

Estimating traffic signal phases from turning movement counters

Mostafa Reisi Gahrooei¹ and Daniel B. Work²

Abstract—This work poses the problem of estimating traffic signal phases from a sequence of maneuvers recorded from a turning movement counter. Inspired by the part-of-speech tagging problem in natural language processing, a hidden Markov model of the intersection is proposed. The model is calibrated from maneuver observations using the Baum–Welch algorithm, and the trained model is used to infer phases via the Viterbi algorithm. The approach is validated through numerical and experimental tests, which highlight that good performance can be achieved when sufficient training data is available, and when diverse maneuvers are observed during each phase. The supporting codes and data are available to download at <https://github.com/reisiga2/Estimating-phases-from-turning-movement-counts>.

I. INTRODUCTION

A. Motivation

This work is part of a larger effort to build a next generation traffic sensing platform for *extreme congestion events*, such as sporting events, political events, and natural disasters. With a few exceptions, the current state of traffic sensing infrastructure is extremely limited, especially on surface streets. The promise of GPS enabled devices has allowed a new class of statistical models of surface street traffic dynamics [5], [7], [8], [9], [15] to be developed. Even when the real-time data is sparse, it can be accumulated over longer time horizons to build good models of day-to-day surface street traffic.

Because statistical models rely strongly on historical priors when data is sparse, their performance during extreme congestion events could deteriorate significantly if sensing is limited. This is because extreme congestion events may change the network topology (e.g. due to planned road closures, or unplanned infrastructure failures), change travel demands (e.g. spikes in numbers of trips near sporting venues, storm evacuations, etc), and traffic control devices (e.g. restrictions on travel, overriding traffic signal timings by traffic control police officers).

To overcome the sensing limitation, a new sensing technology called *TrafficTurk* has been proposed to quickly and cheaply add temporary sensors on surface streets. Inspired by the 18th century human-based chess playing “machine” (and later the human based web service from Amazon) called the *Mechanical Turk*, *TrafficTurk* is a *turning movement counter* which is implemented on a smartphone (Figure

1). Users are assigned intersections through the application, and swipe gestures to indicate the movement of vehicles through the intersection. The path of a vehicle through the intersection is called a *maneuver* (e.g. *northbound left* (NBL), *eastbound right* (EBR), etc.). Because the turning movement counter is implemented on a phone, the data can be sent continuously to a central computational infrastructure for real-time processing and archival. *TrafficTurk* has been tested on a 100+ sensors (persons) deployment in Urbana–Champaign, IL to monitor traffic induced by a football game, and also in New York City following the 2012 superstorm Sandy.

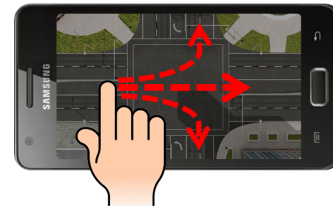


Fig. 1: *TrafficTurk* smartphone turning movement counter

Because *TrafficTurk* is intended to be used in real-time information processing systems relying on flow models, we are interested in extracting as much information as possible from the human-generated datastream. The specific problem which is the focus of this article is as follows. *What is the most probable traffic signal phase sequence which generates a sequence of observed maneuvers?* Identifying the signal phase is important to infer queue lengths (for example, is the flow zero because the light is red, or because there are no vehicles?), as well as to infer traffic signal timings which are needed in short term congestion prediction algorithms.

Since humans must go to the intersection to collect the data with *TrafficTurk*, an obvious solution to the problem is to have the users directly record the phases through the mobile phone. This approach was tested in prototype versions of *TrafficTurk*, but suffered from two major drawbacks. First, users were not able to simultaneously record traffic phases while recording maneuvers during peak hours, which meant any time spent recording phases prevented data collection on the flows, which are the primary measurement of interest. Second, the complexity of the application increased by having separate modes for phase detection and maneuver recording, which required more training when large groups of novice counters were recruited to assist in data collection.

Because of these limitations, we instead decided to infer the phases directly from the observed maneuvers algorithmically. The proposed framework is based on modeling the

¹M. Reisi Gahrooei is a Ph.D. Student in Civil and Environmental Engineering, University of Illinois at Urbana–Champaign, Urbana IL, 61801, USA. reisiga2@illinois.edu

²D. Work is an Assistant Professor in Civil and Environmental Engineering and Coordinated Science Laboratory, University of Illinois at Urbana–Champaign, Urbana IL, 61801, USA. dbwork@illinois.edu

intersection as a *hidden Markov model* (HMM). The model is first trained from the maneuver observations, and then later used to infer the most probable phase sequences. Despite the existence of several decades of data collection via classical turning movement counters, to our knowledge the problem of inferring phases from this data has not been studied. Thus, the main contribution of this article is the modeling of the phase estimation problem as an inference problem on a hidden Markov model.

B. Related work

Although signal timing plans can be obtained through municipal agencies, the difficulty of obtaining signal information at large scales has motivated interest in obtaining signal information directly by observing the intersection. In [1], GPS data from mobile phones is used to infer the queue length at an intersection and good performance can be achieved when 30% of the vehicles transmit data. In [10], mobile phones mounted in vehicles are used to identify the signal timing through imagery recorded by the phone's camera. The *SMART-SIGNAL* project [12] represents a more structured technology to unlock the data on traffic signal controllers by augmenting the control box with a wireless communications platform that can stream data from the signal in real-time. In the longer term, technologies like *SMART-SIGNAL* and ubiquitous GPS devices have the potential to circumvent ad hoc collection technologies such as *TrafficTurk* if they can be widely deployed.

The methodology proposed in this work is similar to an existing body of literature on the *part-of-speech* (POS) tagging and speech recognition problem found in natural language processing [2], [11], [13], [14]. In this problem, one wishes to obtain an estimate of the part of speech (e.g. noun, verb, etc) corresponding to each word in a document.

C. Outline of the article

In Section II, the definition of a HMM is reviewed, and a HMM model of an intersection is proposed. The learning and inference algorithms used to calibrate the HMM and infer phases from a sequence of turning movements are summarized in Section III. The performance of the algorithm is tested in a synthetic numerical experiment and also on experimental data obtained from *TrafficTurk* in Section IV.

II. PROBLEM FORMULATION

Formally, a HMM is a 5-tuple (P, V, Π, A, B) where $P = \{p_1, p_2, \dots, p_N\}$ is a set of N states, and $V = \{v_1, v_2, \dots, v_m\}$ is a set of 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 $\lambda = \{\Pi, A, B\}$.

For the phase estimation problem, a HMM for a signalized intersection is constructed as follows. The set of states P corresponds to the set of possible phases at a given

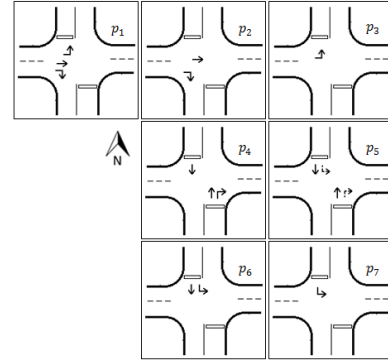


Fig. 2: Enumeration of phases for an intersection of a one-way (eastbound) street with a two-way (northbound and southbound) street.

intersection. For example, one can define the phase which allows only *eastbound through* (EBT) and *eastbound right* (EBR) maneuvers as phase p_2 , and the phase which allows only the *eastbound left* (EBL) maneuvers as phase p_3 . To complete the definition of the state space, all N possible phases for the intersection must be defined and enumerated. Figure 2 shows an enumeration of possible states for an intersection of a one-way street and a two-way street with prohibited right turn movements on a red light.

We do not assume knowledge of a particular phase sequence (i.e. a two phase cycle, a three phase cycle, etc) in advance, although this could significantly reduce the state space if it is known *a priori*.

When building the state space for the HMM at an intersection, we do assume knowledge of the number of streets as well as any one way restrictions, since this information is readily available in both commercial and open source map databases such as *OpenStreetMap* (OSM) which is used in *TrafficTurk*. We do not assume any additional knowledge about the intersection such as the number of lanes or the existence of dedicated turn lanes, as this information is not uniformly available in OSM, and it is completely absent in the Urbana-Champaign OSM data where most of our experimental tests take place. Use of commercial databases could provide further information which would assist in pruning the set of possible states.

One must also define the possible outcomes or observations $V = \{v_1, v_2, \dots, v_m\}$. In our model, the set of possible outcomes is simply the set of m maneuvers which are permitted at an intersection. For a typical four-way intersection, the total number of maneuvers is twelve (three maneuvers in each direction), while a three-way intersection has six possible outcomes. 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 state to the same state, 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 avoided probabilistically.

The final set of parameters are the emission probabilities $b_i(v_j)$ which define the probability of observing maneuver v_j when in phase p_i . In theory, these parameters could be estimated reasonably well without observational data. For example, if maneuver v_j is not permitted in phase p_i , then $b_i(v_j) = 0$. If a maneuver is permitted but not protected in a particular state, it should have a lower emission probability, and the predominant maneuvers in the phase should have higher emission probabilities.

III. HMM LEARNING AND INFERENCE ALGORITHMS

This section summarizes several well known algorithms to solve learning and inference problems on HMMs [6], [16]. 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 $O = O_1, O_2, \dots, O_k, \dots, O_K$ be a sequence of observed maneuvers, in which $O_k \in V$ is the observed maneuver at step k in the sequence. Also let $q_k \in P$ denote the state at step k . The learning problem is to estimate the parameters λ given O .

1) *Forward and backward algorithms:* The *forward algorithm* calculates the probability of a sequence of maneuvers given an set of parameters. Define

$$\alpha_k(i) = \Pr(O_1, O_2, \dots, O_k, q_k = p_i | \lambda) \quad (1)$$

to be the probability of the partial sequence of maneuvers up to step k and phase p_i appearing in step k , given the parameter set λ . The forward algorithm calculates $\Pr(O|\lambda) = \sum_{j=1}^N \alpha_K(j)$, where $\alpha_K(i)$ can be calculated as:

- 1) Initialize $\alpha_1(i) = \pi_i b_i(O_1)$.
- 2) Compute $\alpha_{k+1}(i) = b_i(O_{k+1}) \sum_{j=1}^N \alpha_k(j) a_{ji}$ for each k .

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

$$\beta_k(i) = \Pr(O_{k+1}, O_{k+2}, \dots, O_K | q_k = p_i, \lambda) \quad (2)$$

be the probability of a partial sequence of maneuvers after step k , given the parameters λ and that phase p_i appeared in step k . The backward algorithm calculates $\Pr(O|\lambda) = \sum_{j=1}^N \alpha_1(j) \beta_1(j)$, where $\beta_1(j)$ can be calculated as:

- 1) Initialize $\beta_K(i) = 1$.
- 2) Compute $\beta_k(i) = \sum_{j=1}^N \beta_{k+1}(j) a_{ij} b_j(O_{k+1})$ for each k .

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 [4] finds locally optimal parameters λ which maximize the probability of observing a sequence of maneuvers, O . 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 $\xi_k(i, j) = \Pr(q_k = p_i, q_{k+1} = p_j | O, \lambda)$ 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 .

If λ is the initial set of parameters, then updated set of parameters $\tilde{\lambda} = (\tilde{\Pi}, \tilde{A}, \tilde{B})$ given the sequence of observations and λ can be computed as follows. The learned initial state probabilities $\tilde{\Pi} = \{\tilde{\pi}_i\}$ are given by

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

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

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

and

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

In equations (3), (4), and (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 as:

$$\gamma_k(i) = \frac{\beta_k(i) \alpha_k(i)}{\sum_{i=1}^N \beta_k(i) \alpha_k(i)},$$

$$\xi_k(i, j) = \frac{\beta_{k+1}(j) \alpha_k(i) a_{ij} b_j(O_{k+1})}{\sum_{i=1}^N \sum_{j=1}^N \beta_{k+1}(j) \alpha_k(i) a_{ij} b_j(O_{k+1})}.$$

Baum et. al [3], [4] proved that the learned set of parameters $\tilde{\lambda}$ given by (3), (4), and (5) gives at least as high probability for a given sequence of observations compared to the initial parameters (e.g. $\Pr(O|\tilde{\lambda}) \geq \Pr(O|\lambda)$).

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 . It can be solved using the *Viterbi Algorithm* [17], described next.

Let

$$\delta_k(i) = \max_{q_{1:k-1}} \Pr(q_1, \dots, q_{k-1}, q_k = p_i, O_1, \dots, O_k | \lambda)$$

where $q_{1:k-1}$ denotes the sequence q_1, q_2, \dots, q_{k-1} , be the highest probability state sequence ending at p_i in step k , with observations O_1, \dots, O_k . Then by induction one can see that

$$\delta_{k+1}(j) = b_j(O_{k+1}) \max_{i=1,2,\dots,N} \{\delta_k(i) a_{ij}\} \quad (6)$$

Moreover to retrieve the optimal sequence of states, we keep track the optimal transition to state p_j at step $k+1$ through an array ψ where $\psi_{k+1}(j) = \arg \max_i \{\delta_k(i) a_{ij}\}$. The Viterbi algorithm initiates with $\delta_1(j) = \pi_j b_j(O_1)$ and $\psi_1(i) = 0$. Then, using equation (6), the terminal probability S^* can be found as $S^* = \max_j \delta_K(j)$, and the terminal phase as

$$q_K^* = \arg \max_j \delta_K(j). \quad (7)$$

Starting from the the highest probability terminal phase (7), one can backtrack the most probable sequence of phases given the observations O through ψ .

IV. NUMERICAL AND EXPERIMENTAL VALIDATION

This section explores the numerical and experimental performance of inferring traffic phase sequences from observed maneuvers through a hidden Markov model.

Both experiments have four main components: (i) obtaining traffic maneuvers, either via synthetic data generation (numerical simulation) or through the *TrafficTurk* application (experimental), (ii) initializing the HMM parameters λ , (iii) training the HMM given the observed maneuvers using the Baum–Welch algorithm, (iv) and using the trained HMM to infer the state sequence with the Viterbi algorithm.

The numerical and experimental tests are performed on an intersection shown in Figure 2. To assess the performance of the framework, several error metrics are introduced. Let K denote the length of the phase (and maneuver) sequence, and let N_m denote the total number of maneuvers which were prohibited in their predicted phase. This occurs, for example, if the collected data contains errors, and therefore can be interpreted as predicted measurement errors. Conversely, let N_p denote the number of maneuvers which are permitted in the predicted phase, but the predicted phase is incorrectly estimated. The percent prediction error can be computed as $E_m = \frac{N_m}{K} \times 100$ and $E_p = \frac{N_p}{K} \times 100$, where E_m denotes the percentage of maneuvers which were prohibited in their

Maneuver	SBL	SBT	SBR	WBR	WBL	WBT
Phase p_1 (%)	0.1	0.1	0.1	0.1	0.1	0.1
Phase p_5 (%)	14.2	35	0.1	0.1	0.1	0.1
Maneuver	NBT	NBR	NBL	EBL	EBT	EBR
Phase p_1 (%)	0.1	0.1	0.1	25	58.1	16
Phase p_5 (%)	35	15	0.1	0.1	0.1	0.1

TABLE I: Maneuver emission probabilities within each phase, used for synthetic data generation.

predicted phase, and E_p denotes the percentage of maneuvers which are permitted in the predicted phase, but the predicted phase is incorrectly estimated. It can be easily seen that $E_{total} = E_m + E_p$ is the total error in prediction.

A. Numerical tests

1) *Synthetic data generation*: To simulate traffic maneuver observations, a state sequence is constructed from alternating phases p_1 and p_5 for a fixed number of cycles, and a random sequence of maneuvers is generated within each phase. The total number of vehicles that pass through the intersection is drawn from a uniform distribution \mathcal{U} [5, 27] which ensures that every phase generates some observations. Note 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 identified according to the emission probabilities in Table I. For example, the probability of generating a SBL maneuver in phase when in phase p_5 is 35%, and all prohibited maneuvers within each phase are assigned a probability of 0.1% to simulate counting errors that occur in data collected during field experiments. It should be clarified that the maneuver observations are not generated from a HMM, because forward simulation of a HMM resulted in unrealistic (e.g. too frequent) traffic phase transitions in the simulated data.

2) *Training the HMM and estimating the state sequence*: Given the synthetic sequence of maneuvers, the parameters of the HMM are estimated with the Baum–Welch algorithm (Section III-A.2). Since the algorithm calculates locally optimal parameters, the specification of the initial probabilities is the key in achieving a good set of learned parameters. Previous experiments [16] suggest a uniform distribution on all initial probabilities Π and transition probabilities A performs well in a wide variety of applications.

As an alternative, a traffic specific set of initial probabilities was also explored. To model the fact that the signal is expected to stay in the same phase over many maneuver observations, the initial probability of remaining in the same phase a_{ii} was set at 94%, and the transition probability of moving to another state $a_{ij}, i \neq j$ was set at one percent. The initial emission probability $b_i(v)$ is set at one percent if v is a prohibited maneuver in phase p_i , and all allowed maneuvers within the phase receive a uniform initial emission probability. The initialized initial probabilities π_i are equal across all phases.

After the HMM is calibrated, a new synthetic sequence of maneuvers was generated as a test data set, using the proce-

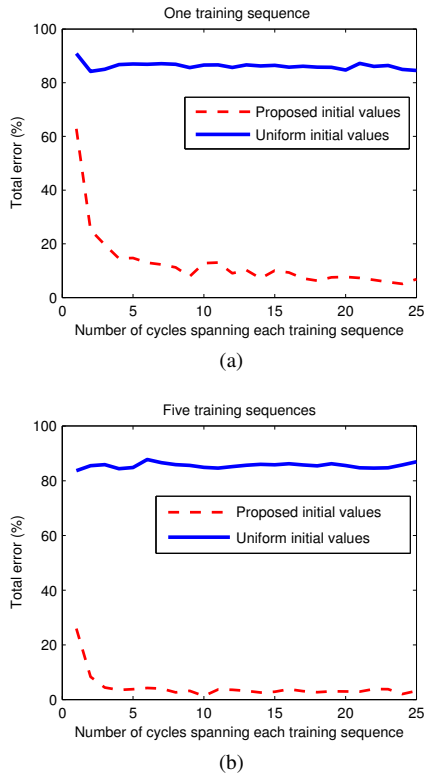


Fig. 3: Phase prediction error using a HMM trained on (a) one sequence (b) five sequences. Reported errors are averaged over 100 training and test datasets.

ture and parameters detailed for synthetic data generation. The trained HMM is used in the Viterbi algorithm to predict the phase sequence in the test dataset from the maneuver sequence.

3) *Results*: A variety of numerical experiments were run to identify the best case performance of the algorithm, as well as to identify weak points in the prediction framework.

First, the sensitivity of the initial data prediction performance was tested. The results of the numerical tests (Figure 3) highlight that the use of traffic specific initialization parameters for the Baum–Welch algorithm significantly outperforms a uniform parameter initialization independent of the training data. If the initial parameters are uniform, the prediction from the trained HMM is almost always wrong, with $E_{total} > 80\%$. When the initialization parameters are selected to strongly favor transitions that remain in the same state, and high emission probabilities of permitted maneuvers within each phase, E_{total} was reduced to about five percent.

To assess the impact of the size of the training data on the prediction accuracy, the training datasets were generated with cycle lengths varying from one to 25. In each simulation, the test data set contains a randomly generated dataset with five cycles. When the HMM is trained using only one sequence (Figure 3a), the prediction error is large when the number of cycles is small (25–62%), but reduces to about five percent as the number of cycles increases. If more sequences are

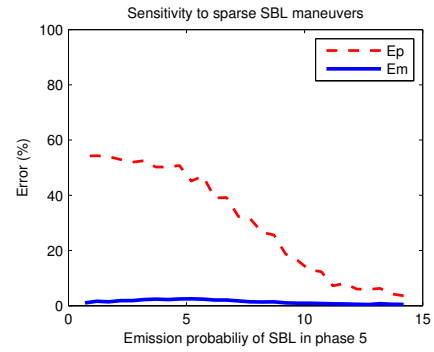


Fig. 4: Sensitivity of phase prediction errors to sparse SBL turning movements in phase 5. Reported errors are averaged over 100 training and test datasets.

used in the training data (Figure 3b) a lower prediction error can be achieved (three percent). The results suggest that a good set of training data should contain sufficient cycles to estimate the state transitions, and enough sequences to correctly estimate the initial probabilities.

Finally, to investigate the sensitivity of the algorithm to sparse turning movements within a phase, we reduced the emission probability of the *southbound left* (SBL) maneuver in phase p_5 in the data generation process. The problem is challenging because phase p_5 is nearly identical to phase p_4 as the number of SBL maneuvers in p_5 decreases. When the emission probability of SBL maneuvers drops to zero, the two phases admit exactly the same allowed maneuvers, which would make correct identification of the phase from maneuvers nearly impossible. For each emission probability shown in Figure (4), 20 sequences of training data, each with 15 cycles is used to train a HMM, and predict the phases of a test dataset with 5 cycles. The results are then averaged over 100 simulations and shown in Figure 4. Because of the large training dataset, the phase prediction errors are very small when the emission probability of the SBL approaches 15%. While the E_m error remains small across all emission probabilities, the E_p grows to nearly 60% when only few SBL maneuvers are present. This is because the vast majority of incorrectly predicted phases are predictions of phase p_4 , when the true phase is p_5 .

B. Experimental performance

To evaluate the experimental performance of the HMM approach to phase estimation, turning movements were recorded using the *TrafficTurk* smartphone turning movement counter (Figure 1) at the intersection of W. University Ave. and Prospect Ave. in Champaign, IL. The intersection layout is shown in Figure 2. The turning movements and true phases were simultaneously recorded over two 10 min intervals, with 388 movements and 356 movements respectively over five cycles. The phase sequence is p_1, p_5, p_6 during each cycle.

Figure 5a shows the predicted and actual phases corresponding to the maneuvers during the first two minute interval of data collection, and illustrates the main features

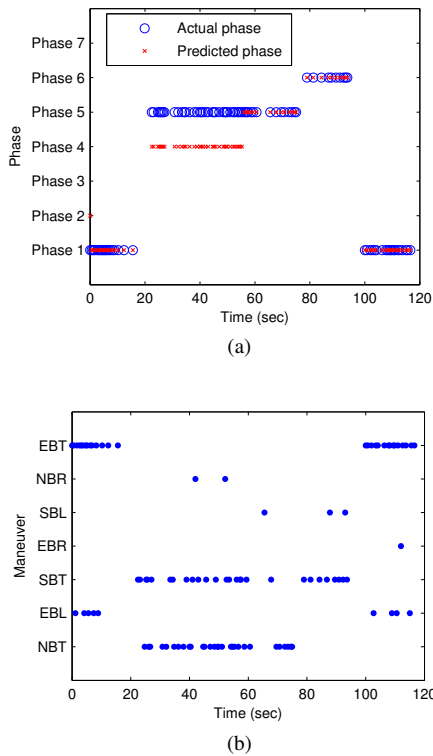


Fig. 5: (a) Actual and predicted phases over first 120 seconds
(b) Distribution of maneuvers over first 120 seconds

of the prediction. With the exception of one prediction error at time 0, all errors in the two minute interval are errors between phase p_4 and phase p_5 . Over the full ten minute test dataset, 137 of the 147 prediction errors were caused by errors between p_4 and p_5 . The total prediction error E_{total} is 41%. Figure 5b depicts the distribution of maneuvers over the same time interval, and highlights the sparsity of the SBL and NBR maneuvers. The percentage of SBL maneuvers compared to all maneuvers in phase p_5 is less than 2.5% throughout the dataset. Based on the numerical experiments with low SBL emission probabilities, it is expected that a larger training data set alone is not sufficient to overcome incorrectly labeling these similar phases.

V. CONCLUSIONS AND FUTURE WORK

This work presented an approach to estimate traffic signal phases from turning movement counts. The proposed approach involved training a hidden Markov model from the observed sequence of maneuvers, and the trained HMM is then used to infer the most probable phase sequence. Numerical experiments show that good performance can be achieved with sufficient training data that has sufficient diversity of maneuvers observed within each phase. Experimental results suggest the lack of maneuver diversity within each phase presents a practical challenge to the proposed approach. This may be addressed in ongoing work by further constraining the HMM state transitions used for inference.

VI. ACKNOWLEDGEMENT

The authors would like to thank S. Gowrishankar for generating Figure 1, and B. Sura, J. Que, C. Vega, and M. Carvajal for developing the *TrafficTurk* mobile application.

REFERENCES

- [1] X. Ban, P. Hao, and Z. Sun. Real time queue length estimation for signalized intersections using travel times from mobile sensors. *Transportation Research Part C: Emerging Technologies*, 19(6):1133–1156, 2011.
- [2] M. Banko and R. C. Moore. Part of speech tagging in context. In *Proceedings of the 20th International Conference on Computational Linguistics*, number 556, Stroudsburg, PA, USA, 2004.
- [3] L. E. Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. *Inequalities*, 3:1–8, 1972.
- [4] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The annals of mathematical statistics*, 41(1):164–171, 1970.
- [5] C. de Fabritiis, R. Ragona, and G. Valenti. Traffic estimation and prediction based on real time floating car data. In *11th International IEEE Conference on Intelligent Transportation Systems*, pages 197–203, 2008.
- [6] A. M. Fraser. *Hidden Markov models and dynamical systems*. SIAM, 2008.
- [7] R. Herring, A. Hofleitner, P. Abbeel, and A. Bayen. Estimating arterial traffic conditions using sparse probe data. In *13th International IEEE Conference on Intelligent Transportation Systems*, 2010.
- [8] R. Herring, A. Hofleitner, S. Amin, T. Abou Nasr, A. A. Khalek, P. Abbeel, and A. Bayen. Using mobile phones to forecast arterial traffic through statistical learning. In *89th Transportation Research Board Annual Meeting*, 2010.
- [9] A. Hofleitner, R. Herring, P. Abbeel, and A. Bayen. Learning the dynamics of arterial traffic from probe data using a dynamic bayesian network. *IEEE Transactions on Intelligent Transportation Systems*, 13(4):1679–1693, 2012.
- [10] E. Koukoumidis, L.-S. Peh, and M. Martonosi. SignalGuru: leveraging mobile phones for collaborative traffic signal schedule advisory. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, pages 353–354, 2011.
- [11] J. Kupiec. Robust part-of-speech tagging using a hidden Markov model. *Computer Speech and Language*, 6(3):225 – 242, 1992.
- [12] H. X. Liu, X. Wu, W. Ma, and H. Hu. Real-time queue length estimation for congested signalized intersections. *Transportation Research Part C: Emerging Technologies*, 17(4):412–427, 2009.
- [13] B. Merialdo. Tagging English text with a probabilistic model. *Computational Linguistics*, 20(2):155–171, 1994.
- [14] N. Mohammadiha, R. Martin, and A. Leijon. Spectral domain speech enhancement using hmm state-dependent super-gaussian priors. *IEEE Signal Process*, 20(3):253–256, 2013.
- [15] T. Park and S. Lee. A Bayesian approach for estimating link travel time on urban arterial road network. In *Computational Science and its Applications ICCSA*, pages 1017–1025, 2004.
- [16] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of The IEEE*, 77(2):257–286, 1989.
- [17] A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967.