Inferring Traffic Signal Phases From Turning Movement Counters Using Hidden Markov Models

Mostafa Reisi Gahrooei and Daniel B. Work

***Abstract*—This work poses the problem of estimating traffic signal phases from a sequence of maneuvers. We model the problem as an inference problem on a discrete-time hidden Markov model (HMM) in which maneuvers are observations and signal phases are hidden states. The model is calibrated from maneuver observations using either the classical Baum–Welch algorithm or a Bayesian learning algorithm. The trained model is then used to infer the traffic signal phases on the data set via the Viterbi algorithm. When training with the Bayesian learning algorithm, we set the prior distribution as a Dirichlet distribution. We identify the best parameters of the prior distribution for both fixed-time and sensor-actuated signals using numerical simulations and employ them in the field experiments. It is shown that when the model is trained by the Bayesian learning method with appropriate prior parameters from the Dirichlet distribution, the inferred phases are more accurate in both numerical and field experiments. Because the best set of prior parameters for a fixed-time intersection is different from those for sensor-actuated signals, a classification strategy to distinguish between these two types of signals is proposed. The supporting source code and data are available for download at https://github.com/reisiga2/TrafficSignalPhaseEstimation.**

***Index Terms*—Hidden Markov model, traffic signal phase estimation, TrafficTurk.**

I. Introduction

KNOWLEDGE of the traffic signal phasing is important for a variety of traffic modeling and estimation applications on surface streets. Because the signal phase timing influences the congestion level, knowing the detailed signal operation is important for improving consumer-facing applications, such as estimating the travel time on surface streets, and optimizing routes and fuel efficiency by minimizing stops at intersections. Because the acquisition of signal timing information at large scales is surprisingly difficult in practice, most commercial routing and traffic estimation algorithms do not use signal timing information as input, despite their influence on traffic conditions. Instead, they rely on average traffic conditions, often averaged over several phases or cycles of the light (see, for example, [1]–[4]).

Manuscript received August 18, 2013; revised December 23, 2013 and March 25, 2014; accepted May 12, 2014. Date of publication June 12, 2014; date of current version January 30, 2015. The Associate Editor for this paper was F. Zhu.

M. Reisi Gahrooei is with the Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: reisiga2@illinois.edu).

D. B. Work is with the Department of Civil and Environmental Engineering and with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: dbwork@illinois.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2014.2327225

In principle, accurate traffic signal phasing information should be available from local transportation authorities. Unfortunately, in the United States, this is not always the case in practice. *The 2012 National Traffic Signal Report Card* [5] indicates that the single biggest challenge agencies face is collecting data on the performance of their traffic signal controllers. Moreover, acquiring data over a wide area can be difficult due to the fact that, often, multiple agencies manage the data [6]. Even if the agency has and is willing to share information on their traffic signals, it is not uncommon to find errors between the data sets and the actual field operations [7].

Because of the difficulty and the importance of acquiring traffic signal phasing information, several approaches have been recently proposed to estimate the traffic signal phases from various data streams. For example, Ban *et al.* [6] propose to estimate traffic signal phases using GPS data obtained from smartphones and navigation devices. While these approaches are promising, the relatively highly required penetration rates prevent their current application to most roadways in the United States. Koukoumidis *et al.* [8] use a smartphone camera mounted on the dashboard of a vehicle to identify traffic lights and infer the timing, whereas Barkely *et al.* [9] use hidden Markov models (HMMs) to estimate the traffic signal phase timing plans (with knowledge of the existing phases) using the in-pavement sensors.

The problem of estimating traffic signal phases will likely disappear in the long term, either when all traffic signals become instrumented with sensors and have communication capabilities (for example, see the *SMART-SIGNAL* project [10]) or when GPS penetration rates become high enough. However, due to the relatively high cost of deploying ITS infrastructure and the existing penetration rates of GPS-equipped vehicles, it will still be a long time before the phase information is widely available through these technologies.

Currently and in the foreseeable future, *turning movement counters* will remain a standard approach for obtaining data about the performance of surface streets. While not particularly elegant from a technological standpoint, turning movement counters are accurate (typically within 1%–3% error [11]) and remain the workhorse of many transportation agencies.

Because of the wide use of turning movement counters, it is logical to acquire the traffic signal phases at the same time the counts are recorded. In our experience, the simultaneous recording of turning movement counts and the phase information is not possible for actuated signals, unless the traffic flows are very light. Thus, our work addresses the problem of estimating traffic signal timing information directly from data collected from turning movement counters.

The main contributions of this work are as follows:

- the modeling of the phase estimation problem as an inference problem on an HMM,
- the identification of suitable parameters of the Dirichlet distribution as the prior distribution of the HMM parameters, and
- the introduction of a classification strategy to distinguish fixed-time and sensor-actuated signals using a sequence of maneuvers.

This paper is an extended and improved version of our preliminary work [12], which tested the approach with synthetic numerical data and experimental data using the *Baum–Welch* learning algorithm on fixed-time intersections. When inferring on experimental data, the earlier algorithm was unable to correctly distinguish between phases when specific maneuvers were rarely observed. This paper introduces a Bayesian learning algorithm that significantly outperforms the earlier Baum–Welch algorithm on fixed-time and sensor-actuated signals. It is shown both numerically and experimentally that the Bayesian algorithm allows the phases to be recovered with high accuracy using the classical Viterbi algorithm. Thus, the proposed approach in this paper not only improves the phase inference accuracy for fixed-time signals but also demonstrates good performance on sensor-actuated signals.

In Section II, the definition of an HMM is reviewed, and an HMM model of an intersection is proposed. The classical *Baum–Welch* learning algorithm and a Bayesian learning algorithm, which are used to calibrate the HMM, are described in Section III. In Section IV, the performance of the learning algorithms (Baum–Welch and Bayesian learning) are compared, and the appropriate initial HMM and prior parameters are determined. The results of training and inferring signal phases from experimental data are presented in Section V. Finally, because different prior parameters are required for fixed-time and sensor-actuated signals, Section VI introduces a classification algorithm to distinguish the signal type.

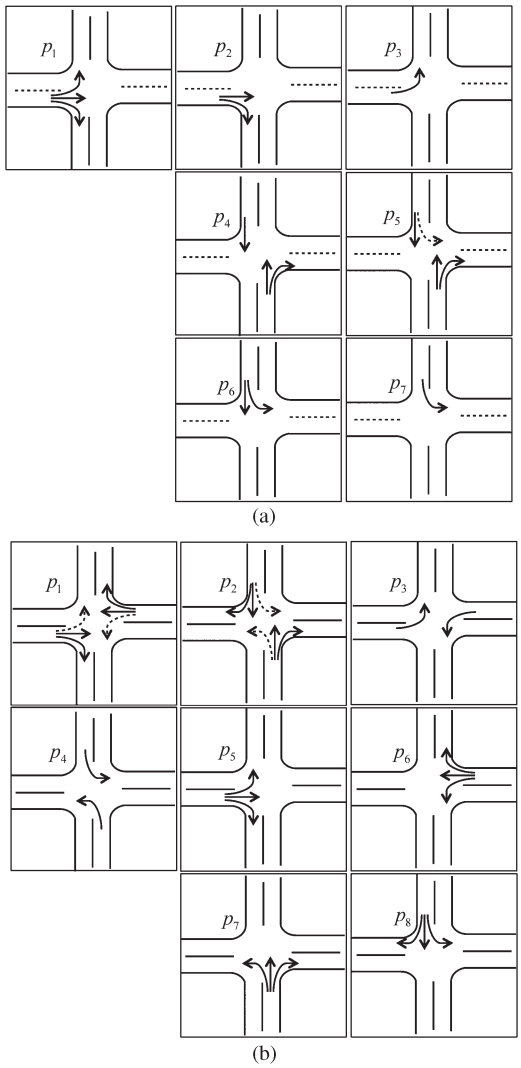

Fig. 1. Phase enumeration for an intersection composed of (a) a one-way (eastbound) street with a two-way (northbound and southbound) street and (b) two, two-way streets. For clarity, protected right turns are shown, and permitted right turns (which appear in all phases) are not shown.

II. Problem Formulation

We propose to use a discrete ergodic HMM to model traffic at an intersection. Formally, the HMM is a 5-tuple (P, V, Π, A, B), where $P = \{p_1, p_2, \ldots, p_N\}$ is a set of N states, and $V = \{v_1, v_2, \ldots, v_m\}$ is a set of m possible outcomes. $\Pi = \{\pi_i\}$ is the vector of initial state probabilities, and $A = \{a_{ij}\}$ is the state transition probability matrix that stores the probability of transitioning from state p_i to state p_j. The matrix $B = \{b_i(v_j)\}$ stores the emission probabilities (e.g., the probability of observing outcome v_j from state p_i). It will be convenient to denote the parameters of the HMM as $\lambda = \{\Pi, A, B\}$.

To estimate traffic signal phases, we construct an HMM for a signalized intersection as follows. The state space P consists of all possible phases at a given intersection. Fig. 1(a) depicts the enumerated states for an intersection of a one-way and a two-way street when traffic flows on the right. For a general four-way intersection, one can enumerate more than 30 possible phases (for example, if we assume a separate state for each maneuver and a separate state for all combinations of two nonconflicting maneuvers, and so on), but only eight of these phases are commonly used in practice [13]. For example, a possible but impractical phase would include protected right turns in all directions, with all other maneuvers prohibited, which we ignore in this work. Fig. 1(b) illustrates the most common phases that appear at a four-way intersection.

The knowledge of the number of streets and one-way restrictions is assumed when building the phase space for the HMM at an intersection. This information is readily available in both commercial and open source map databases such as *OpenStreetMap* (OSM) [14].

To construct the HMM, the possible outcomes or observations V must also be defined. In our model, the set of possible outcomes is simply the set of m maneuvers, which are permitted at an intersection. The total number of maneuvers is 12 (three maneuvers in each direction) at a typical four-way intersection and is six at a three-way intersection. We do not currently consider U-turn maneuvers, although the model can easily be adapted to support them. Finally, the model does

not distinguish between permitted and protected maneuvers in the outcome set since this can be directly inferred from the phase.

With the states and outcomes defined, one needs to define the parameters λ of the HMM. The initial state probabilities π_i denote the probability of the HMM starting in phase p_i. The parameters a_{ij} denote the probability of transitioning from phase p_i to phase p_j. Note that in our definition of the state of the HMM, there is no notion of time. To prevent rapid switching from one phase to another, the HMM must have a large probability of transitioning from the current phase to the same phase and relatively lower probabilities to transition into other phases. It should be noted that this representation would not be particularly helpful for simulating phase evolutions of actual traffic signals, since short phases could only be probabilistically avoided.

The final set of parameters is the emission probabilities $b_i(v_j)$, which define as the probability of observing maneuver v_j when in phase p_i. For example, if maneuver v_j is prohibited in phase p_i, then in principle, $b_i(v_j) = 0$. Moreover, permitted maneuvers can be set to receive lower probability compared with protected maneuvers in a particular phase, and the predominant maneuvers in the phase should have higher emission probabilities.

We briefly mention several other forms of HMMs that could be considered with various improvements for traffic modeling. To capture the fact that the phase switch should directly depend on time, one could consider a continuous-time HMM and explicitly take advantage of the time-stamped observations. However, the continuous-time HMMs have increased mathematical and computational complexities [15], [16] compared with the discrete HMM used in this work. An alternative approach is to discretize time into fixed time steps rather than using the maneuvers to define the time steps. In our experiments, this formulation led to a significant increase in the number of phase inference errors.

It should also be noted that traffic phase evolutions at signalized intersections do not satisfy the Markov assumption, which assumes that the phase in the next time step depends only on the current phase. In reality, the probability of the signal switching from one phase to the next depends also on the duration the signal has been in the current phase and, possibly, traffic conditions at all approaches of the intersection. Several approaches have been developed to extend HMMs to systems that violate the Markov assumption. One approach has been to explicitly model the dwell times in the HMM (see [17] for an overview), resulting in a *hidden semi-Markov model* (HSMM). The HSMM allows for the specification of a variety of distributions on the dwell times, instead of the geometric distribution (implicitly) assumed in the HMM.

Another approach requires specifying a strong prior on the HMM to favor large dwell times. The idea of using priors is employed in *sticky HMMs* [18], [19] to encourage the HMM to stay in the same state for multiple observations. As we will show, this approach works well in practice for phase estimation. It is possible that the development of a continuous-time discrete-outcome HSMM might further improve performance of the phase inference problem considered in this work.

III. HMM Learning and Inference Algorithms

This section summarizes several well-known algorithms to solve learning and inference problems on HMMs [20]–[22]. The learning problem can be stated as identifying the parameters of the HMM given a sequence of observed outcomes. After the parameters of the HMM are defined, the inference problem solves the problem of identifying the phase sequence given the observed maneuvers. The algorithms used to solve these problems are described next.

A. Learning

Let $o = o_1, o_2, \ldots, o_k, \ldots, o_K$ be a sequence of observed maneuvers with a total length of K, in which $o_k \in V$ is the kth observed maneuver. Moreover, let $q_k \in P$ denote the state at step k. The learning problem is to estimate the parameters λ from the observation sequence, i.e., o, while possibly considering prior information on λ.

1) Forward and Backward Algorithms: The *forward algorithm* calculates the probability of a sequence of maneuvers and a state in step k, given a set of parameters, i.e., λ. Define

$$\alpha_k(i) = \Pr(o_1, o_2, \ldots, o_k, q_k = p_i|\lambda) \tag{1}$$

to be the probability of the partial sequence of observed maneuvers up to step k, and the phase p_i appears in step k, given the parameter set λ. The forward algorithm calculates

$$\Pr(o|\lambda) = \sum_{i=1}^{N} \alpha_K(i)$$

where $\alpha_K(i)$ can be sequentially calculated [22].

The *backward algorithm* is similar to the forward algorithm and finds the probability of a sequence of maneuvers starting at the final observation. Let

$$\beta_k(i) = \Pr(o_{k+1}, o_{k+2}, \ldots, o_K|q_k = p_i, \lambda) \tag{2}$$

be the probability of a partial sequence of maneuvers after step k, given the parameters λ and that phase p_i appears in step k. The backward algorithm calculates

$$\Pr(o|\lambda) = \sum_{i=1}^{N} \alpha_1(i)\beta_1(i)$$

where $\beta_1(i)$ can be sequentially calculated [22].

The values of $\alpha_k(i)$ and $\beta_k(i)$ given by (1) and (2) are used in the *Baum–Welch* learning algorithm, which estimates the HMM parameters from the observation sequence.

2) Baum–Welch Algorithm: The *Baum–Welch* algorithm [23] is a generalized *expectation–maximization* (EM) algorithm, which uses an iterative procedure to find parameters λ that locally maximize the likelihood of observing a sequence of maneuvers (i.e., it finds a local maximum of the likelihood function). Let

$$\gamma_k(i) = \Pr(q_k = p_i|o, \lambda)$$

denote the probability that phase p_i appears in step k given a sequence of observations o and a set of parameters λ. Moreover, define

$$\xi_k(i,j) = \Pr\left(q_k = p_i, q_{k+1} = p_j | o, \lambda\right)$$

as the probability that p_i and p_j appear at steps k and $k+1$, respectively, given a sequence of observations o and a set of parameters λ.

Within each iteration step, λ denotes parameters at the start of the step, and $\tilde{\lambda} = (\tilde{\Pi}, \tilde{A}, \tilde{B})$ denotes the updated parameters at the end of the step. The updated parameters $\tilde{\lambda}$ can be computed given the sequence of observations and λ as follows. The learned initial state probabilities $\tilde{\Pi} = \{\tilde{\pi}_i\}$ are given by

$$\tilde{\pi}_i = \gamma_1(i). \tag{3}$$

The learned state transition probabilities $\tilde{A} = \{\tilde{a}_{ij}\}$ and outcome probabilities $\tilde{B} = \{\tilde{b}_i(v_j)\}$ are given by

$$\tilde{a}_{ij} = \frac{\sum_{k=1}^{K-1} \xi_k(i,j)}{\sum_{k=1}^{K-1} \gamma_k(i)} \tag{4}$$

$$\tilde{b}_i(v_j) = \frac{\sum_{k=1 \ s.t.\ o_k = v_j}^{K} \gamma_k(i)}{\sum_{k=1}^{K} \gamma_k(i)}. \tag{5}$$

In (3)–(5), the values of $\gamma_k(i)$, $\xi_k(i,j)$ can be calculated using the values $\alpha_k(i)$ and $\beta_k(i)$ obtained in forward and backward algorithms [22]. Note that (3)–(5) depend on λ through $\gamma_k(i)$ and $\xi_k(i,j)$.

Baum *et al.* [23], [24] proved that the updated parameters $\tilde{\lambda} = (\tilde{A}, \tilde{B}, \tilde{\Pi})$ given by (3)–(5) gives at least as high probability for a given sequence of observations compared with the initial parameters (e.g., $\Pr(o|\tilde{\lambda}) \geq \Pr(o|\lambda)$) within each iteration step. Given this, one can replace λ with $\tilde{\lambda}$ and repeat to iteratively update the parameter values until the improvement of $\Pr(o|\lambda)$ becomes negligible.

3) Bayesian Learning Algorithm: The Bayesian learning approach, similar to the Baum–Welch algorithm, updates the value of the initial HMM parameters based on the observed data in an iterative procedure. Both algorithms are variants of an EM algorithm and, thus, are conceptually and practically very similar. The only difference is that the Bayesian approach allows prior knowledge to be integrated into the learned HMM parameters, i.e., λ. This turns out to be quite important, as it allows us to prevent rapid switching between phases from one observation to the next as part of the prior, and leads to significantly improved performance in our application compared with the Baum–Welch algorithm.

As will be explained, the Dirichlet distribution is a good prior distribution on the parameters for HMMs describing signalized intersections.

Definition 1: Dirichlet Distribution (see, for example, [25] and [26]): Let $W = (W_1, W_2, \ldots, W_N)$ be a random vector whose elements are nonnegative and sum up to one. If W follows Dirichlet distribution with concentration parameter vector $\theta = (\theta_1, \theta_2, \ldots, \theta_N)$ with $\theta_i > 0$, then the probability density function of W can be written as

$$f(w, \theta) = \frac{\Gamma\left(\sum_{i=1}^{N} \theta_i\right)}{\prod_{i=1}^{N} \Gamma(\theta_i)} \prod_{i=1}^{N} w_i^{\theta_i - 1} \tag{6}$$

where Γ is the Gamma function, and $w = (w_1, w_2, \ldots, w_N)$ is a realization of the random vector W.

Note that w_i can be considered as a probability of an event, because the elements w_i are nonnegative and sum up to one. Moreover, by increasing the value of θ_i in (6) relative to the others, we in fact increase the probability of the ith event.

Hence, the parameters of the Dirichlet distribution provide a machinery to impose a prior belief about observing different events. For example, in the context of traffic signals, we can assume that the ith row of the transition matrix A follows the Dirichlet distribution. By increasing θ_i relative to θ_j for $j \neq i$, we can encode a prior preference in the learning algorithms for transition matrices that have a high probability of transitions, which stay in the same phase (i.e., transition from p_i to p_i).

Assuming that Π, A, and B are independent and follow Dirichlet distribution, the density function of λ is [21]

$$f(\lambda) = C \prod_{i=1}^{N} \left(\pi_i^{\vartheta_i - 1} \prod_{j=1}^{N} a_{ij}^{\mu_{ij} - 1} \prod_{k=1}^{m} \left(b_i(v_k)\right)^{\kappa_{ij} - 1} \right) \tag{7}$$

where C is a constant, and ϑ_i, μ_{ij}, and κ_{ij} are the corresponding concentration parameters of the probability density functions of Π, A, and B, respectively. More precisely, ϑ_i is the Dirichlet parameter corresponding to the ith phase in the initial condition Π, and μ_{ij} is the Dirichlet parameter corresponding to the jth element of the ith row of the matrix A. The term κ_{ij} is similarly defined. Considering the density function (7), Hou *et al.* [21] then developed an EM algorithm that iteratively updates the HMM parameters Π, A, and B, as

$$\tilde{\pi}_i = \frac{\gamma_1(i) + \vartheta_i - 1}{\sum_{i=1}^{N} \left(\gamma_1(i) + \vartheta_i - 1\right)}$$

$$\tilde{a}_{ij} = \frac{\sum_{k=1}^{K-1} \left(\xi_k(i,j)\right) + \mu_{ij} - 1}{\sum_{k=1}^{K-1} \left(\gamma_k(i)\right) + \sum_{j=1}^{N} \left(\mu_{ij}\right) - N}$$

$$\tilde{b}_i(v_j) = \frac{\sum_{k=1 \ s.t.\ O_k = v_j}^{K} \left(\gamma_k(i)\right) + \kappa_{ij} - 1}{\sum_{k=1}^{K} \left(\gamma_k(i)\right) + \sum_{j=1}^{m} \left(\kappa_{ij}\right) - m}$$

where N is the number of states, m is the number of possible outcomes, and K is the length of the observation sequence. By selecting sufficiently large values of μ_{ii} in the prior Dirichlet distributions, one can actively keep the diagonal values of the

matrix A close to one during the training iterations to avoid rapid switching from one phase to another.

We note that the Baum–Welch algorithm can be obtained from the Bayesian method by setting all the prior parameters to one. This is not surprising since both of the algorithms are in fact EM algorithms, and by setting all the initial values equal to one, the Dirichlet distribution changes to a multivariate uniform distribution.

B. Inference

The inference problem is to estimate the most likely sequence of phases given an observed sequence of maneuvers and a parameter set λ. In this paper, it is solved using the classical *Viterbi algorithm* [22], [27].

IV. Numerical Validation

This section explores the performance of the HMM framework for inferring traffic phase sequences from synthetically generated maneuvers trained by the Baum–Welch algorithm or the Bayesian learning method. In these tests, we evaluate the performance of the algorithm for inferring the phases as a function of length of the training data, the initial HMM parameters, and the prior Dirichlet distribution parameters. Through the numerical experiments, we discovered that our new Bayesian method significantly outperforms the Baum–Welch approach; however, separate prior parameters are required for fixed-time intersection and sensor-actuated signals to achieve good performance on both signals. A classification procedure for selecting the correct parameters in the prior is discussed later in Section VI.

The numerical experiments have four main components: *1)* obtaining traffic maneuvers via synthetic data generation (i.e., numerical simulation); *2)* initializing the HMM parameters λ (and setting the prior distribution parameters for the Bayesian approach); *3)* training the HMM given the observed maneuvers using the Baum–Welch algorithm or the Bayesian algorithm; and *4)* using the trained HMM to infer the state sequence with the Viterbi algorithm. Steps *3* and *4* use the same data set generated in step *1)*. If the inference is performed on new data sets without retraining, our numerical experiments lead to an increase in the error by 1%–2%.

A. HMM Parameter Initialization

To simplify the parameter initialization for the Baum–Welch algorithm, in the transition probability matrix, we initialize

$$a_{ij} = \begin{cases} a, & \text{if } i \neq j \\ 1-(N-1)a, & \text{if } i = j \end{cases}$$

where N is the total number of phases. Similarly, $b_i(v) = b$ if v is a prohibited maneuver in phase p_i, whereas all allowed maneuvers are given a uniform initial emission probability.

This specific form adds structure to the initial HMM model. By decreasing the value of a, we can initialize the HMM with a high probability of staying in the same phase across multiple observations, although this may not remain true after learning. Moreover, the value of b models how many errors are expected in the data set. Increasing b allows erroneous observations of prohibited errors to be observed without forcing the HMM to transition to a state that allows the maneuver. Finally, the initialized initial probabilities π_i are set to be equal across all phases (i.e., $\pi_i = 1/N$ for $i = 1, 2, \ldots, N$).

When the Bayesian learning algorithm is used, one should set the parameters of the Dirichlet prior distribution on λ. Again, to simplify selection of the parameters of the Dirichlet distribution while still adding meaningful structure, we set

$$\mu_{ij} = \begin{cases} \mu_t, & \text{if } i \neq j \\ \mu_d m_i, & \text{if } i = j \end{cases} \tag{8}$$

where $\mu_t \geq 1$ is a parameter that controls the transition probabilities, and $\mu_d \geq 1$ controls the dwelling time (transitions from one state to the same state), and m_i is the number of allowed maneuvers in phase p_i. We assume that phases with more allowed maneuvers will have longer dwell sequences, as more maneuvers can be observed in those phases. Later, we will show that this structure for the prior parameters is most effective for fixed-time signals, and a different structure can be used for sensor-actuated signals.

We set the values of κ_{ij} so that the permitted maneuvers receive higher weight than prohibited maneuvers. In particular, we assign

$$\kappa_{ij} = \begin{cases} C_s, & \text{if } v_j \text{ is a permitted straight maneuver in } p_i \\ C_t, & \text{if } v_j \text{ is a permitted turn maneuver in } p_i \\ C_p, & \text{if } v_j \text{ is a prohibited maneuver in } p_i \end{cases} \tag{9}$$

where C_s, C_t, $C_p \geq 1$. Suitable values of C_s and C_t can be estimated using historical data by considering the ratio of the turning maneuvers to the straight-through maneuvers to avoid inference errors when turning maneuvers are sparse. In this case, we used our data set. Finally, all ϑ_i in (7) are set to one.

To initialize the Bayesian learning method, we use the mean of the prior density function on the parameters given by

$$a_{ij} = \frac{\mu_{ij}}{\sum_{j=1}^{N} \mu_{ij}} \tag{10}$$

$$b_i(v_j) = \frac{\kappa_{ij}}{\sum_{j=1}^{m} \kappa_{ij}} \tag{11}$$

$$\pi_i = \frac{\vartheta_i}{\sum_{i=1}^{N} \vartheta_i}. \tag{12}$$

B. Experiments on Fixed-Time Traffic Signal

1) Intersection of a One-Way Street With a Two-Way Street: The first test is similar to the first numerical experiment in [12] and serves as a benchmark to highlight the significant performance improvement of the new Bayesian method proposed in this work.

TABLE I
MANEUVER EMISSION PROBABILITIES WITHIN EACH PHASE USED FOR SYNTHETIC DATA GENERATION

Maneuver	SBT	SBR	SBL	WBT	WBR	WBL
Phase p_1 (%)	0.2	4	0.2	0.2	4	0.2
Phase p_5 (%)	35	4	8.1	0.2	4	0.2
Maneuver	NBT	NBR	NBL	EBT	EBR	EBL
Phase p_1 (%)	0.2	4	0.2	58.1	9	20
Phase p_5 (%)	35	9	0.2	0.2	4	0.2

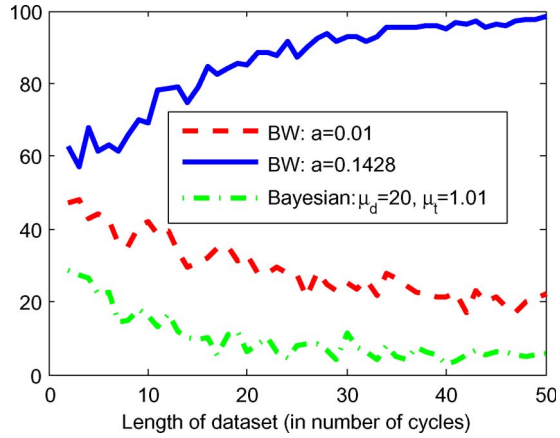

Fig. 2. Average phase inference error (over 30 experiments) using an HMM trained by the Baum–Welch algorithm and the Bayesian approach, as a function of the length of the training data.

To synthetically generate traffic maneuver observations, a state sequence is constructed from alternating phases p_1 and p_5 [shown in Fig. 1(a)] for a fixed number of cycles. The total number of vehicles that pass through the intersection is drawn from $\mathcal{U}$ $[5, 27]$, which ensures that every phase generates some observations. Note that this distribution is selected for numerical illustration, and other bounds could also be considered. Once the number of vehicles within each phase is drawn, the specific movement of each vehicle in the phase is randomly assigned according to the emission probabilities in Table I. For example, the probability of generating an SBT maneuver when in phase p_5 is 35%, and each prohibited maneuver within each phase is assigned a probability of 0.2% (resulting in about 1.1% total error in the data set) to simulate counting errors that occur in data collected during field experiments. Moreover, notice that right-turn maneuvers are permitted in both phases with an emission probability of 4%.

The sensitivity of the performance of the Baum–Welch algorithm to the initial transition probabilities a and the length of the training data was tested, with the emission probability $b = 0.1$. If the initial transition parameters are uniform $(a = 0.1429)$, the phase prediction from the Baum–Welch-trained HMM is generally incorrect, with more than 50% labeling error (see Fig. 2), and the error increases with more training data. When the initialization parameters are selected to strongly favor transitions that remain in the same state $(a = 0.01)$, the total labeling error was reduced to about 10% as the training data span more cycles.

Next, an initial HMM generated from (10)–(12) is trained by the Bayesian algorithm with initial parameter values of $\mu_d = 20$ and $\mu_t = 1.001$. As shown in Fig. 2, learned HMM from the Bayesian approach outperforms the Baum–Welch-learned HMM in nearly every aspect. The method can achieve remarkable accuracy, with errors around 1% for sufficiently long training sequences. The excellent performance is attributed to the good values of the Dirichlet parameters, which encode the prior knowledge that the learned HMM should have long dwell sequences.

TABLE II
MANEUVER EMISSION PROBABILITIES WITHIN EACH PHASE USED FOR ERROR-FREE SYNTHETIC DATA GENERATION FOR A FOUR-WAY INTERSECTION

Maneuver	SBT	SBR	SBL	WBT	WBR	WBL
Phase p_1 (%)	0	1	0	39	5	5
Phase p_2 (%)	39	5	5	0	1	0
Phase p_3 (%)	0	1	0	0	1	48
Maneuver	NBT	NBR	NBL	EBT	EBR	EBL
Phase p_1 (%)	0	1	0	39	5	5
Phase p_2 (%)	39	5	5	0	1	0
Phase p_3 (%)	0	1	0	0	1	48

Because the Bayesian-trained HMM consistently outperforms the one trained by the Baum–Welch algorithm, the remainder of this section will only explore the performance of the Bayesian algorithm.

2) Four-Way Intersection: This section illustrates the performance of the Bayesian algorithm on a common four-way intersection and explores the influence of the Dirichlet prior distribution parameters (μ_t and μ_d) on the performance of the Bayesian algorithm over a wide variety of numerical simulations. The goal is to identify for what values of prior distribution parameters the algorithm performs the best (lowest inference error). This evaluation is particularly important since the right choice of prior distribution parameters is crucial for the algorithm to perform well.

Synthetic data are similarly generated as the experiment in Section IV-B1. The phase sequence alternates between phases p_1, p_2, and p_3 [as illustrated in Fig. 1(b)] for ten cycles. To generate a sequence of maneuvers at each phase, first, the total number of maneuvers was drawn from $\mathcal{U}$ $[5, 27]$ for phases p_1 and p_2 and from $\mathcal{U}$ $[2, 8]$ for phase p_3. The values illustrated in Table II were used as the emission probabilities to randomly assign the specific movement of each vehicle during each phase.

The setup of the sensitivity experiment is as follows. First, a synthetic data set is generated and used to train an HMM [with initial parameters generated from Dirichlet distribution parameters as in (10)–(12)] using the Bayesian learning algorithm. The trained HMMs are then used to infer the phase sequence corresponding to the maneuver counts in the generated data set. The inferred phase sequences are compared with the true ones to determine inference errors. This process was repeated over 30 simulations, and the errors were averaged over all the simulations.

Fig. 3(a) illustrates average inference error as a function of μ_t and μ_d. When $1 \leq \mu_t \leq 5$ and $\mu_d \leq 150$, good inference accuracy is achieved. The values of C_s and C_t were set based on the ratio of the turning maneuvers and straight-through maneuvers. In particular, we set $C_s = 8000$, $C_t = 2000$, and $C_p = 1$ because the straight-through maneuvers occurred about four times as often as through movements in the data sets.

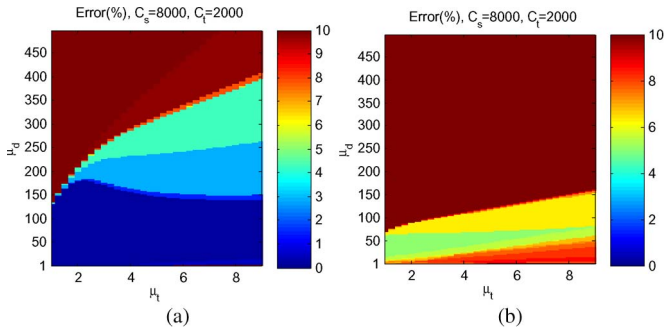

Fig. 3. Inference error as a function of the Dirichlet prior parameters μ_t and μ_d when the observation sequence (a) is error free and (b) contains 6% error. Note the color scale is capped at 10% to enhance visibility.

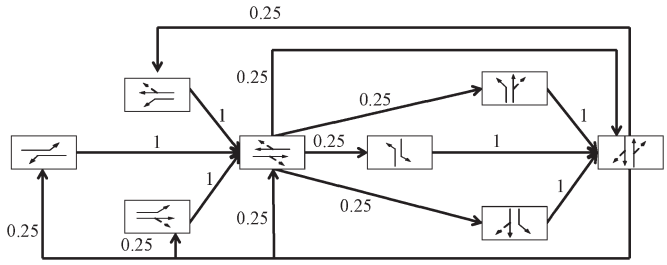

Fig. 4. Markov chain model to generate a phase sequence for sensor-actuated signals. Numbers denote transition probabilities.

Notice that we set the values of C_s and C_t large to keep the probability of prohibited maneuvers low.

In the above experiment, synthetic data were generated assuming that prohibited maneuvers receive zero emission probability. To test the influence of errors in the observation sequence, we increased the emission probability of each of the prohibited maneuvers from 0% to 1% in the data generation step. Because of the introduction of errors, the minimum inference error increased to about 5% [see Fig. 3(b)]. Most of these inference errors are caused by the prohibited maneuvers, which are likely to be labeled as a phase in which they are permitted.

C. Experiments on Sensor-Actuated Signals

Inference on sensor-actuated signals is difficult because the phase sequence depends on the traffic conditions. For example, protected left-turn phases may be skipped or included based on vehicle arrivals at the intersection. The short duration of these phases (if they appear) generates another challenge. When a phase is short in time, only a few maneuvers will be observed, and thus, the phase is not strongly supported by the data. Consequently, the maneuvers might not be separated from adjacent long-duration phases.

To generate synthetic data for a sensor-actuated signal, a common four-way intersection [see Fig. 1(b)] is considered. A sequence of 20 phases is randomly generated using a Markov chain model illustrated in Fig. 4. Next, for each of the phases in the sequence, the total number of vehicles that pass through the intersection is drawn from $\mathcal{U}$ $[5, 27]$ for phases p_1 and p_2, from $\mathcal{U}$ $[2, 8]$ for phases p_3 and p_4, and from $\mathcal{U}$ $[3, 15]$ for the other phases. Once the number of vehicles within each phase is drawn, the specific movement of each vehicle in the phase is randomly identified according to the emission probabilities illustrated in Table III.

TABLE III
MANEUVER EMISSION PROBABILITIES WITHIN EACH PHASE USED FOR ERROR-FREE SYNTHETIC DATA GENERATION FOR A SENSOR-ACTUATED FOUR-WAY INTERSECTION

Maneuver	SBT	SBR	SBL	WBT	WBR	WBL
Phase p_1 (%)	0	1	0	39	5	5
Phase p_2 (%)	39	5	5	0	1	0
Phase p_3 (%)	0	1	0	0	1	48
Phase p_4 (%)	0	1	48	0	1	0
Phase p_5 (%)	0	1	0	0	1	0
Phase p_6 (%)	0	1	0	60	20	20
Phase p_7 (%)	0	1	0	0	1	0
Phase p_8 (%)	60	20	20	0	1	0
Maneuver	NBT	NBR	NBL	EBT	EBR	EBL
Phase p_1 (%)	0	1	0	39	5	5
Phase p_2 (%)	39	5	5	0	1	0
Phase p_3 (%)	0	1	0	0	1	48
Phase p_4 (%)	0	1	48	0	1	0
Phase p_5 (%)	0	1	0	60	20	20
Phase p_6 (%)	0	1	0	0	1	0
Phase p_7 (%)	60	20	20	0	1	0
Phase p_8 (%)	0	1	0	0	1	0

The synthetic data were used to train an initial HMM and to infer phases. Similar to fixed-time intersections, we explored appropriate prior Dirichlet distribution parameters for the Bayesian learning algorithm. For each pair of (μ_t, μ_d), 30 simulations were conducted, and the inference errors were averaged over all the simulations. Similar to the previous experiment, the values of C_s and C_t were fixed to 8000 and 2000, respectively, based on the ratio of observed maneuvers. In this experiment, the best inference performance obtained was when $1 \leq \mu_t \leq 2$ and $1 \leq \mu_d \leq 1.5$. With these values of parameters, the inference error drops to about 5%.

A closer look at the results revealed that most of the errors occur in two specific situations. The first case is when a phase switches to a similar phase (e.g., the allowed maneuvers of one phase are a subset of the next phase). In this case, a few maneuvers at the beginning or end of the phase might be incorrectly labeled by the similar adjacent phase. For instance, when switching from phase p_5 to phase p_1, the last maneuvers generated by phase p_5 might be labeled as p_1 depending on the order of the observations. The second case is when a long sequence of maneuvers generated by one phase also belongs to another phase as well. For example, while in phase p_1, if a subsequence of maneuvers consists only of WBT and WBL, all the maneuvers in this subsequence might be mislabeled as phase p_5. This is because the selected prior parameters do not sufficiently encourage remaining in one phase compared with transitioning into a new phase.

While the first type of error is very difficult to eliminate, the second type can be removed by adding a different structure to the prior parameters based on the knowledge from sensor-actuated signals. For example, instead of setting μ_{ij} with μ_d and μ_t in (8), one can tune the values of μ_{ii} and μ_{ij} based on the common phase transitions that appear in a sensor-actuated signal. For example, by setting $\mu_{ii} = 50\,000$, $\mu_{ij} = 500$ if transitioning from i to j is likely, and $\mu_{ij} = 1.0001$ if transitioning from i to j is not likely, the inference error can be reduced from 5% error to about 1% error. In Section V, we use these prior

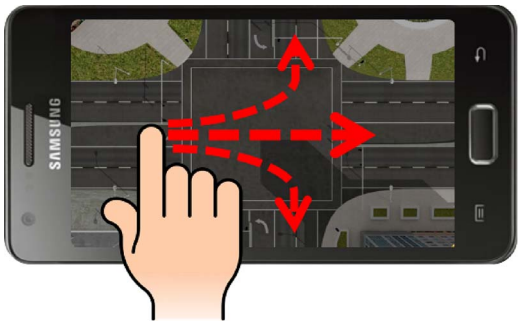

Fig. 5. *TrafficTurk* smartphone turning movement counter.

parameters to infer phases for a sensor-actuated signal. Because of the difference in priors for sensor-actuated and fixed-time traffic signals, a method to distinguish between the signal types is needed. In Section VI, this problem will be investigated.

V. Field Validation

The proposed learning methods were tested on two fixed-time signals and a sensor-actuated signal in Champaign, IL. Turning movements were collected using *TrafficTurk* (see Fig. 5), which is a new and free smartphone-based turning movement counter for Android devices developed for monitoring traffic during special events [28]. Turning movements were recorded with *TrafficTurk*, and phases were manually recorded (with a second person) to validate the inference results.

The collected data are then prefiltered to remove as many errors as possible. Since the phases are unknown during the prefiltering step, the prefilter only exploits the difference in time between each maneuver, and it looks for consecutive maneuvers that have conflicting paths but occur close in time. Specifically, if the maneuver conflicts with the maneuver before and after and all three maneuvers occur in under 5 s, we label the conflicting maneuver as an error and remove it from the training data. Less than 1% of the data is removed as part of the preprocessing filter. During the data collection, the phases at each intersection were recorded to be compared with the inferred phases.

A. Fixed-Time Intersection

The performance of the Baum–Welch algorithm and the Bayesian learning algorithm is compared in the first experiment. For each algorithm, the best parameters identified in Section IV are used.

1) Intersection of a One-Way and a Two-Way Street: In the first experiment, traffic maneuvers were collected at the intersection of W. University Ave. and Prospect Ave. The intersection layout is shown in Fig. 1(a), and the phase sequence is p_1, p_5, and p_6 during each cycle. The turning movements and the true phases were simultaneously recorded for 6 min, with 305 movements over five cycles. The data are preprocessed to eliminate five errors (1.6%).

Fig. 6 shows the time series of observed maneuvers and highlights the sparsity of the SBL and NBR maneuvers. The percentage of SBL maneuvers compared with all maneuvers in

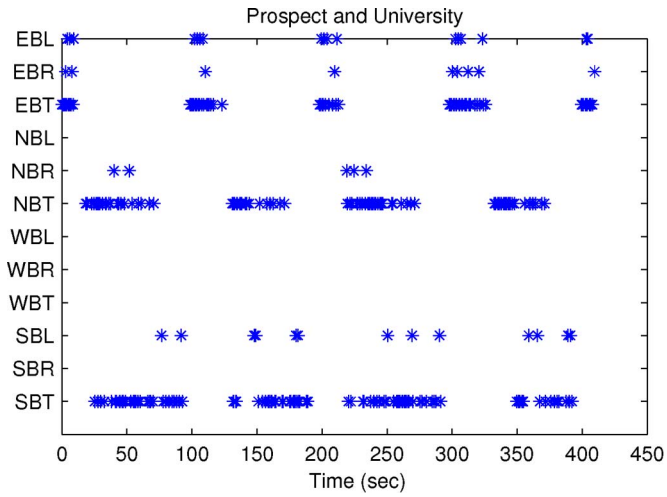

Fig. 6. Turning movement counts collected at the intersection of W. University Ave. and Prospect Ave. as a function of time.

phase p_5 is less than 2.5%, which is known to cause problems in the Baum–Welch learning algorithm.

Fig. 7 shows the inference results based on trained HMMs from each of the learning models. Due to the sparsity of the data, the HMM trained by the Baum–Welch algorithm incorrectly predicts the phase roughly 40% of the time. Although most of these errors are due to the similarity between phases p_4 and p_5, several fast switching errors from p_2 to p_1 also appear. On the other hand, the HMM trained by the Bayesian learning approach estimates the correct phase for all but one maneuver (less than 1% error). Moreover, this single error appears at the end of phase p_5 and is quite difficult to remove, since the maneuver is allowed in both phases.

2) Four-Way Intersection: In the second experiment, data were collected at the intersection of S. Fourth St. and W. Kirby Ave. The possible phases are shown in Fig. 1(b), and the true phase sequence at this intersection is p_1, p_2, and p_3 during each cycle. A total of 464 maneuvers over ten cycles were observed (15 min). The time series of the maneuvers in Fig. 8 shows the sparsity of the right- and left-turn maneuvers. The percentage of NBL and SBL are 1% and 1.5%, respectively, throughout the data set.

The maneuver sequence was prefiltered, but no errors were identified. On this data set, inference with the Bayesian-trained HMM achieves about 2% error (see Fig. 9), all of which appear at the end of the phases.

B. Sensor-Actuated Signal

The third experiment occurred on a sensor-actuated signal at S. Neil St. and W. Kirby Ave., which is also a four-way intersection. Similar to the previous experiments, both maneuver counts and phases were simultaneously collected over about 6 min. In this data set, the prefilter identified two conflicting maneuvers (1%). The prior Dirichlet distribution parameters were selected to encourage transitions that are more likely to appear in sensor-actuated signals, using the parameters described in Section IV-C. The phases were inferred by the Bayesian-trained HMM and were compared with the actual phase sequence. Fig. 10 shows

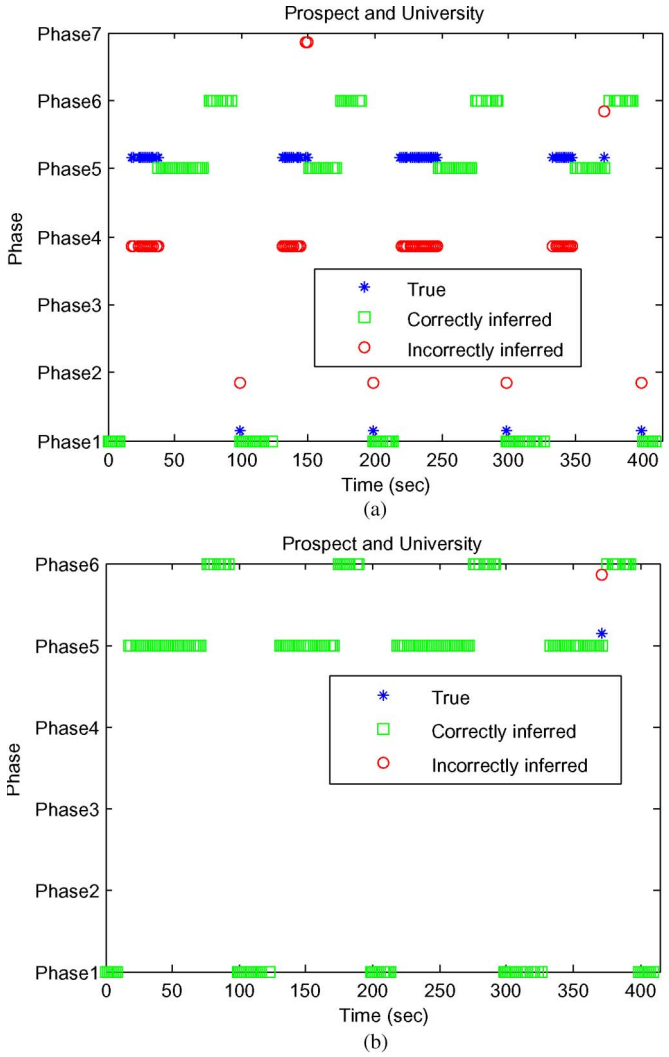

Fig. 7. Inferred phases at the intersection of W. University Ave. and Prospect Ave. with an HMM trained by (a) the Baum–Welch algorithm and (b) the Bayesian learning algorithm.

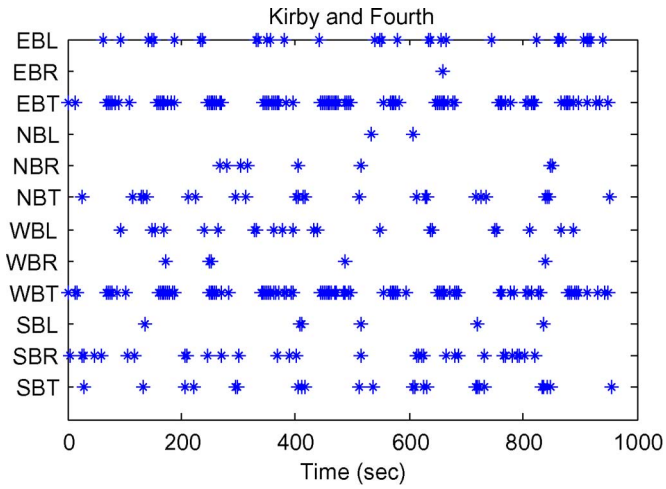

Fig. 8. Turning movement counts collected at the intersection of W. Kirby Ave. and S. Fourth St. as a function of time.

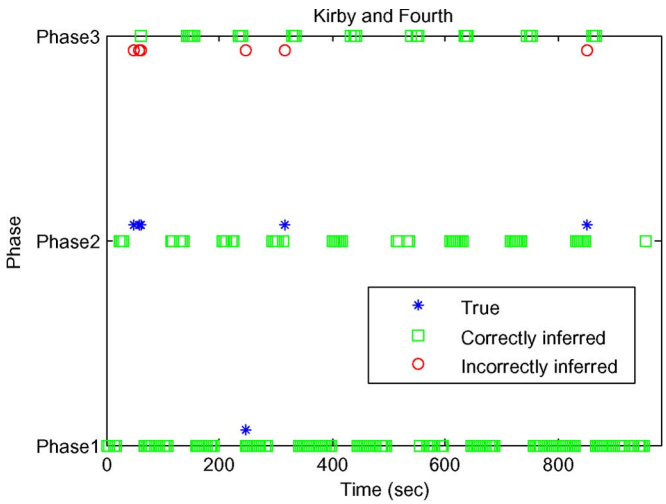

Fig. 9. Inferred phases at the intersection of W. Kirby Ave. and S. Fourth St. with an HMM trained by the Bayesian algorithm.

the observed maneuvers, and Fig. 11 depicts the true and inferred phases at this intersection. Although the data set is short and many phase switches occurred, inference with the Bayesian HMM achieves near-perfect recovery of the phases.

This example demonstrates the power of our inference approach, since the traffic at this intersection is highly complex. The signal was observed for just under 7 min, and the signal showed seven unique phases. If the traffic was extremely light or if the traffic was extremely heavy, we would likely observe a very regular signal operation, although the signal is sensor actuated.

Despite the complexity of the signal phasing at this intersection, our algorithm shows excellent performance. In fact, the only error is the incorrect classification of a left-turn maneuver at the end of the protected phase 3, which we incorrectly classified as belonging to the start phase 1, where it is also permitted.

The same set of prior parameters was used to train the initial HMM with different sets of data collected at another sensor-actuated signal in Champaign, IL, which also showed high accuracy in the inferred phases.

VI. Controller Classification

A. Description of the Classifier Based on SVMs

As shown in Sections IV and V, the Bayesian-learned HMM can achieve high inference accuracy on the phases, provided the Dirichlet prior parameters are chosen well. However, the right choice of prior parameters depends on whether the signal is fixed time or sensor actuated, which is not known in advance. To predict if the signal is fixed time or sensor actuated, we developed a simple *support vector machine* (SVM) [29] classifier that can be applied to the data set to determine what prior parameters are most appropriate. Alternatively, the hyperparameters ϑ_i, μ_{ij}, and κ_{ij} could be estimated using a Bayesian approach. However, this increases the complexity of the algorithm and may not result in more accurate inference results [30] compared with the simple classification approach.

The main idea of this classifier is that $\Pr(o|\lambda_f) > \Pr(o|\lambda_s)$ for observations generated from fixed-time signals, whereas the inequality is reversed for data from sensor-actuated signals. We set λ_s and λ_f according to (10)–(12), but with different

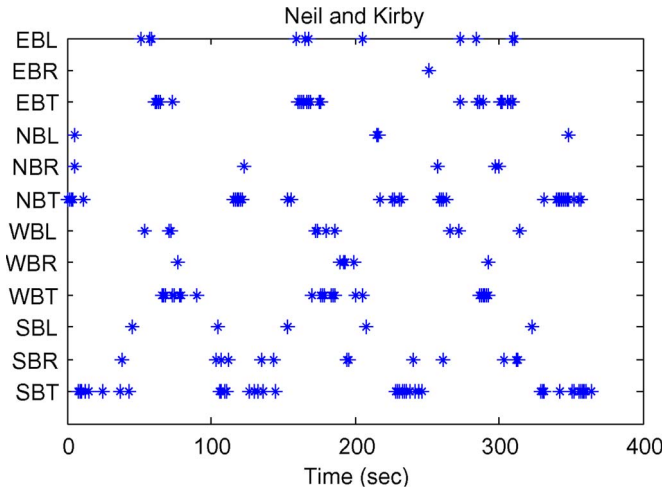

Fig. 10. Turning movement counts collected at the intersection of W. Kirby Ave. and S. Neil St. as a function of time.

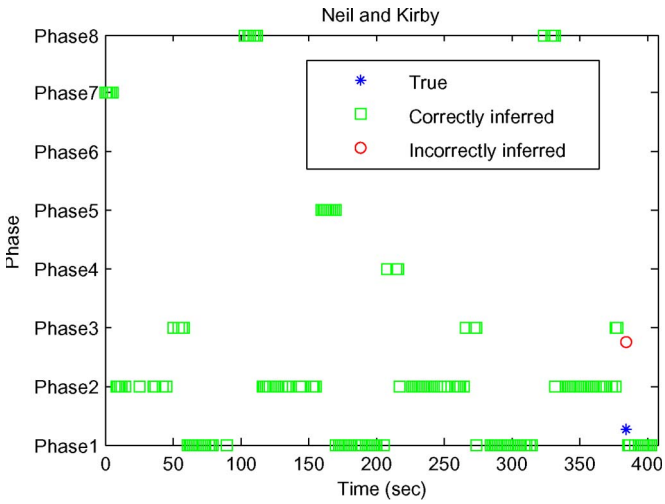

Fig. 11. Inferred phases at the intersection of W. Kirby Ave. and S. Neil St. with an HMM trained by the Bayesian algorithm.

values of the Dirichlet distribution parameters (8) for fixed-time and sensor-actuated signals. Specifically, we use $\mu_d = 20$ and $\mu_t = 1.001$ to compute λ_f, and we use the values introduced in Section IV-C (i.e., $\mu_{ii} = 50\,000$ and $\mu_{ij} = 500$ for likely transitions, $\mu_{ij} = 1.0001$ for unlikely transitions) to calculate λ_s. The remaining parameters of the Dirichlet distribution are set as described in Section IV-A and are identical for fixed-time and sensor-actuated signals.

B. Experimental Validation

We generate 400 synthetic data sets generated from 200 fixed-time signals and 200 sensor-actuated signals. We then use half of the data (100 fixed and 100 sensor-actuated data sets) to train the SVM. The remaining 200 data sets are reserved for validating the classifier.

We generate the fixed-time signal data by simulating data over a range of traffic conditions. To simulate data from an intersection with fixed-time signal, we assumed that the signal has phases p_1, p_2, and p_3 [recall Fig. 1(b)]. The total number of

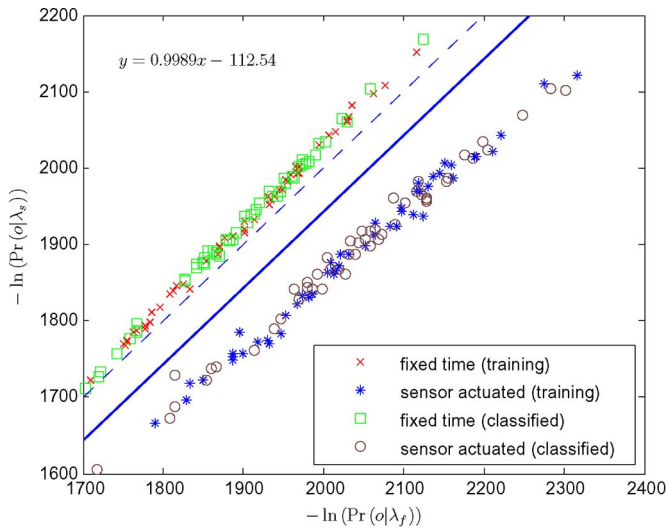

Fig. 12. Fitness of the HMM models to data obtained from fixed-time and sensor-actuated signals. The dashed line is $\ln(\Pr(o|\lambda_s)) = \ln(\Pr(o|\lambda_f))$, whereas the solid blue line is the separating hyperplane calculated by the SVM.

maneuvers observed in each phase was drawn from $\mathcal{U}\ [5, 27]$ for phases p_1 and p_2 and from $\mathcal{U}\ [2, 8]$ for phase p_3. The specific maneuvers observed in each phase are randomly drawn according to the emission probabilities given in Table II and perturbed by adding random values from $\mathcal{N}(0, 10)$ to the protected maneuvers and from $\mathcal{N}(0, 1.5)$ to permitted maneuvers except right turns. To guarantee that the probabilities sum up to one, we normalize each row.

The data sets for the sensor-actuated signal are constructed in a similar way, except we assume that the sensor-actuated signal can operate through all of the phases in Fig. 1(b) and that the phase sequence is generated based on the Markov chain illustrated in Fig. 4.

For each data set, we compute $\ln(\Pr(o|\lambda_s))$ and $\ln(\Pr(o|\lambda_f))$, which are the two features used to classify the signal. The separating hyperplane is computed, and the performance of the classifier is shown in Fig. 12. The data are linearly separable, resulting in perfect performance on the training and test data sets. While our experiments show $\Pr(o|\lambda_f) > \Pr(o|\lambda_s)$ for observations generated from fixed-time signals, and the opposite for sensor-actuated signals, the potential for misclassification errors can be reduced using the SVM instead of $\ln(\Pr(o|\lambda_s)) = \ln(\Pr(o|\lambda_f))$ as the naive separating hyperplane.

VII. Conclusion and Future Work

This paper has introduced the problem of inferring traffic signal phases from turning movement count data with the help of an HMM. Training the HMM is achieved through a Bayesian algorithm, which significantly outperforms the standard Baum–Welch algorithm. Inference is performed with the Viterbi algorithm.

The algorithms were tested on synthetic data sets and data sets collected from *TrafficTurk*, which is a smartphone-based turning movement counter. Because the experimental tests showed that different prior parameters are needed for fixed-time

and sensor-actuated signals, a simple classifier was proposed, which appears to work quite well both on numerical tests and field data.

The tests performed in this work did show that the inference accuracy is sensitive to the number of errors (specifically conflicting maneuvers) in the data set. These errors can also cause problems when trying to identify the traffic signal control logic [31] from the inferred phases. Fortunately, these errors are also fairly easy to prefilter from the data sets.

As a next step, one could also consider making the Markov chain nonhomogeneous by allowing the dwelling probability to decrease as a function of the time since the last maneuver, which might improve the robustness of the model to changes in traffic intensity. Furthermore, the algorithms developed in this work will be implemented in the *TrafficTurk* system to allow further field testing and to infer phases for users of the application.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for providing insightful comments and ideas.

REFERENCES

[1] R. Herring, A. Hofleitner, P. Abbeel, and A. Bayen, "Estimating arterial traffic conditions using sparse probe data," in *Proc. 13th IEEE Conf. Intell. Transp. Syst.*, 2010, pp. 929–936.
[2] R. Herring *et al.*, "Using mobile phones to forecast arterial traffic through statistical learning," in *Proc. 89th Transp. Res. Board Annu. Meet.*, 2010.
[3] A. Hofleitner, R. Herring, P. Abbeel, and A. Bayen, "Learning the dynamics of arterial traffic from probe data using a dynamic Bayesian network," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1679–1693, Dec. 2012.
[4] T. Park and S. Lee, "A Bayesian approach for estimating link travel time on urban arterial road network," in *Proc. ICCSA*, 2004, pp. 1017–1025.
[5] "National traffic signal report card technical report," National Transportation Operations Coalition, Washington, DC, USA, Tech. Rep. 2012, 2012.
[6] X. Ban, P. Hao, and Z. Sun, "Real time queue length estimation for signalized intersections using travel times from mobile sensors," *Transp. Res. Part C: Emerging Technol.*, vol. 19, no. 6, pp. 1133–1156, Dec. 2011.
[7] R. D. Henry, "Signal timing on a shoestring," Federal Highway Administration, Washington, DC, USA, Tech. Rep. FHWA-HOP-07-006, 2005.
[8] E. Koukoumidis, L.-S. Peh, and M. Martonosi, "SignalGuru: Leveraging mobile phones for collaborative traffic signal schedule advisory," in *Proc. 9th Int. Conf. Mobile Syst., Appl., Serv.*, 2011, pp. 353–354.
[9] T. Barkely, R. Hranac, K. Fuentes, and P. Law, "Heuristic approach for estimating arterial signal phases and progression quality from vehicle arrival data," *Transp. Res. Record: J. Transp. Res. Board*, vol. 2259, pp. 48–58, 2011.
[10] H. X. Liu, X. Wu, W. Ma, and H. Hu, "Real-time queue length estimation for congested signalized intersections," *Transp. Res. Part C: Emerging Technol.*, vol. 17, no. 4, pp. 412–427, Aug. 2009.
[11] R. Schneider, "Comparison of turning movement count data collection methods for a signal optimization study," URS Corp., Grand Rapids, MI, USA, White Paper, May 2011, White Paper.
[12] M. Reisi Gahrooei and D. Work, "Estimating traffic signal phases from turning movement counters," in *Proc. 16th IEEE Conf. Intell. Transp. Syst.*, 2013, pp. 1113–1118.
[13] J. Bonneson, S. Sunkari, and M. Pratt, "Traffic signal operations handbook," Texas Transportation Institute, College Station, TX, USA, Tech. Rep. FHWA/TX-09/0-5629-P1, 2009.
[14] OpenStreetMap. [Online]. Available: http://www.openstreetmap.org
[15] A. Bureau, S. Shiboski, and J. P. Hughes, "Applications of continuous time hidden Markov models to the study of misclassified disease outcomes," *Statist. Med.*, vol. 22, no. 3, pp. 441–462, Feb. 2003.
[16] M. Zraiaa, "Hidden Markov models: A continuous-time version of the Baum–Welch algorithm," M.S. thesis, Imperical College London, London, U.K., 2010.
[17] S.-Z. Yu, "Hidden semi-Markov models," *Artif. Intell.*, vol. 174, no. 2, pp. 215–243, Feb. 2010.
[18] L. Du, M. Chen, J. Lucas, and L. Carin, "Sticky hidden Markov modeling of comparative genomic hybridization," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5353–5368, Oct. 2010.
[19] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky, "An HDP—HMM for systems with state persistence," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 312–319.
[20] A. M. Fraser, *Hidden Markov Models and Dynamical Systems*. Philadelphia, PA, USA: SIAM, 2008.
[21] Q. Huo, C. Chan, and C. H. Lee, "Bayesian adaptive learning of parameters of hidden Markov model for speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 3, no. 5, pp. 334–345, Sep. 1995.
[22] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.
[23] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *Ann. Math. Statist.*, vol. 41, no. 1, pp. 164–171, 1970.
[24] L. E. Baum, "An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes," *Inequalities*, vol. 3, pp. 1–8, 1972.
[25] B. Frigyik, A. Kapila, and M. R. Gupta, "Introduction to the Dirichlet distribution and related processes," University of Washington Seattle, Seattle, WA, USA, Techn. Rep. UWEETR-2010-0006, 2010.
[26] C. Forbes, M. Evans, N. Hastings, and B. Peacock", *Statistical Distributions*. Hoboken, NJ, USA: Wiley, 2011.
[27] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theory*, vol. 13, no. 2, pp. 260–269, Apr. 1967.
[28] TrafficTurk, 2014. [Online]. Available: http://www.trafficturk.com
[29] N. Cristianini and J. Shawe-Taylor", *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
[30] S. Goldwater and T. Griffiths, "A fully Bayesian approach to unsupervised part-of-speech tagging," in *Proc. 45th Annu. Meet. Assoc. Comput. Linguistics*, 2007, pp. 744–751.
[31] S. Gowrishankar and D. Work, "Estimating traffic control strategies with inverse optimal control," in *Proc. 16th IEEE Conf. Intell. Transp. Syst.*, 2013, pp. 2304–2309.

Mostafa Reisi Gahrooei received M.S. degrees in transportation engineering and in applied mathematics from Southern Illinois University, Edwardsville, Edwardsville, IL, USA. He is currently working toward the Ph.D. degree in the Department of Civil Engineering, University of Illinois at Urbana-Champaign, Champaign, IL.

His research interests include traffic estimation and prediction.

Daniel B. Work received the B.S. degree from Ohio State University, Columbus, OH, USA, in 2006 and the M.S. and Ph.D. degrees from University of California, Berkeley, CA, USA, in 2007 and 2010, all in civil engineering.

He is an Assistant Professor with the Department of Civil and Environmental Engineering and the Coordinated Science Laboratory at the University of Illinois at Urbana-Champaign, Champaign, IL, USA. His research interests include control, estimation, and optimization of cyberphysical systems, mobile sensing, and inverse modeling and data assimilation, applied to problems in civil and environmental engineering.

Dr. Work received the 2014 National Science Foundation CAREER Award and the 2011 IEEE Intelligent Transportation Systems Society Best Dissertation Award.