle^{w1}, \langle a_2, b_2\rangle^{w2}, \ldots, \langle a_n, b_n\rangle^{wn}\}$, and $T = \{\langle a_1, d_1\rangle^{v1}, \langle c_2, d_2\rangle^{v2}, \ldots, \langle c_n, d_n\rangle^{vn}\}$, if for all $\langle a_i, b_i\rangle^{wi} \in S$, $\langle a_i, b_i\rangle^{vi} \in T$, and $v_i \geq w_i$, then S is subsumed by T, which is denoted as $S \subseteq T$.

Definition 10 (fuzzy property). A fuzzy property P is defined as $P = R \cdot C$, where R is a fuzzy role and C is a fuzzy concept denoting the range of the fuzzy role.

Concept C is the restriction on the range of the role R in property P. This restriction requires that all objects in the range of R should be a member of the concept C. P is interpreted as a fuzzy set of pairs of fuzzy role and fuzzy object such as $(\langle a_i, b_i\rangle, b_i)^{vi}$, where $\langle a_i, b_i\rangle$ is a member of the fuzzy role R, b_i is a member of fuzzy concept C, and v_i is the degree of the object a_i possessing the property P. The degree of the object that processes the property $P = R \cdot C$ is given by the function $\mu_P : R \times C \to [0,1]$, where R is the set of fuzzy roles and C is the set of fuzzy concepts.

Definition 11 (fuzzy property subsumption). For two fuzzy property $P_1 = \{(\langle a, c\rangle, c)^{v1i} \mid (\langle a, c\rangle, c)^{w1i} \in S, c^{y1i} \in C\}$ and $P_2 = \{(\langle a, c\rangle, c)^{v2i} \mid (\langle a, c\rangle, c)^{w2i} \in S, c^{y2i} \in D\}$, if for all $(\langle a, c\rangle, c), (\langle a, c\rangle, c)^{v1i} \in P_1$, and $(\langle a, c\rangle, c)^{v2i} \in P_2$, $v_{2i} \geq v_{1i}$, then P_1 is considered subsumed by P_2, which is denoted by $P_1 \subseteq P_2$.

Definition 12 (fuzzy property with linguistic variable). A fuzzy property can be represented by $\langle V, M\rangle$, where V is a set of linguistic variables and M is the membership function.

For example, "red car" is a fuzzy concept that subsumes another fuzzy concept, that is, "car." "Bob extremely likes a sports car" is a relation of a fuzzy role, and the degree of strength of this relation is very high (extremely). The fuzzy property "drive.speed" can be set to $\langle\{Slow, Medium, Fast\}, \{S, M, F\}\rangle$. S, M, and F are depicted in (1).

Definition 13 (fuzzy context). A fuzzy context (FC) is defined as a triple $\mathrm{FC} = \langle N_c, N_o, N_p\rangle$, where N_c is a set of fuzzy concepts, N_o is a set of objects, and N_p is a set of fuzzy properties. Consider

$$S = \int_0^{100} \frac{((100 - x)/100)}{x},$$
$$M = \int_0^{100} \frac{(x/100)}{x} + \int_{100}^{200} \frac{((200 - x)/100)}{x}, \qquad (1)$$
$$F = \int_{100}^{200} \frac{((x - 100)/100)}{x}.$$

Definition 14 (fuzzy event context). A fuzzy event context (FEC) is defined as $\mathrm{FEC} = \{fc_1^{t1}, fc_2^{t2}, \ldots, fc_n^{tn}\}$, where fc_i^{ti} is the fuzzy context at time t_i.

The context may change during a complex event. The change in context according to time in a complex event can be modeled by using Definition 12. A part of the traffic domain fuzzy ontology is shown in Figure 2. The concept subsumption is implemented using Definitions 7 and 9. Fuzzy properties with linguistic variables are implemented based on Definitions 10, 11, and 12. The event context can be created and reasoned on the basis of this ontology.

The fuzzy ontology is represented according to Fuzzy OWL 2 (http://gaia.isti.cnr.it/~straccia/software/FuzzyOWL/), and the ontology reasoning component is created based on Fuzzy DL (http://www.straccia.info/software/fuzzyDL/fuzzyDL.html).

4. Predictive Analytic by Using Bayesian Model Averaging

In this section, we first introduce how to implement PA with single Bayesian model. Then, a Bayesian model averaging method is proposed for multiple Bayesian models. Finally, to address the model selection problem, a dynamic model selection method based on data partitioning is proposed.

4.1. Basic Predictive Analytic Method. A multilayered adaptive dynamic Bayesian network (mADBN) model for predictive analytics is designed, as shown in Figure 3. The model contains a state plane and a set of location planes. Each plane is an ADBN with two dimensions: time and space. Bayesian networks are directed acyclic graphs whose nodes represent random variables and edges represent the conditional dependences among them. In the state plane, the nodes denote the states in different time instants or spatial locations, whereas edges denote the probabilistic relations of the states. The term "dynamic" in mADBN indicates that we are modeling a dynamic system. Figure 3 shows that the state (i, t) is related to a set of states before time t. The term "adaptive" indicates that the graph structure is created on the basis of the analysis of historical data. Each location plane represents the change in the location of an object (running path). The arrow denotes the transaction probability from one place to another, and the arrow with solid line denotes the real path. The running path of an object is represented as a SEQ event.

The structure of the state plane can be created by analyzing the object location planes. Conditional probability table (CPT) is used to save and sort the conditional probability that an object proceeds to the next place. CPT is learned from massive historical data using Bayesian formula. First, we select nodes before time t that can affect the state of node (t, i) on the basis of CPT as candidates. Then, we use a search-and-score algorithm based on the Bayesian information criterion score [26] to learn the structure of the Bayesian network.

If $f_{i,t}$ represents the flow state of (i, t), $\mathrm{pa}(i, t)$ represents the parent nodes of (i, t), and N_P denotes the number of nodes in $\mathrm{pa}(i, t)$, then the set of flow states for $\mathrm{pa}(i, t)$ is $F_{\mathrm{pa}(i,t)} = \{f_{j,s} : (j, s) \in \mathrm{pa}(i, t)\}$. According to BN theory, the joint distribution of all nodes in the flow state network can be expressed as follows:

$$p(F) = \prod_{i,t} p\left(f_{i,t} \mid F_{\mathrm{pa}(i,t)}\right). \qquad (2)$$

The conditional probability $p(f_{i,t} \mid F_{\mathrm{pa}(i,t)})$ can be calculated as follows:

$$p\left(f_{i,t} \mid F_{\mathrm{pa}(i,t)}\right) = \frac{p\left(f_{i,t}, F_{\mathrm{pa}(i,t)}\right)}{F_{\mathrm{pa}(i,t)}}. \qquad (3)$$

The joint distribution $p(f_{i,t}, F_{\mathrm{pa}(i,t)})$ can be modeled by using GMM as follows:

$$p\left(f_{i,t}, F_{\mathrm{pa}(i,t)}\right) = \sum_{m=1}^{M} \alpha_m g_m\left(f_{i,t}, F_{\mathrm{pa}(i,t)} \mid \mu_m, C_m\right), \qquad (4)$$

where M is the number of nodes and $g_m(\cdot \mid \mu_m, C_m)$ is the mth Gaussian distribution with $(N_P + 1) \times 1$ vector of mean values μ_m and $(N_P + 1) \times (N_P + 1)$ covariance matrix C_m. Parameters $\{\alpha_m, \mu_m, C_m\}_{m=1}^{M}$ can be inferred from the historical data by using expectation-maximization (EM) algorithm [27]. Once $p(f_{i,t}, F_{\mathrm{pa}(i,t)})$ is obtained, the conditional distribution $p(f_{i,t} \mid F_{\mathrm{pa}(i,t)})$ can be derived, and $\hat{f}_{i,t}$ can be estimated from $F_{\mathrm{pa}(i,t)}$ by using minimum mean square error method.

4.2. Bayesian Model Averaging. For a set of H models, the model ensemble posterior distribution of a quantity Q (e.g., the future model predictions using new input data) given the data D can be expressed as follows:

$$p(Q \mid D) = \sum_{k=1}^{H} p(Q \mid M_k, D)\, p(M_k \mid D), \qquad (5)$$

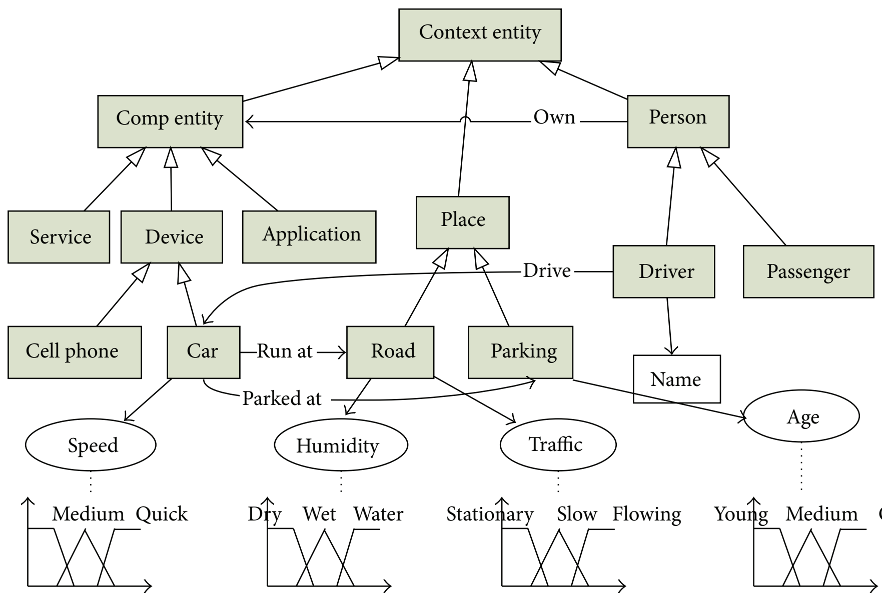

FIGURE 2: A simplified traffic domain fuzzy ontology.

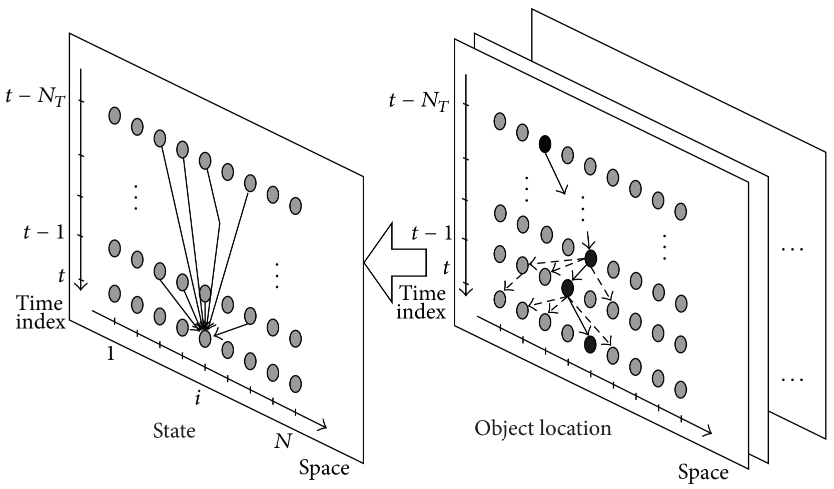

FIGURE 3: The mADBN model.

where $p(Q, M_k, D)$ is the posterior distribution of Q under model M_k and data D and $p(M_k \mid D)$ is the posterior model probability or model weight. This linear combination of predictions is individually performed by each model, in which the weighting coefficients are given by the posterior model probabilities. The precision of different models is compared to obtain the model weight. Assuming that model M_k has a vector of m_k parameters $\Theta_k = (\theta_{1k}, \theta_{2k}, \dots, \theta_{mk})$ and $D = (d_1, d_2, \dots, d_n)$ is a vector of n observation on the m_k parameters, according to Bayesian theory, we obtain

$$p(\Theta_k \mid D, M_k) = \frac{p(D \mid \Theta_k, M_k)\, p(\Theta_k \mid M_k)}{p(D \mid M_k)}, \quad (6)$$

where $p(D \mid M_k)$ is the model evidence or marginal likelihood function, which can be calculated as follows:

$$p(D \mid M_k) = \int p(D \mid \theta_k, M_k)\, p(\theta_k \mid M_k)\, d\theta_k. \quad (7)$$

The posterior probability of each model can be calculated as follows:

$$p(M_k \mid D) = \frac{p(D \mid M_k)\, p(M_k)}{p(D)}, \quad (8)$$

where $p(M_k)$ represents the prior distribution of the model k. In general, if we have no preference for any model, we assume that every model has the same prior probability. $p(D)$ is not related to the models, suggesting that the quality of the different models is primarily determined by $p(D \mid M_k)$. To determine the weight of every model, (8) can be transformed as follows:

$$p(M_k \mid D) = \frac{p(D \mid M_k)}{\sum_{i=1}^{H} p(D \mid M_i)}. \quad (9)$$

Thus, the main issue is calculating the integration in (7). According to Gelfand and Dey [28], the model evidence $p(D \mid M_k)$ can be estimated effectively by using cross-validation distribution as follows:

$$\hat{p}(D \mid M_k) = \prod_{i=1}^{n} p(y_i \mid M_k), (x_i, y_i) \in D, \quad (10)$$

where $p(y_i \mid M_k)$ can be calculated as follows:

$$p(y_i \mid M_k) = \int p(y_i \mid \theta_k, M_k)\, p(\theta_k \mid M_k)\, d\theta_k. \quad (11)$$

If the model is relatively complex, (11) is difficult to calculate directly. Therefore, MCMC method is used for approximate calculation. We can obtain a series of independent samples $\theta_k^{(t)} : t = 1, \ldots, T$ of θ_k from the distribution $p(\theta_k \mid M_k)$ of θ_k through sampling. Then, (11) can be approximated as follows:

$$\hat{p}(y_i \mid M_k) = \frac{1}{T}\sum_{t=1}^{T} p\left(y_i \mid \theta^{(t)} k, M_k\right). \quad (12)$$

The primary goal in using MCMC method is to find independent series of samples. We use a Markov chain $\theta_0, \theta_1, \ldots, \theta_k$ in which every value depends only on its previous value. When certain conditions are satisfied, the series will converge to static distribution $p(\theta)$ after m iterations regardless of the original value of the series. Then, the samples $\theta^{(t)} : t = m+1, \ldots, n$ generated by the subsequent iterations can be used as independent samples of MCMC.

4.3. Dynamic Model Selection Based on Data Partitioning. Event data, even from the same event type, can be appropriate for different models when the system is in different states. We have proved this by experimentations. In our work, different states of the system can be represented by event context. We partition the event data into classes by using a context-based clustering method and then use the method presented in Section 4.2 to determine the appropriate model averaging for each class. When predicting the state for time t, we partition the data in time [t-h-1, t-1], where h is the size of the time window, into the existing classes. These data are typically portioned into one class so that we can use model averaging for that class. If the data are portioned into numerous classes, we reaverage the models of the classes according to the proportion to which the data belongs to each class.

First, we use fuzzy c-means (FCM) method to cluster the historical data. Through iteration process, FCM attempts to determine the cluster center that can minimize the following target function:

$$J_m = \sum_{i=1}^{N}\sum_{c=1}^{C} u_{ij}^{m} \left\| x_i - c_j \right\|^2, \quad 1 \le m \le \infty, \quad (13)$$

where c_j denotes the center of the jth cluster, x_i denotes the ith training sample, and u_{ij} is the affiliation degree at which x_i affiliates to the jth data center. We can calculate u_{ij} as follows:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left((x_i - c_j)/(x_i - c_k)\right)^{1/(m-1)}}. \quad (14)$$

In this study, the samples are complex events. Thus, we calculate the sample distance (or similarity) on the basis of the context of the events. As previously described in Section 3, the fuzzy ontology that we used to represent the event context is of hierarchical structure. The similarity between two nodes is defined based on the distance between them in the hierarchical structure.

$$\mathrm{sim}\left(C_i, C_j\right) := \frac{\alpha_i * \alpha_j}{l_{ij}}. \quad (15)$$

Here, α_i and α_j represent the weight of the context concepts C_i and C_j, respectively, and l_{ij} is the distance between the concepts in the ontology hierarchical structure. Distance is calculated by counting the number of edges in the path between two nodes. We assume the context sets of event e_1 and e_2 to be $C_{e1} = (c_{e11}, \ldots, c_{e1m})$ and $C_{e2} = (c_{e21}, \ldots, c_{e2m})$, respectively, to compare the similarity of the context set of the events. For each $c_{e1i} \in C_{e1}$, a c_{e2j} that satisfies $\max_{ce2k}(\mathrm{sim}(c_{e1i}, c_{e2k}))$ is found. The similarity between C_{e1} and C_{e2} can be calculated as follows:

$$Q\left(C_{e1}, C_{e2}\right) = \frac{n}{k+l-n}\sum_{i=1}^{m} \beta\left(c_{e1i}\right) \mathrm{sim}\left(c_{e1i}, c_{e2j}\right), \quad (16)$$

where function β means weights of similarity. Similarly, the similarity between C_{e2} and C_{e1} can be calculated as follows:

$$Q\left(C_{e2}, C_{e1}\right) = \frac{n}{k+l-n}\sum_{i=1}^{n} \beta\left(c_{e2i}\right) \mathrm{sim}\left(c_{e1i}, c_{e2j}\right). \quad (17)$$

Finally, the similarity between C_{e1} and C_{e2} can be defined as follows:

$$\mathrm{sim}\left(C_{e1}, C_{e2}\right) = \frac{Q\left(C_{e1}, C_{e2}\right) + Q\left(C_{e2}, C_{e1}\right)}{2}. \quad (18)$$

A sample can belong to numerous classes because of the use of fuzzy clustering method. The following librarian criteria are used to adjust the samples that belong to multiple classes.

Librarian Criterion 1 (compactness). Copies of the same book might be placed in different shelves, allowing for multiple classifications. However, the selected classification must minimize the need for multiple copies to reduce costs.

Librarian Criterion 2 (even dimensionality). Books should be evenly distributed in various shelves.

To evaluate compactness, we first calculate the probability that an event context ce_i belongs to a class C_h as follows:

$$\hat{p}\left(C_h \mid ce_i\right) = \frac{\mathrm{Sim}\left(ce_i, C_h\right)}{\sum_j \mathrm{Sim}\left(ce_i, C_j\right)}, \quad (19)$$

where C_j means the class that is different from C_h. Assuming that ce_i is partitioned into k classes in the original partition, we can compute the normalized entropy with

$$H_{\text{norm}}(ce_i) = \frac{-\sum_{h=1}^{k} \widehat{p}(C_h \mid ce_i) \log_2 \widehat{p}(C_h \mid ce_i)}{\log_2 k}. \quad (20)$$

According to the compactness criterion and the meaning of entropy in information theory, the closer $H_{\text{norm}}(ce_i)$ is to 0, the better is the compactness. We calculate $H_{\text{norm}}(ce_i)$ for all ce_i, and the best partition is that with minimum value.

To evaluate the even dimensionality, according to Bayesian theory, we have

$$p(ce_i \mid C_h) = \frac{p(C_h \mid ce_i)\, p(ce_i)}{p(C_h)}, \quad (21)$$

which satisfies $\sum_{i=1}^{N} p(ce_i \mid C_h) = 1$. Assuming that $p(ce_i) = 1/N$ (N is the number of classes), then

$$\begin{aligned} \sum_{i=1}^{N} p(ce_i \mid C_h) &= \sum_{i=1}^{N} \frac{p(C_h \mid ce_i)\, p(ce_i)}{p(C_h)} \\ &= \frac{1}{N \cdot p(C_h)} \sum_{i=1}^{N} p(C_h \mid ce_i) = 1. \end{aligned} \quad (22)$$

Then, we obtain

$$p(C_h) = \frac{\sum_{i=1}^{N} p(C_h \mid ce_i)}{N}. \quad (23)$$

Finally, we define the normalized entropy of C_h as follows:

$$H_{\text{norm}}(C_h) = \frac{-\sum_h \widehat{p}(C_h) \log_2 \widehat{p}(C_h)}{\log_2 k}. \quad (24)$$

We can calculate the average of all $H_{\text{norm}}(C_h)$ for a partition. The larger the average value, the better the distribution.

By clustering massive historical data with different granularity, we can compare the model quality of different classes of data and attempt to find the appropriate data partition and corresponding model averaging. When new events are generated, the events are classified into existing classes by using a similar method.

5. Experimental Evaluations

In this section, we report our experimental study on PABMA. A traffic simulation system is developed based on SUMO [29]. In this system, the mobility trace of cars is supported by using an OpenStreetMap [30] road map of a part of Beijing. A great number of "induction loops," which can detect cars that pass by, are placed on the roads. Virtual RFID or GPS readers are simulated by external applications that use the TraCI interface of SUMO to obtain the induction loop variables. Each induction loop covers a region. The closer a car is to the center of the region, the higher the probability that the event is detected. We select 55 junctions from the map and place 50,000 vehicles into the map. A series of rules is defined to simulate a real-life traffic system. Each vehicle has a home location and an office location. A vehicle v_i runs between home and office with probability α_i. Vehicles go to other places, such as the supermarket or the hospital, with corresponding probabilities. The routing rules are related to contexts such as weather, congestion state, and car accidents. On the basis of this simulation system, the precision and performance of PABMA are evaluated. We use five Lenovo ThinkServer RD series servers with 4 GB of memory as a cluster, and the operating system is Ubuntu Server 12. One server is used for the traffic simulating system, and the others are used for PABMA. The tasks of evaluating different models are partitioned into different servers to run in parallel.

Table 1: Deviation of different models.

	Deviation		
	Single model	SBMA	DBMA
Max	23	19	17
Min	2	3	0
Average	11.3	8.5	6.2

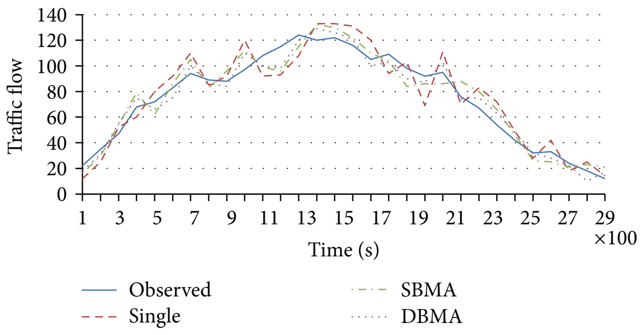

Figure 4: PA accuracy of a typical node.

First, we run the simulation several times to obtain the historical data of the vehicle paths. In the first experiment, the accuracy of the PA methods is evaluated, and the result for a typical node is shown in Figure 4 and Table 1. "Single," "SBMA," and "DBMA" denote the single Bayesian model (described in Section 4.1), the static BMA model (described in Section 4.2), and the dynamic BMA model (described in Section 4.3). Results show that through model averaging, SBMA exhibits better accuracy than the single BMA model does. DBMA has better accuracy than SBMA because an appropriate model averaging is created for different contexts.

In the succeeding experiment, the accuracy of SBMA and DBMA with different clustering granularity is evaluated, and the results are shown in Figure 5. Figure 5 shows that the maximum and average deviation decrease when the cluster number increases. However, when the cluster number increases to a certain number (10 in this experiment), the deviation no longer decreases. The reason is that, in certain extent, the increase of cluster number can make the models and clusters match better. The accuracy of the models with

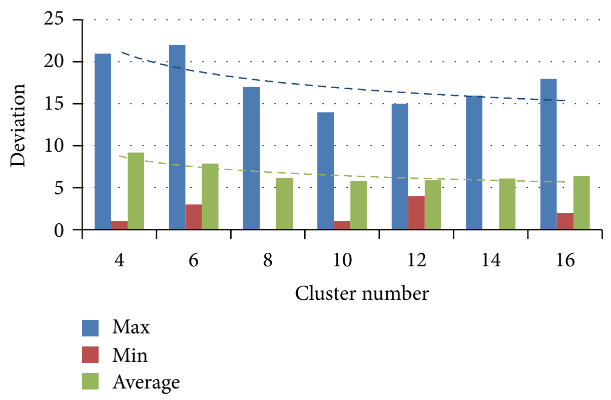

FIGURE 5: Accuracy for different clustering granularity with DBMA model. The training data size is 800 M, and the model number is 8.

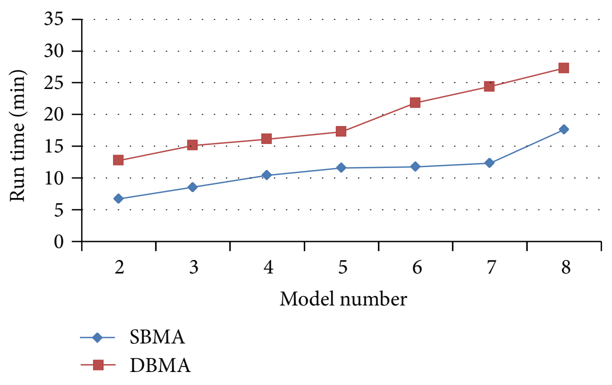

FIGURE 7: Performance for different model numbers. The training data size is 800 M, and the cluster number in DBMA is 10.

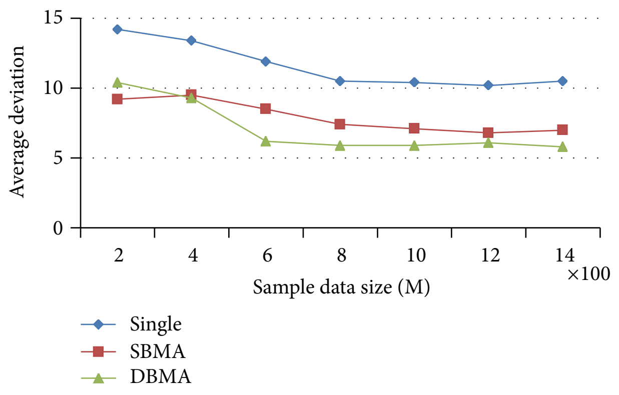

FIGURE 6: Accuracy for different sample data size. The model number is 8, and the cluster number in DBMA is 10.

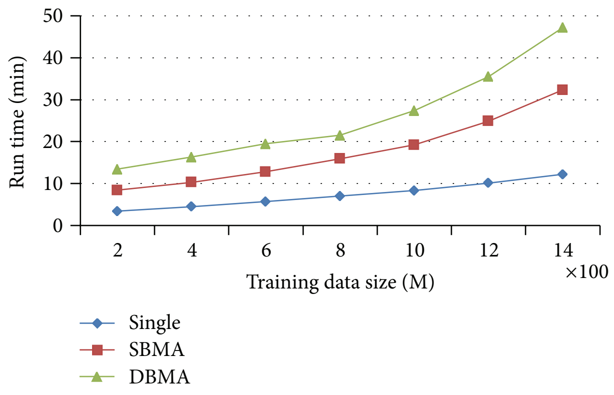

FIGURE 8: Performance for different training data size. The model number is 8, and the cluster number in DBMA is 10.

the sample data size is also evaluated, and the results are shown in Figure 6. The average deviation decreases when the sample data size increases. However, the decline ceases when the sample data size reaches a certain value (approximately 800 M in this experiment). The DMBA model decreases rapidly compared with the other models, suggesting that the DMBA model requires more training data to obtain better accuracy.

In the succeeding experiment, the performance of SBMA and DBMA with different model numbers is evaluated, and the results are shown in Figure 7. Figure 7 shows that the running time of both models increases linearly when the number of models increases. The performance of DBMA is lower than that of SBMA because DBMA has to evaluate models for different contexts. The performance of the three models with different training data sizes is also evaluated, and the results are shown in Figure 8. The running time for all models increases when the training data size increases. However, the running time of SBMA and DBMA increases more rapidly because the calculation becomes more complex. Although DBMA exhibits better accuracy than traditional methods, DBMA also exhibits the worst performance.

All of these experiments show that PABMA outperforms traditional methods for prediction accuracy. However, PMBA requires more training data and running time. This method trains composed Bayesian models according to different event contexts, thereby improving its accuracy. By using GMM with EM inference and MCMC in BMA, the complex calculation problem is resolved. The parallel method is also used in multicontext and multimodel training to improve performance. The performance is acceptable for normal PA applications.

6. Discussion and Conclusion

In this paper, we propose the use of DBMA to develop a high-accuracy PA method for large-scale IoT applications. Based on a multilayered ADBN model, this method uses GMM and EM inference for basic Bayesian prediction. BMA is implemented by using MCMC. DBMA is supported based on context clustering. The experimental evaluations show that this method has better accuracy compared with traditional methods. Moreover, this method exhibits acceptable performance when implemented in large-scale IoT applications.

The performance of PABMA still requires improvement. The current parallel method works only when learning the structure of models and training models for different contexts. EM inference and BMA process are not parallelized yet.

In the future, we plan to develop MapReduce algorithms to support massive historical data and complex training process.

Acknowledgments

This project was supported in part by the National Key Technology R&D Program of China (2012BAD35B07 and 2013BAD15B02) and the Hunan Provincial Natural Science Foundation (13JJ3046).

References

[1] D. C. Luckham, *The Power of Events: An Introductionto Complex Event Processing in Distributed Enterprise Systems*, Addison Wesley, Boston, Mass, USA, 2002.

[2] Y. Engel and O. Etzion, "Towards proactive event-driven computing," in *Proceedings of the 5th ACM International Conference on Distributed Event-Based Systems (DEBS '11)*, pp. 125–136, New York, NY, USA, July 2011.

[3] D. C. Martins Jr., E. A. de Oliveira, U. M. Braga-Neto et al., "Signal propagation in Bayesian networks and its relationship with intrinsically multivariate predictive variables," *Information Sciences*, vol. 225, pp. 18–34, 2013.

[4] H. C. Cho and K. S. Lee, "Nonlinear networked control systems with random nature using neural approach and dynamic bayesian networks," *International Journal of Control, Automation and Systems*, vol. 6, no. 3, pp. 444–452, 2008.

[5] A. Pascale and M. Nicoli, "Adaptive Bayesian network for traffic flow prediction," in *Proceedings of the IEEE Statistical Signal Processing Workshop (SSP '11)*, pp. 177–180, June 2011.

[6] H. Han and P. Bruce, "Bayesian averaging, prediction and nonnested model selection," *Journal of Econometrics*, vol. 167, no. 2, pp. 358–369, 2012.

[7] A. Artikis, O. Etzion, and Z. Feldman, "Event processing under uncertainty," in *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems (DEBS '12)*, pp. 32–43, Berlin, Germany, July 2012.

[8] O. Etzion and P. Niblett, *Event Processing in Action*, Manning Publications, 2010.

[9] L. R. Rabiner and B. Juang, "An introduction to hidden markov models," *IEEE ASSP Magazine*, vol. 3, no. 1, pp. 4–16, 1986.

[10] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: probabilistic models forsegmenting and labeling sequence data," in *Proceedings of the 18th International Conference on Machine Learning (ICML '01)*, pp. 282–289, Morgan Kaufmann, 2001.

[11] C. Xu, S. Lin, and W. Lei, "Complex event detection in probabilistic stream," in *Proceedings of the 12th International Asia-Pacific Web Conference (APWeb '10)*, pp. 361–363, April 2010.

[12] H. Kawashima, H. Kitagawa, and X. Li, "Complex event processing over uncertain data streams," in *Proceedings of the 5th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, pp. 521–526, November 2010.

[13] S. Dolev, M. Kopeetsky, and A. Shamir, "RFID authentication efficient proactive information security within computational security," *Theory of Computing Systems*, vol. 48, no. 1, pp. 132–149, 2011.

[14] T. Kunz and R. Alhalimi, "Energy-efficient proactive routing in MANET: energy metrics accuracy," *Ad Hoc Networks*, vol. 8, no. 7, pp. 755–766, 2010.

[15] K. Mahbub and G. Spanoudakis, "Proactive SLA negotiation for service based systems," in *Proceedings of the 6th World Congress on Services*, pp. 519–526, July 2010.

[16] Y. Engel, O. Etzion, and Z. Feldman, "A basic model for proactive event-driven computing," in *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems (DEBS '12)*, pp. 107–118, July 2012.

[17] E. Castillo, J. M. Menéndez, and S. Sánchez-Cambronero, "Predicting traffic flow using Bayesian networks," *Transportation Research B*, vol. 42, no. 5, pp. 482–509, 2008.

[18] A. Pascale and M. Nicoli, "Adaptive Bayesian network for traffic flow prediction," in *Proceedings of the IEEE Statistical Signal Processing Workshop (SSP '11)*, pp. 177–180, June 2011.

[19] S. Sun, C. Zhang, and G. Yu, "A Bayesian network approach to traffic flow forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 124–132, 2006.

[20] A. Hofleitner, R. Herring, and P. Abbeel, "Learning the dynamics of arterial traffic from probe data using a Dynamic Bayesian Network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1679–1693, 2012.

[21] R. A. Jacobs, M. I. Jordan, S. J. Nowlan et al., "Adaptive mixtures of local experts," *Neural Computation*, vol. 3, no. 1, pp. 79–87, 1991.

[22] Y. Zhou, A. M. Johansen, and J. A. D. Aston, "Bayesian model comparison via path-sampling sequential Monte Carlo," in *Proceedings of the IEEE Workshop on Statistical Signal Processing*, August 2012.

[23] A. Milias-Argeitis, R. Porreca, S. Summers, and J. Lygeros, "Bayesian model selection for the yeast GATA-factor network: a comparison of computational approaches," in *Proceedings of the 49th IEEE Conference on Decision and Control (CDC '10)*, pp. 3379–3384, IEEE, Atlanta, Ga, USA, December 2010.

[24] G. Karabatsos and S. G. Walker, "Bayesian nonparametric mixed random utility models," *Computational Statistics and Data Analysis*, vol. 56, no. 6, pp. 1714–1722, 2012.

[25] N. Tawara, T. Ogawa, S. Watanabe et al., "Fully Bayesian inference of multi-mixture Gaussian model and its evaluation using speaker clustering," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '12)*, pp. 5253–5256, March 2012.

[26] S. Samaranayake, S. Blandin, and A. M. Bayen, "Learning the dependency structure of highway networks for traffic forecast," in *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC '11)*, pp. 5983–5988, December 2011.

[27] J. J. Verbeek, N. Vlassis, and B. Kröse, "Efficient greedy learning of gaussian mixture models," *Neural Computation*, vol. 15, no. 2, pp. 469–485, 2003.

[28] A. Gelfand and D. Dey, "Bayesian model choice: asymptotics and exact calculations," *Journal of the Royal Statistical Society B*, vol. 56, pp. 501–514, 1994.

[29] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "Sumo—simulation of urban mobility: an overview," in *Proceedings of the 3rd International Conference on Advances in System Simulation*, pp. 63–68, Barcelona, Spain, October 2011.

[30] M. Haklay and P. Weber, "Openstreetmap: user-generated street maps," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, 2008.