City-Wide Traffic Flow Estimation From a Limited Number of Low-Quality Cameras

Tsuyoshi Idé, Takayuki Katsuki, Tetsuro Morimura, and Robert Morris, *Fellow, IEEE*

***Abstract*—We present a new approach to lightweight intelligent transportation systems. Our approach does not rely on traditional expensive infrastructures, but rather on advanced machine learning algorithms. It takes images from traffic cameras at a limited number of locations and estimates the traffic over the entire road network. Our approach features two main algorithms. The first is a probabilistic vehicle counting algorithm from low-quality images that falls into the category of unsupervised learning. The other is a network inference algorithm based on an inverse Markov chain formulation that infers the traffic at arbitrary links from a limited number of observations. We evaluated our approach on two different traffic data sets, one acquired in Nairobi, Kenya, and the other in Kyoto, Japan.**

***Index Terms*—Gaussian mixtures, image analysis, inverse Markov problem, object counting, variational Bayes.**

I. Introduction

TRAFFIC congestion is a major problem in the urban regions of most developing countries, where mismatches are found between rapidly growing economies and the municipal infrastructures. Intelligent transportation systems (ITS) provide a basic framework for traffic management. Unlike urban areas in relatively mature countries, cities with rapid economic growth require a lightweight ITS to adapt to the dynamically changing environment.

What we are interested in here is a "Frugal" approach [1] to ITS. Instead of relying heavily on expensive infrastructures such as an inductive-loop sensor system covering an entire city area for monitoring traffic, we wish to develop an ITS that is easy to deploy, has a minimum entry cost, and offers good enough functionalities. As an alternative to the existing full-scale systems, we focus on a webcam-based monitoring approach. Webcam-based traffic surveillance through web browsers is already being used in many cities in developing countries. For instance, in Nairobi, Kenya, AccessKenya.com [2] runs a web site to provide near real-time information on traffic at major locations. Although simply looking at the webcam images through web browsers is useful enough for personal use, this is not the case for traffic authorities. For the purpose of city planning and traffic optimization, we need to extract key information on traffic flows from webcam images for the entire city. This is the main topic of this paper.

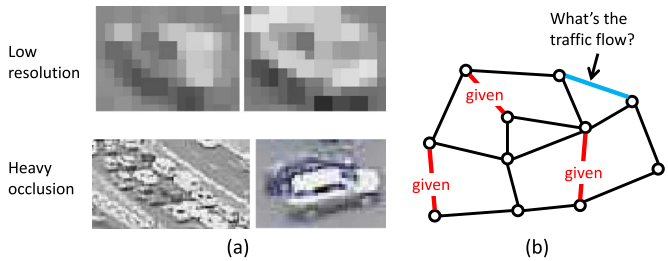


Fig. 1. Challenges in webcam-based ITSs. (a) Very low quality images. (b) Partial observations.

A lot of research has gone into making camera-based traffic monitoring truly useful. Examples include vehicle recognition for traffic volume estimation [3]–[6] and regression modeling for vehicle counting [7], [8]. In addition, origin-destination (OD) matrix estimation algorithms [9]–[12] are often combined with traffic estimation methods since the number of cameras is always limited [13]. Although these studies have made significant contributions to resolving a number of technical issues, major challenges still remain, as summarized below.

The first challenge is how to handle very low-quality images [see Fig. 1(a) for example]. Due to cost and antitheft concerns, the use of special-purpose close-view cameras is impractical in most developing countries. On the other hand, resorting to general-purpose cameras without purpose-built lighting facilities impairs standard object recognition technologies such as those used in number plate recognition [6], [14].

The second challenge is how to eliminate the time-consuming step of camera-wise calibration in the image processing. Most of the recent studies on video-based ITS focus on calibration algorithms when surveillance cameras do not allow calibration on the hardware side [3]–[6]. In either case, however, as long as vehicle recognition is performed on images, camera-wise fine adjustments based on the geometric configuration of cameras and roads are required. Although the use of regression models [7], [8] can reduce the burden, a fair amount of *labeled* training data (i.e. manually counted or recognized images) is still required.

The third challenge is how to derive city-wide information from a limited number of webcams [see Fig. 1(b)]. In particular, what-if simulations for optimized city planning call for estimates of the traffic volume in every single link of the road network. This task is similar to network tomography for the OD matrix [3]–[6], but differs in that it needs to infer the traffic volume at all of the links instead of just the origin-destination flows.

Manuscript received February 16, 2016; revised May 24, 2016; accepted July 21, 2016. Date of publication August 18, 2016; date of current version March 27, 2017. The Associate Editor for this paper was S. Sun.

T. Idé is with the IBM Thomas J. Watson Research Center, IBM Research, Yorktown Heights, NY 10598 USA (e-mail: tide@us.ibm.com).

T. Katsuki and T. Morimura are with the Analytics & Optimization Group, IBM Research—Tokyo, Tokyo 103-8510, Japan.

R. Morris is with Global Laboratories, IBM Research, Singapore 486072.

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.2016.2597160

We tackle these challenges using novel machine learning techniques. Our technical contributions are as follows:

- We developed a novel algorithm for fully calibration-free vehicle counting. One prominent feature of our method (see Section III) is that it requires neither camera-wise calibration nor labeled data generation.
- We developed a new inference algorithm on road networks to estimate the traffic volume at arbitrary links without direct observation by webcams. We formalize the problem as an inverse Markov chain problem, and leverage the mathematical technique of regularization (see Section IV).

These methods are validated with real webcam traffic images from Nairobi, Kenya, as well as simulated traffic data for Kyoto, Japan, as elaborated in Section V.

These methods were already outlined in a preliminary version of this paper [15]. This paper significantly expands on the earlier one by adding algorithmic and experimental details, based on our companion papers [16]–[18] that focus more on theoretical aspects of the vehicle counting and network inference problems.

II. Related Work

This section reviews related work with an emphasis on the task of image-based traffic estimation and network-wide traffic estimation.

A. Image-Based Traffic Estimation

A popular research topic in the ITS research community is how to replace traditional expensive data acquisition infrastructures with lower cost alternative methods. Two alternative methods have mainly been studied to date: GPS (global positioning system) and surveillance cameras.

GPS is a powerful tool for gathering traffic-related information over an extended area and it has attracted a lot of recent attention from the ITS community. Recent studies include those of Fabritiis *et al.* [19], Kong *et al.* [20], and Shan and Zhu [21], which mainly addresses the issue of how to accurately infer real traffic information from trajectories of GPS signals that are known to be quite noisy. In addition to the quality issue, a serious problem is that GPS data is *proprietary* to the companies owning the communication infrastructures. This strongly motivates traffic authorities and companies not having first-hand access to GPS devices to go after the other alternative.

The use of surveillance cameras thus can be viewed as the mainstream of lightweight ITS development. The earliest work on traffic monitoring based on network-connected cameras includes the studies of Santini [13], Yu *et al.* [22], and Huck *et al.* [23]; these studies did not address the issues of low quality or network reconstruction from partial observations.

In image-based traffic monitoring, the issue of calibration is a major research topic. Roughly speaking, it is to ascertain the typical size of vehicles in images by considering the distance and angles to the location being monitored. Cathey and Dailey [3] proposed a sophisticated algorithm to estimate traffic speed based on cross-correlation without calibrations in the camera. Tian *et al.* [5] and Buch *et al.* [6] described how to recognize vehicles in low-quality images. Robert [4] attempted to enhance the accuracy of vehicle recognition by incorporating machine learning algorithms such as support vector machines. All these studies are based on the strategy of individual vehicle recognition, which is not feasible with low-quality images from webcams.

Another important issue with traditional image-based traffic analysis in practice is the cost of preparing properly labeled images. Although a recently proposed regression-based approach [7] removed the image recognition step and reduced the task to supervised learning, it still requires a fair amount of manually labeled training data.

B. Network-Wide Traffic Estimation

Since the seminal work of Zuylen and Willumsen [9], the task of network-level traffic estimation from partial observations has attracted a lot of attention. Most of the previous efforts have focused on the task of OD matrix estimation [10], [11], wherein the following two approaches have been mainly studied.

The first approach is to minimize an error function between the observed and estimated traffic volumes while satisfying flow conservation conditions. Recent work along this line includes Shao *et al.* [12] and Hu *et al.* [24]. Although our approach shares some of their ideas, it can be distinguished from their work in that our goal is to estimate not the OD matrix, but the traffic volume at arbitrary links. Recently, Menon *et al.* [25] introduced an interesting algorithm for sparse OD matrix estimation using an L_1 regularization technique. Their problem, as well as optimization strategy, differs from ours, as can be clearly seen from the fact that the "seed" OD matrix is not available in our problem.

The other major approach is to use Bayesian networks, where graphical Gaussian models (GGMs) [26] are commonly used. Examples include Zhang *et al.* [27], Sun *et al.* [28], and Zhu *et al.* [29]. Although these approaches are built upon the well-grounded theory of GGMs, one of the practical disadvantages is the lack of scalability. The global Gaussian assumption is hard to apply to large networks because of its computational cost and numerical instability [30]; thus, it is inappropriate for our city-wide monitoring scenario.

III. Calibration-Free Vehicle Counting

This section presents an approach to calibration-free low-quality image analysis for counting vehicles. As illustrated in Fig. 2, the approach essentially consists of two steps: feature extraction and counting. Note that since the webcams are analyzed independently, we will focus on images from a single webcam in the rest of this section.

A. Feature Extraction

Let N be the number of training images for the camera we are focusing on. Assume that all the images have the same M pixels, and each of the pixels takes an integer from the 256 luminance levels. Our data set is represented as

$$\mathcal{D}_0 \equiv \{\boldsymbol{z}^{(n)} \in \{0,1,2,\ldots,255\}^M | n = 1,\ldots,N\}. \quad (1)$$

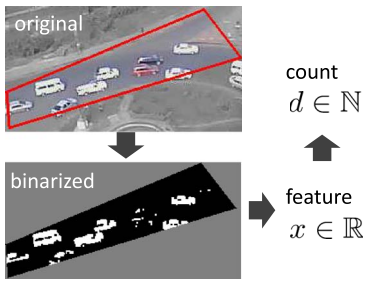


Fig. 2. Vehicle counting from low-quality images. Symbols $\mathbb{R}$ and $\mathbb{N}$ denote the real and natural numbers (nonnegative integers), respectively.

For each image, as a preprocess, we subtract the median over the M pixels to handle variations, e.g., between nighttime and daytime. The goal of the step of feature extraction is to extract a feature $x \in \mathbb{R}$ from a raw image $\boldsymbol{z} \in \{0, \ldots, 255\}^M$ such that x corresponds to a rough estimate of the count of vehicles.

As indicated in Fig. 2, our approach first converts the original image into a binarized one. After that, the feature x is computed as the ratio of white pixels to the total number of pixels

$$x = \frac{1}{M} \sum_{i=1}^{M} I(z_i \geq k^*) \tag{2}$$

where k^* is the threshold for binarization, and $I(\cdot)$ is the indicator function that gives 1 when the argument is true, and 0 otherwise.

To find the threshold k^*, we follow Otsu's method [31]. In the training data $\mathcal{D}_0$, we define

$$p_l \equiv \frac{1}{MN} \sum_{n=1}^{N} \sum_{i=1}^{M} I\left(z_i^{(n)} = l\right) \tag{3}$$

which can be thought of as the probability of the pixel taking a luminance value ℓ. With this probability, we can compute the mean luminances as $\bar{\ell} \equiv \sum_{l=0}^{255} p_l l$. Similarly, the mean luminances for the black class and the white class are respectively

$$\ell_1(k) \equiv \frac{1}{P_1(k)} \sum_{l=0}^{k-1} p_l l, \quad \ell_2(k) \equiv \frac{1}{P_2(k)} \sum_{l=k}^{255} p_l l \tag{4}$$

where $P_1(k) \equiv \sum_{l=0}^{k-1} p_l$ and $P_2(k) \equiv \sum_{l=k}^{255} p_l$. Obviously, these are functions of the threshold k. The optimal threshold k^* is determined by solving the following optimization problem,

$$k^* = \arg\max_k \left\{ \left[\ell_1(k) - \bar{\ell}\right]^2 P_1(k) + \left[\ell_2(k) - \bar{\ell}\right]^2 P_2(k) \right\}. \tag{5}$$

This problem can be simply solved by evaluating the objective function for all of the 256 different values and picking the one giving the maximum.

B. Probabilistic Counting Framework

Given the optimized threshold k^*, the training data $\mathcal{D}_0$ is now converted into

$$\mathcal{D} \equiv \{x^{(n)} \in \mathbb{R} | n = 1, \ldots, N\}. \tag{6}$$

In practice, it is recommended to further standardize the feature as $x^{(n)} \leftarrow 2[x^{(n)} / \max_{n'} x^{(n')}] - 1$ before model fitting. As discussed in the Introduction, in the city-wide traffic monitoring scenario using low-quality cameras, the task of collecting manually counted images for each camera is quite costly. Here, we propose a fully *unsupervised* approach for counting vehicles.

The vehicle counting part in Fig. 2 consists of two sub-steps. *First*, we find the predictive distribution for the feature, x, in the form of a Gaussian mixture model

$$p(x|\mathcal{D}) = \sum_{d=0}^{D} \pi_d(x) \mathcal{N}\left(x | \boldsymbol{m}^\top \boldsymbol{\phi}_d, \sigma_d^2\right) \tag{7}$$

where d denotes the number of vehicles, $^\top$ is the transpose, $\boldsymbol{\phi}_d \equiv (1, d)^\top$, and $\mathcal{N}$ denotes the Gaussian distribution (see Appendix A for the explicit definition). The function $\pi_d(x)$, which is often called the gating function, and the coefficients $\boldsymbol{m}$ are to be determined from the data. The number of mixtures D is treated as a given constant, and fixed as $D = N$ hereafter to make it large enough. As derived in Appendix B, the variance σ_d^2 is given as a function of other model parameters a, b, and Σ as

$$\sigma_d^2 = \frac{b}{a-1} + \boldsymbol{\phi}_d^\top \Sigma \boldsymbol{\phi}_d. \tag{8}$$

Second, once the predictive model (7) has been learned, finding the count d' is trivial. For a new observation $x = x'$, the corresponding vehicle count d' is given by

$$d' = \arg\max_d \left\{ \pi_d(x') \mathcal{N}\left(x' | \boldsymbol{m}^\top \boldsymbol{\phi}_d, \sigma_d^2\right) \right\}. \tag{9}$$

As a result of Bayesian learning, mixture components irrelevant to the data are automatically removed from the model. As illustrated later in Fig. 4, the max function evaluates the one-dimensional function only some ten times, which is negligible in terms of computational cost. It can work in real-time upon image refresh, which typically happens every few seconds (4 seconds in the case of AccessKenya.com [2]).

Learning the predictive model itself is actually an advanced task. The major challenge is in how to handle the interchangeability of cluster labels in the mixture model. We will tackle it by introducing a certain prior distribution. Luckily, the resulting equations for finding the model parameters involve only simple matrix-vector operations, and are extremely easy to implement as shown in Algorithm 1. The details are explained in Appendix B.

Algorithm 1 Unsupervised counting model training

procedure MIXPARAM $\boldsymbol{m}_0, \Sigma_0, a_0, b_0, \beta, D$
  Initialize as $\pi_d^{(n)} = 1/D, a = a_0 + N/2, \boldsymbol{m} = \boldsymbol{m}_0, \Sigma = \Sigma_0$
  repeat
    for $d \leftarrow 1, D$ **do**
      $N_d = \sum_{n=1}^{N} \pi_d^{(n)}, \bar{x}_d = \sum_{n=1}^{N} \pi_d^{(n)} x^{(n)}$
      for $n \leftarrow 1, N$ **do**
        $\Delta_d^{(n)} = (x^{(n)} - \boldsymbol{\phi}_d^\top \boldsymbol{m})^2 + \boldsymbol{\phi}_d^\top \Sigma \boldsymbol{\phi}_d$
      end for

$\alpha_d = 1 + N_d, \beta_d = \beta + \sum_{k=d+1}^{D} \sum_{n=1}^{N} \pi_d^{(n)}$
end for
$(b, \Sigma, \boldsymbol{m}) \leftarrow$ (Eq. (44), (45), (46))
for $n \leftarrow 1, N$ **do**
for $d \leftarrow 1, D$ **do**
$\ln \pi_d^{(n)} \leftarrow$ (Eq. (48))
end for
$\pi_d^{(n)} \leftarrow \pi_d^{(n)} / \sum_{l=0}^{D} \pi_l^{(n)}$
end for
until Convergence
return $\boldsymbol{m}, \Sigma, a, b, \{\alpha_d, \beta_d\}$.
end procedure

IV. Network Flow Estimation

This section presents an algorithm for estimating the traffic volume at arbitrary links of the network, given the observed traffic volume at a limited number of the links, as illustrated in Fig. 1(b).

A. Inverse Markov Chain Problem

We formalize this problem as an inverse Markov chain problem: Given the traffic volumes at a limited number of the links, find the Markov transition probability $p(i|j)$, which is defined as the transition probability from an arbitrary link j to another arbitrary link i.

We will assume the Markov chain is irreducible meaning that there are no completely isolated areas in the map and any link is reachable in a finite number of hops from any other link. Any irreducible Markov chain has a stationary distribution. Let the stationary distribution of this Markov chain be $s(i), i = 1, \ldots, L$, where L denotes the total number of links in the network. Our fundamental assumption is that the observed traffic volume is proportional to $s(i)$ up to a measurement error:

$$y(i) = cs(i) \quad \forall i \in \mathcal{C} \tag{10}$$

where $\mathcal{C}$ is the set of links directly monitored by webcams, $y(i)$ denotes the observed traffic volume for the i-th link (typically estimated from the approach in the previous section), and c is an unknown constant to be determined. Obviously, p and s satisfy

$$s(i) = \sum_{j=1}^{L} p(i|j)s(j). \tag{11}$$

In the matrix form, this equation is written as $\mathsf{P}\boldsymbol{s} = \boldsymbol{s}$ in the obvious notation. This means that the stationary state is computed as the eigenvector of P having the eigenvalue of 1.

B. Parameterizing the Transition Model

We parameterize the probability distribution $p(i|j)$ as

$$p(i|j) = (1 - \gamma)q(i|j; \boldsymbol{u}) + \gamma r(i; \boldsymbol{w}) \tag{12}$$

where γ is called the restart probability (assumed to be a fixed parameter), and $\boldsymbol{u}$ and $\boldsymbol{w}$ are the model parameters to be learned. In this decomposition, r is interpreted as the initial probability distribution over the links, while q is interpreted as the "partial" transition probability distribution. This type of decomposition is natural for traffic analysis on road networks because it is consistent with a typical data generation process in traffic simulation. Specifically, when we generate traffic data using a multi-agent simulator [32], we first generate the starting locations and then generate paths obeying a given transition rule.

TABLE I
Road Type Weight h_t for (15)

motorway	1.5	secondary	0.3
motorway_link	1.3	secondary_link	0.1
trunk	1.1	tertiary	−0.1
trunk_link	0.9	tertiary_link	−0.3
primary	0.7	unclassified	−0.5
primary_link	0.5	other	−0.7

For r and q, we use the following particular forms:

$$r(i; \boldsymbol{w}) \propto \exp(w_i) \tag{13}$$

$$q(i|j; \boldsymbol{u}) \propto I(i \sim j) \exp\left[g(\boldsymbol{u})\right] \tag{14}$$

$$g(\boldsymbol{u}) \equiv u_{i,j} + u_0 \cos(i|j) + u_1 h_{\text{type}(i)} \tag{15}$$

where $i \sim j$ represents that the i-th link is directly connected to the j-th link. The indicator function $I(i \sim j)$ returns 1 if directly connected, 0 otherwise. Unlike the conventional approach, which imposes the traffic conservation constraint directly on the flow itself, we take account of the conservation law *probabilistically*. Once the transition probability is found, the normalization condition of the transition probability automatically guarantees the conservation law in the expectation.

In Eq. (15), $\cos(i|j)$ is the cosine of the geometric angle between the i-th and j-th links. For example, if the j-th link points in the opposite direction to the i-th link, $\cos(i|j) = -1$ and the transition probability between them is down-weighted. In the last term of Eq. (15), the function type(i) returns the road type index for the i-th link. On the basis of the link attributes available in a digital road map, we use 14 road types (see Table I in Section V-B), and each one is weighted differently with $h_t (t = 1, \ldots, 14)$.

Combining Eqs. (12)–(14), we obtain the stationary distribution $s(i)$ as a function of the model parameters $\boldsymbol{w} = [w_0, \ldots, w_L]^\top$ and $\boldsymbol{u} = [u_0, u_1, u_{1,1}, \ldots, u_{L,L}]^\top$, where $u_{i,j}$'s for unconnected pairs are omitted. Optimal model parameters are those that minimize the discrepancies between the left- and right-hand sides of Eq. (10). The next section looks at how to measure the discrepancy.

C. Designing the Objective Function

Now that we have introduced $s(i)$ as a probability distribution, Eq. (10) can be viewed as a relationship between two distributions. The most natural discrepancy measure for distributions is the Kullback-Leibler (KL) divergence [33], which has been used in a number of traffic estimation problems [25], [34], [35]. The KL divergence can be interpreted as the expectation of information loss. Let us define $\rho(i)$ by

$$\rho(i) \equiv \ln \frac{cs(i)}{y(i)}. \tag{16}$$

$\rho(i)$ represents the local information loss at link i. Since we are interested in the stationary state, the information loss on the network should be as uniform as possible. If there is a large loss at a particular link, it will be dissipated through the transition process. With this intuition in mind, we define the error function to be minimized as the variance of the information loss

$$L(\boldsymbol{u}, \boldsymbol{w}) \equiv \frac{1}{|\mathcal{C}|} \sum_{i \in \mathcal{C}} [\rho(i) - \bar{\rho}]^2 \tag{17}$$

where $|\mathcal{C}|$ is the number of links directly monitored by the webcams, and $\bar{\rho}$ is the mean of the information loss defined by. $\bar{\rho} 1/|\mathcal{C}| \sum_{i \in \mathcal{C}} \rho(i)$ Using the definition of ρ, after some algebra, we obtain the final expression of the error function as

$$L(\boldsymbol{u}, \boldsymbol{w}) = \frac{1}{2|\mathcal{C}|^2} \sum_{i,j \in \mathcal{C}} \left[\ln \frac{cs(i|\boldsymbol{u}, \boldsymbol{w})}{y(i)} - \ln \frac{cs(j|\boldsymbol{u}, \boldsymbol{w})}{y(j)} \right]^2 \tag{18}$$

where we explicitly represented the dependency on the model parameters $\boldsymbol{u}$ and $\boldsymbol{w}$ in $s(i)$ and $s(j)$. We can see that the unknown c is canceled in this objective.

In addition, we impose an elastic net type regularization [36] on the model parameters

$$R_1(\boldsymbol{u}, \boldsymbol{w}) \equiv |u_0| + |u_1| + \sum_{i \sim j} |u_{i,j}| + \sum_{i=1}^{L} |w_i| \tag{19}$$

$$R_2(\boldsymbol{u}, \boldsymbol{w}) \equiv u_0^2 + u_1^2 + \sum_{i \sim j} u_{i,j}^2 + \sum_{i=1}^{L} w_i^2. \tag{20}$$

Putting this all together, we arrive at the final objective function to be minimized

$$Q(\boldsymbol{u}, \boldsymbol{w}) \equiv L(\boldsymbol{u}, \boldsymbol{w}) + \lambda_1 R_1(\boldsymbol{u}, \boldsymbol{w}) + \lambda_2 R_2(\boldsymbol{u}, \boldsymbol{w}) \tag{21}$$

where the new parameters λ_1 and λ_2 control the tradeoff between the error function and the regularization terms, and are determined through cross validation.

D. Solving the Optimization Problem

The objective function $Q(\boldsymbol{u}, \boldsymbol{w})$ can be minimized with the gradient method. One challenge here is that the dependency of s on $\boldsymbol{u}, \boldsymbol{w}$ is not explicitly given. Fortunately, the use of the natural gradient [37] allows us to efficiently compute the gradient without an explicit function $s(i|\boldsymbol{u}, \boldsymbol{w})$. For the detail, see a companion paper [17].

Once the minimizer $\boldsymbol{u}^*$, $\boldsymbol{w}^*$ of $Q(\boldsymbol{u}, \boldsymbol{w})$ is found, one can determine an optimal c (denoted by c^*) by solving the least squares problem

$$c^* = \arg\min_c \sum_{j \in \mathcal{C}} [y(j) - cs(j|\boldsymbol{u}^*, \boldsymbol{w}^*)]^2 \tag{22}$$

to get the solution

$$c^* = \left[\sum_{i \in \mathcal{C}} s(i|\boldsymbol{u}^*, \boldsymbol{w}^*)^2 \right]^{-1} \sum_{j \in \mathcal{C}} y(j) s(j|\boldsymbol{u}^*, \boldsymbol{w}^*). \tag{23}$$

Fig. 3. Examples of original webcam images in Nairobi [2].

The network inference algorithm is summarized in Algorithm 2.

Algorithm 2 Inverse Markov network traffic inference

procedure INVERSEMARKOV ($\{y(i) \mid i \in \mathcal{C}\}, \gamma, \lambda_1, \lambda_2$).
  $(\boldsymbol{u}^*, \boldsymbol{w}^*) = \arg\min_{\boldsymbol{u}, \boldsymbol{w}} Q(\boldsymbol{u}, \boldsymbol{w})$.
  Find the transition probability matrix P using Eq. (12).
  Find the stationary distribution $\boldsymbol{s}$ by solving $\mathsf{P}\boldsymbol{s} = \boldsymbol{s}$.
  Compute c^* using Eq. (23).
  Compute $\hat{\boldsymbol{y}} = c^* \boldsymbol{s}$.
return Estimated traffic flow $\{\hat{y}(i)\}$ at all the links.
end procedure

V. EXPERIMENTS

This section describes experiments using two traffic data sets: *Nairobi* from Nairobi, Kenya, and *Kyoto* from Kyoto, Japan.

A. Vehicle Counting: Nairobi Data

We tested the vehicle counting algorithm with images taken by traffic webcams in Nairobi, provided by AccessKenya.com [2]. In the downtown area of Nairobi, there are $L = 1\,497$ links, while only $|\mathcal{C}| = 52$ links are monitored by the webcams (about 3.5%). For those locations, we prepared $N = 100$ images by randomly picking still images taken at different times over the course of several days. Fig. 3 shows examples of the images. As can be seen, the webcams are typically mounted on buildings and are quite far from the roads due to antitheft concerns. Although the original size of the images is 640×480, the number of pixels in the region of interest is just several hundred, as suggested by Fig. 2.

To show how the counting algorithm works, Fig. 4 shows $\pi_d(x')\mathcal{N}(x'|\boldsymbol{m}^\top \boldsymbol{\phi}_d, \sigma_d^2)$ of Eq. (9) for two different x''s. Picking the maximum probability density, we see that $d' = 4$ for (a) and 13 for (b), which are consistent with the insets of the figure.

We compared our counting algorithm with a number of supervised alternatives. The first category of the alternatives includes regression algorithms: least squares linear regression (LS), least absolute values (LAV), and MM estimator (MM). See [38] for explanations of these algorithms. To train them,

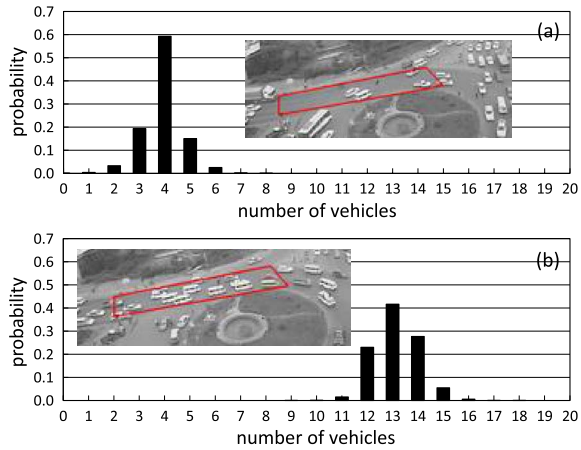


Fig. 4. Examples of the posterior probability of (9). (Insets) Corresponding images taken at Upperhill Roundabout in Nairobi.

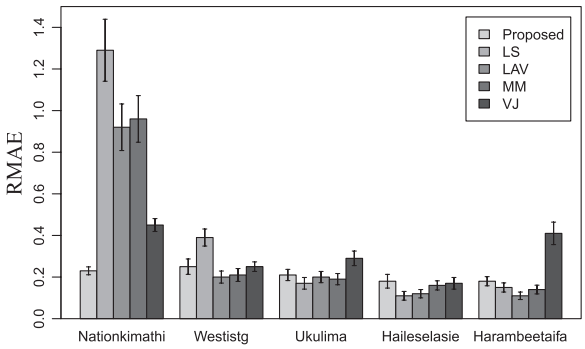


Fig. 5. Comparison of RMAE for vehicle counts (*Nairobi*).

we used the true count labels in addition to the vehicle-pixel-area feature; hence, the comparison is extremely preferable to the alternatives. We did not use nonlinear regression methods such as Gaussian process regression [7], because our preliminary experiments showed that the vehicle pixel area feature is mostly linearly correlated with the count. The second category is the object recognition approach by Viola and Jones (VJ) [39]. To make it work, we gave several hundred manually labeled images from the webcams in our setting in addition to 2,000 labeled images with positive (vehicle) and negative (non-vehicle) labels from general image databases containing vehicles [40]–[43]. Thus in terms of the cost to prepare the training data, the following inequality holds:

$$\text{(proposed method)} \lll \text{(LS, LAV, MM)} \ll \text{VJ}. \tag{24}$$

We remind the reader again that the proposed method does not need any labeled data for training.

Fig. 5 shows the relative mean absolute error (RMAE) computed by the leave-one-out cross validation scheme, which is defined by

$$\text{RMAE} = \frac{1}{100}\sum_{n=1}^{100}\frac{\left|d_{\text{true}}^{(n)} - d_{\text{estimate}}^{(n)}\right|}{d_{\text{true}}^{(n)} + 1} \tag{25}$$

where d_{true} is the ground truth count, which was manually prepared and entailed several person-days of labor. The error bars represent the standard deviation.

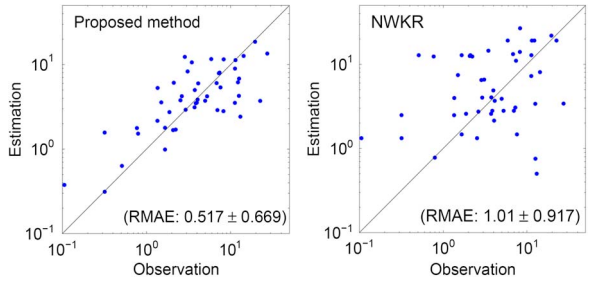


Fig. 6. Comparison of observed and estimated traffic flows (*Nairobi*).

In MIXPARAM of Algorithm 1, we gave $\boldsymbol{m}_0 = (-1, 0.3)^\top$, $\Sigma_0 = 10^{10}\mathrm{I}_2, a_0 = 1$ and $b_0 = 10^{-10}$, where I_2 is the two-dimensional identity matrix. Regarding β, we used a non-informative hyper-prior for numerical stability. This leads to a slight modification of the VB updating equations. For details, see our companion paper [16]. The proposed VB algorithm took only a few seconds for training on a moderate laptop computer. The time complexity is $O(N)$. The figure shows that our method is comparable to or even better than the supervised alternatives in terms of the error and robustness. In particular, when the quality of images is very low as is the case in Nationkimathi, our method outperforms the supervised alternatives.

B. Network Inference: Nairobi Data

Given the vehicle counts estimated by our unsupervised counting algorithm, we reconstructed the traffic flow at arbitrary links by using Algorithm 2. We set $\lambda_1 = 1$ and initialized $\boldsymbol{w} = \boldsymbol{0}, u_0 = u_1 = 1, u_{i,j} = 0$. For the road-type weight h_t, we used the values listed in Table I multiplied by $\ln[1 + N_{\text{L}}]$ with N_{L} being the number of road lanes. λ_2 and γ were determined by cross-validation. Vehicle velocities were also estimated with another algorithm (the details are omitted here due to space limitations). Thanks to the L_1 regularizer, more than 70% of the entries of $\boldsymbol{w}, \boldsymbol{u}$ became zero after optimization.

We compared our method with Nadaraya-Watson kernel regression (NWKR), which estimates the traffic at an arbitrary link j by

$$s_{\text{NWKR}}(j) = \frac{\sum_{i=1}^{|\mathcal{C}|} \mathrm{e}^{-\alpha N(j|i)} y(i)}{\sum_{i=1}^{|\mathcal{C}|} \mathrm{e}^{-\alpha N(j|i)}}.$$

Here, $N(j|i)$ is the number of hops from the i-th to the j-th links, and α is a parameter determined using cross validation. This form of kernel function is often used in network flow analysis [44].

Fig. 6 shows the results. To compare the estimation and observation, we used leave-one-out cross validation over the 52 observed links. In these figures, the 45° line represents perfect agreement. As can be seen, our method gives much better agreement than NWKR. In terms of RMAE, our method is about twice as good as than the alternative.

Fig. 7 compares the original and estimated traffic in color in downtown Nairobi. The red and yellow roads are most congested, while traffic on the blue roads flows smoothly. In

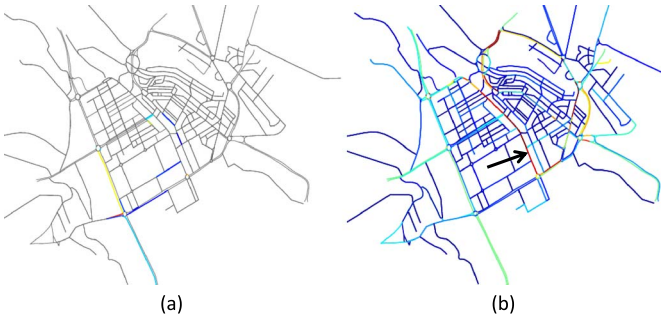


Fig. 7. Network flow estimation results in downtown Nairobi.

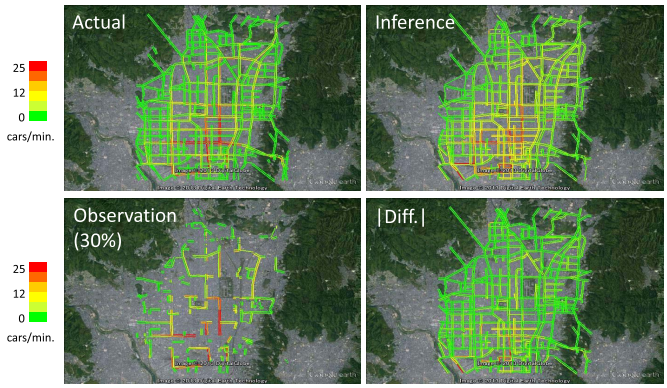


Fig. 8. Comparison of actual and inferred traffic in the central area of Kyoto. Map data 2013 Google.

Nairobi, traffic congestion in the downtown is a serious social problem, as pointed out by a local traffic survey report [45]. The most congested road highlighted with the arrow was in fact consistent with the survey.

C. Network Inference: Kyoto Data

Next, we tested our network inference algorithm against *Kyoto* data generated by an agent-based traffic simulator [32]. The simulation was carefully conducted so as to reproduce real driver behaviors and source-destination demand obtained from a person-trip survey [46]. The road network covers a roughly 11×10 kilometers square in the central area of the City of Kyoto, Japan, having $L = 2040$ links in total. We generated traffic volumes at all of the links and then randomly removed 70% of the traffic to create partially observed data with a 30% observation rate (see the left column of Fig. 8).

The results of the network inference are presented in the right column of Fig. 8, where |Diff.| means the absolute difference between inferred and the original traffic. The parameter settings were the same as those of Nairobi. In spite of the highly non-uniform traffic distribution over the links, the difference is quite uniform and mostly less than 4 cars per minute, which is within the acceptable range of the traffic authority of the city. We can also see that major traffic jams highlighted in red and orange are correctly captured in the inferred traffic. In those areas, a major source of traffic is taxis and buses, whose route choice behaviors are quite restrictive. Since our network inference algorithm does not have an explicit driver behavior model, the traffic looks more smeared-out in Fig. 8.

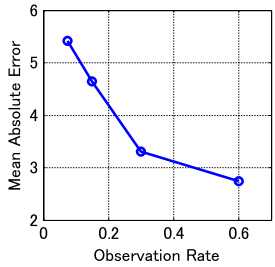


Fig. 9. Mean absolute error as a function of observation rate (*Kyoto*).

Finally, Fig. 9 shows the mean absolute error averaged over all the links as a function of observation rate. We can see that the acceptable range (4 cars per min.) is achievable at observation rates from 0.2 to 0.3.

VI. Concluding Remarks

We reported on a lightweight ITS approach that does not require infrastructure other than the Internet. The approach captures images from a web site of traffic cameras, and estimates the traffic by using the unsupervised probabilistic vehicle counting algorithm. Although the number of webcams is much smaller than the number of links in a road network in general, our new network inference algorithm, which is based on the inverse Markov formulation, makes it possible to estimate the entire traffic with reasonable accuracy. We tested our approach using traffic data collected in Nairobi, Kenya, and simulated traffic data of Kyoto, Japan.

Our "Frugal" ITS has already been deployed in Nairobi and has generated significant social impact as an example of a new model of innovation in developing countries. For example of the media coverage, we would suggest watching a TED talk by Navi Radjou [47]. The details of the deployed solution, combined with a mobile-based route recommendation, can be found in [48], [49].

Appendix A
Probability Distributions

The definitions of the gamma, beta, Gaussian, and Student's t distributions are as follows:

$$\mathrm{Gam}(\lambda|a,b) \equiv \frac{b^a}{\Gamma(a)}\lambda^{a-1}\mathrm{e}^{-b\lambda}$$

$$\mathrm{Beta}(v|\alpha,\beta) \equiv \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}v^{\alpha-1}(1-v)^{\beta-1}$$

$$\mathcal{N}(\boldsymbol{\theta}|\boldsymbol{m},\Sigma) \equiv \frac{|\Sigma|^{-\frac{1}{2}}}{(2\pi)^{\frac{W}{2}}}\exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m})^\top\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{m})\right\}$$

$$\mathcal{S}(z|\nu) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\pi\nu}\Gamma\left(\frac{\nu}{2}\right)}\left(1+\frac{z^2}{\nu}\right)^{-\frac{\nu+1}{2}}$$

where $\Gamma(\cdot)$ is the gamma function and W is the dimensionality of $\boldsymbol{\theta}$.

APPENDIX B
UNSUPERVISED COUNTING MODEL

This section explains the probabilistic counting framework outlined in Section III-B.

A. Observation Model and Prior Distributions

We first define the observation process by

$$p(x|\boldsymbol{h},\boldsymbol{\theta},\lambda) \equiv \prod_{d=0}^{D} \mathcal{N}\left(x|\boldsymbol{\theta}^\top \boldsymbol{\phi}_d, \lambda^{-1}\right)^{h_d} \tag{26}$$

where λ is a parameter representing the precision of observation, $\boldsymbol{\theta}$ represents the coefficients for $\boldsymbol{\phi}_d \equiv (1,d)^\top$, and $\boldsymbol{h} = (h_0,\ldots,h_D)^\top$ is the indicator vector of $h_d \in \{0,1\}$, and $\sum_{d=0}^{D} h_d = 1$. The key idea is to suppress the interchangeability in different d's by using the following form of the prior distribution for $\boldsymbol{h}$:

$$p(\boldsymbol{h}|\boldsymbol{v}) \equiv \prod_{d=0}^{D} \left\{ v_d \prod_{k=0}^{d-1} (1-v_k) \right\}^{h_d} \tag{27}$$

$$p(\boldsymbol{v}) \equiv \prod_{d=0}^{D} \mathrm{Beta}(v_d|1,\beta) \tag{28}$$

where β is a hyper-parameter treated as a given constant and Beta is the beta distribution (see Appendix A). These distributions are commonly called the stick-breaking process (SBP). As clearly indicated in the definition, the SBP prior is not symmetric in the cluster index d and naturally introduces the order in the components.

For the other parameters $\boldsymbol{\theta}, \lambda$, we set the conjugate priors as

$$p(\boldsymbol{\theta}|\boldsymbol{m}_0,\Sigma_0) \equiv \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{m}_0,\Sigma_0) \tag{29}$$

$$p(\lambda|a_0,b_0) \equiv \mathrm{Gam}(\lambda|a_0,b_0) \tag{30}$$

where $\boldsymbol{m}_0, \Sigma_0, a_0$, and b_0 are hyper-parameters treated as given constants, and Gam is the gamma distribution (see Appendix A).

B. Variational Bayes Solution

To derive the predictive distribution Eq. (7), we need to find the posterior distributions for $\boldsymbol{h}, \boldsymbol{\theta}$, and $\lambda, \boldsymbol{v}$. This can be systematically done via the variational Bayes (VB) algorithm [50]. The VB approach approximately finds the posterior distribution in a factorized form

$$p_{\text{post.}}(\mathsf{H},\boldsymbol{\theta},\lambda,\boldsymbol{v}) = q(\mathsf{H})q(\boldsymbol{\theta})q(\lambda)q(\boldsymbol{v}) \tag{31}$$

where we have used the same symbol q to represent different distributions for simplicity of notation.

The VB algorithm starts by writing down the complete likelihood as

$$P(\mathcal{D},\mathsf{H},\boldsymbol{\theta},\lambda,\boldsymbol{v}) \equiv \prod_{n=1}^{N} p(x^{(n)}|\boldsymbol{h}^{(n)},\boldsymbol{\theta},\lambda)p(\boldsymbol{h}^{(n)}|\boldsymbol{v}) \times p(\boldsymbol{\theta}|\boldsymbol{m}_0,\Sigma_0)p(\lambda|a_0,b_0)p(\boldsymbol{v}) \tag{32}$$

where H represents $\{\boldsymbol{h}^{(1)},\ldots,\boldsymbol{h}^{(N)}\}$. The main result of the VB algorithm is that the posterior distributions are given by the following simultaneous equations:

$$\ln q(\mathsf{H}) = \text{const.} + \langle \ln P(\mathcal{D},\mathsf{H},\boldsymbol{\theta},\lambda,\boldsymbol{v})\rangle_{\boldsymbol{\theta},\lambda,\boldsymbol{v}} \tag{33}$$

$$\ln q(\boldsymbol{\theta}) = \text{const.} + \langle \ln P(\mathcal{D},\mathsf{H},\boldsymbol{\theta},\lambda,\boldsymbol{v})\rangle_{\mathsf{H},\lambda,\boldsymbol{v}} \tag{34}$$

$$\ln q(\lambda) = \text{const.} + \langle \ln P(\mathcal{D},\mathsf{H},\boldsymbol{\theta},\lambda,\boldsymbol{v})\rangle_{\mathsf{H},\boldsymbol{\theta},\boldsymbol{v}} \tag{35}$$

$$\ln q(\boldsymbol{v}) = \text{const.} + \langle \ln P(\mathcal{D},\mathsf{H},\boldsymbol{\theta},\lambda,\boldsymbol{v})\rangle_{\mathsf{H},\boldsymbol{\theta},\lambda} \tag{36}$$

where $\langle\cdot\rangle_*$ represents the expectation w.r.t. the random variables $*$. By simply expanding the $\ln P$ term, we can easily see that the posterior distributions take the following forms:

$$q(\mathsf{H}) = \prod_{n=1}^{N}\prod_{d=0}^{D} \left\{\pi_d^{(n)}\right\}^{h_d^{(n)}} \tag{37}$$

$$q(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{m},\Sigma) \tag{38}$$

$$q(\lambda) = \mathrm{Gam}(\lambda|a,b) \tag{39}$$

$$q(\boldsymbol{v}) = \prod_{d=0}^{D} \mathrm{Beta}(v_d|\alpha_d,\beta_d) \tag{40}$$

where $\pi_d^{(n)}, \boldsymbol{m}, \Sigma, a, b, \alpha_d$, and β_d are unknown parameters to be determined.

To find these parameters, first we assume that $q(\mathsf{H})$ is given. Using the properties of the Gaussian, gamma, and beta distributions such as $\langle\boldsymbol{\theta}\rangle_{\boldsymbol{\theta}} = \boldsymbol{m}$ and $\langle\lambda\rangle_\lambda = a/b$, we easily see that the parameters satisfy the following relations:

$$N_d \leftarrow \sum_{n=1}^{N} \pi_d^{(n)},\ \bar{x}_d \leftarrow \sum_{n=1}^{N} \pi_d^{(n)} x^{(n)} \tag{41}$$

$$\Delta_d(x^{(n)}) \leftarrow (x^{(n)} - \boldsymbol{\phi}_d^\top \boldsymbol{m})^2 + \boldsymbol{\phi}_d^\top \Sigma \boldsymbol{\phi}_d \tag{42}$$

$$a \leftarrow a_0 + \frac{N}{2} \tag{43}$$

$$b \leftarrow b_0 + \frac{1}{2}\sum_{n=1}^{N}\sum_{d=0}^{D} \pi_d^{(n)} \Delta_d\left(x^{(n)}\right) \tag{44}$$

$$\Sigma \leftarrow \left[\Sigma_0^{-1} + \frac{a}{b}\sum_{d=0}^{D} N_d \boldsymbol{\phi}_d \boldsymbol{\phi}_d^\top\right]^{-1} \tag{45}$$

$$\boldsymbol{m} \leftarrow \Sigma^{-1}\left[\Sigma_0^{-1}\boldsymbol{m}_0 + \frac{a}{b}\sum_{d=0}^{D} \bar{x}_d \boldsymbol{\phi}_d\right] \tag{46}$$

$$\alpha_d \leftarrow 1 + N_d,\ \beta_d \leftarrow \beta + \sum_{k=d+1}^{D} N_k. \tag{47}$$

To compute $\{\pi_d^{(n)}\}$, we assume in turn that $q(\boldsymbol{\theta}), q(\lambda)$ and $q(\boldsymbol{v})$ are given. Expanding the $\ln P$ term and taking the

expectation w.r.t. these variables, we have

$$\ln \pi_d^{(n)} \leftarrow \sum_{k=1}^{d-1} [\psi(\beta_k) - \psi(\alpha_k + \beta_k)] + \psi(\alpha_d) - \psi(\alpha_d + \beta_d) - \frac{a}{2b}\Delta_d(x^{(n)}) \quad (48)$$

$$\pi_d^{(n)} \leftarrow \frac{\pi_d^{(n)}}{\sum_{l=0}^{D} \pi_l^{(n)}} \quad (49)$$

where $\psi(\cdot)$ is the digamma function. This follows from the well-known formula [51]

$$\int \mathrm{d}v_d \, \mathrm{Beta}(v_d|\alpha_d, \beta_d) \ln v_d = \psi(\alpha_d) - \psi(\alpha_d + \beta_d). \quad (50)$$

Equations (41)–(47) and (48), (49) are iteratively computed until convergence.

C. Deriving the Predictive Distribution

Now we are ready to derive the predictive distribution (7). By definition, it is formally written as

$$p(x|\mathcal{D}) = \sum_{\boldsymbol{h}} \int \mathrm{d}\boldsymbol{\theta} \int \mathrm{d}\lambda p(x|\boldsymbol{h}, \boldsymbol{\theta}, \lambda) q(\boldsymbol{h}) q(\boldsymbol{\theta}) q(\lambda).$$

One problem here is that there is no explicit expression for the posterior $q(\boldsymbol{h})$ for an arbitrary value of x. Here, we can use the following approximation. Imagine we had an augmented data set $\mathcal{D} \cup x$, and we got a posterior on this $N+1$ data as

$$p_{\text{post.}}(\boldsymbol{h}, \mathsf{H}, \Psi|\mathcal{D}, x) = p(\boldsymbol{h}, \mathsf{H}|\Psi, \mathcal{D}, x) p(\Psi|\mathcal{D}, x) \quad (51)$$

where Ψ collectively represents $\boldsymbol{\theta}, \lambda, \boldsymbol{v}$. Equation (37) suggests that $p(\boldsymbol{h}, \mathsf{H}|\Psi, \mathcal{D}, x)$ should be factorized as

$$\begin{aligned} p(\boldsymbol{h}, \mathsf{H}|\Psi, \mathcal{D}, x) &= p(\boldsymbol{h}|\Psi, \mathcal{D}, x) p(\mathsf{H}|\Psi, \mathcal{D}, x) \\ &= p(\boldsymbol{h}|\Psi, x) p(\mathsf{H}|\Psi, \mathcal{D}). \end{aligned} \quad (52)$$

The second line follows from the fact that the dependency of $\pi_d^{(n)}$ on $\mathcal{D}$ is only through Ψ except for $x^{(n)}$. In Eq. (51), we can use the approximation as $p(\Psi \mid \mathcal{D}, x) \approx p(\Psi \mid \mathcal{D})$ as long as $N \gg 1$, so that Ψ in $p(\boldsymbol{h} \mid \Psi, x)$ can be thought of the one learned from the original N sample data $\mathcal{D}$. Therefore, we conclude that the posterior distribution of $\boldsymbol{h}$ is the categorical distribution whose d-th probability mass is given by

$$\ln \pi_d(x) = \sum_{k=1}^{d-1} [\psi(\beta_k) - \psi(\alpha_k + \beta_k)] + \psi(\alpha_d) - \psi(\alpha_d + \beta_d) - \frac{a}{2b}\Delta_d(x) \quad (53)$$

$$\pi_d(x) \leftarrow \frac{\pi_d(x)}{\sum_{l=0}^{D} \pi_l(x)}. \quad (54)$$

Using this approximation, we get

$$\begin{aligned} p(x|\mathcal{D}) &\approx \sum_{d=0}^{D} \pi_d(x) \int \mathrm{d}\boldsymbol{\theta} \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{m}, \Sigma) \\ &\quad \times \sqrt{\frac{a}{b}} \mathcal{S}\left(\sqrt{\frac{a}{b}}(x - \boldsymbol{\theta}^\top \boldsymbol{\phi}_d) \middle| 2a \right) \\ &\approx \sum_{d=0}^{D} \pi_d(x) \mathcal{N}\left(x \middle| \boldsymbol{m}^\top \boldsymbol{\phi}_d, \frac{b}{a-1} + \boldsymbol{\phi}_d^\top \Sigma \boldsymbol{\phi}_d \right) \end{aligned} \quad (55)$$

where $\mathcal{S}$ is Student's t-distribution (see Appendix A). The last expression of Eq. (55) follows from the fact that the t distribution is approximated by a Gaussian when the degree of freedom is large. In this case, a is a large number on the order of N [see Eq. (43)], and this approximation is almost always justified. Equation (55) also shows that the variance σ_d^2 in Eq. (7) is given by Eq. (8). Since Σ is positive semidefinite, we see that σ_d^2 monotonically increases as $d = 0,1,2,\ldots$.

Algorithm 1 summarizes the algorithm for learning model parameters.

REFERENCES

[1] N. Radjou, J. Prabhu, and S. Ahuja, *Jugaad Innovation*. San Francisco, CA, USA: Jossey-Bass, 2012.

[2] AccessKenya.com. [Online]. Available: http://traffic.accesskenya.com/

[3] F. Cathey and D. Dailey, "A novel technique to dynamically measure vehicle speed using uncalibrated roadway cameras," in *Proc. IEEE Intell. Veh. Symp.*, 2005, pp. 777–782.

[4] K. Robert, "Video-based traffic monitoring at day and night vehicle features detection tracking," in *Proc. IEEE 12th Int.*, 2009, pp. 1–6.

[5] B. Tian, Q. Yao, Y. Gu, K. Wang, and Y. Li, "Video processing techniques for traffic flow monitoring: A survey," in *Proc. IEEE 14th Int. Conf. Intell. Transp. Syst.*, 2011, pp. 1103–1108.

[6] N. Buch, S. A. Velastin, and J. Orwell, "A review of computer vision techniques for the analysis of urban traffic," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 3, pp. 920–939, Sep. 2011.

[7] M. Liang, X. Huang, C. H. Chen, X. Chen, and A. Tokuta, "Counting and classification of highway vehicles by regression analysis," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2878–2888, Oct. 2015.

[8] R. Guerrero-Gómez-Olmedo, B. Torre-Jiménez, R. López-Sastre, S. Maldonado-Bascóon, and D. Oñoro-Rubio, "Extremely overlapping vehicle counting," in *Proc. 7th Iberian Conf. Pattern Recog. Image Anal.*, 2015, vol. 9117, pp. 423–431.

[9] H. J. V. Zuylen and L. G. Willumsen, "The most likely trip matrix estimated from traffic counts," *Transp. Res. B, Methodol.*, vol. 14, no. 3, pp. 281–293, 1980.

[10] R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu, "Network tomography: Recent developments," *Statist. Sci.*, vol. 19, no. 3, pp. 499–517, 2004.

[11] E. Lawrence, G. Michailidis, V. N. Nair, and B. Xi, "Network tomography: A review and recent developments," in *In Fan and Koul, editors, Frontiers in Statistics*. London, U.K.: Imperial College Press, 2006, pp. 345–364.

[12] H. Shao, W. H. Lama, A. Sumalee, A. Chen, and M. L. Hazelton, "Estimation of mean and covariance of peak hour origin–destination demands from day-to-day traffic counts," *Transp. Res. B, Methodol.*, vol. 68, pp. 52–75, 2014.

[13] S. Santini, "Analysis of traffic flow in urban areas using Web cameras," in *Proc. IEEE Workshop Appl. Comput. Vis.*, 2000, pp. 140–145.

[14] C.-N. Anagnostopoulos, I. Anagnostopoulos, I. Psoroulas, V. Loumos, and E. Kayafas, "License plate recognition from still images and video sequences: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 3, pp. 377–391, Sep. 2008.

[15] T. Idé, T. Katsuki, T. Morimura, and R. Morris, "Monitoring entire-city traffic using low-resolution Web cameras," in *Proc. 20th ITS World Congr.*, Tokyo, Japan, 2013, pp. 1–10.

[16] T. Katsuki, T. Morimura, and T. Idé, "Bayesian unsupervised vehicle counting," IBM Res. Rep., Armonk, NY, USA, RT0951, 2013.

[17] T. Morimura, T. Osogami, and T. Idé, "Solving inverse problem of Markov chain with partial observations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 1655–1663.
[18] T. Katsuki, T. Morimura, and T. Idé, "Unsupervised object counting without object recognition," in *Proc. 23rd ICPR*, 2016, to be published.
[19] C. de Fabritiis, R. Ragona, and G. Valenti, "Traffic estimation and prediction based on real time floating car data," in *Proc. IEEE 11th Int. ITSC*, 2008, pp. 197–203.
[20] Q.-J. Kong, Q. Zhao, C. Wei, and Y. Liu, "Efficient traffic state estimation for large-scale urban road networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 1, pp. 398–407, Mar. 2013.
[21] Z. Shan and Q. Zhu, "Camera location for real-time traffic state estimation in urban road network using big GPS data," *Neurocomputing*, vol. 169, pp. 134–143, Dec. 2015.
[22] X. D. Yu, L. Y. Duan, and Q. Tian, "Highway traffic information extraction from Skycam MPEG video," in *Proc. IEEE 5th Int. Conf. Intell. Transp. Syst.*, 2002, pp. 37–42.
[23] R. Huck, J. Havlicek, J. Sluss, and A. Stevenson, "A low-cost distributed control architecture for intelligent transportation systems deployment in the State of Oklahoma," in *Proc. IEEE Intell. Transp. Syst.*, 2005, pp. 919–924.
[24] S.-R. Hu, S. Peeta, and H.-T. Liou, "Integrated determination of network origin–destination trip matrix and heterogeneous sensor selection and location strategy," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 1, pp. 195–205, Jan. 2016.
[25] A. K. Menon, C. Cai, W. Wang, T. Wen, and F. Chen, "Fine-grained OD estimation with automated zoning and sparsity regularisation," *Transp. Res. B, Methodol.*, vol. 80, pp. 150–172, Oct. 2015.
[26] S. L. Lauritzen, *Graphical Models*. Oxford, U.K.: Clarendon, 1996.
[27] C. Zhang, S. Sun, and G. Yu, "A Bayesian network approach to time series forecasting of short-term traffic flows," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, 2004, pp. 216–221.
[28] S. Sun, C. Zhang, and G. Yu, "A Bayesian network approach to traffic flow forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 124–132, Mar. 2006.
[29] S. Zhu, L. Cheng, Z. Chu, A. Chen, and J. Chen, "Identification of network sensor locations for estimation of traffic flow," *Transp. Res. Rec.*, vol. 2443, pp. 32–39, 2014.
[30] M. Drton and M. D. Perlman, "A SINful approach to Gaussian graphical model selection," *J. Statist. Plan. Inference*, vol. 138, pp. 1179–1200, 2008.
[31] N. Otsu, "A threshold selection method from gray-level histogram," *IEEE Trans. Syst., Man Cybern.*, vol. 9, no. 1, pp. 62–66, Jan. 1979.
[32] T. Osogami, T. Imamichi, H. Mizuta, T. Suzumura, and T. Idé, "Toward simulating entire cities with behavioral models of traffic," *IBM J. Res. Develop.*, vol. 57, no. 5, pp. 6:1–6:10, 2013.
[33] S. Kullback and R. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 55, pp. 79–86, 1951.
[34] Y. Zhang, M. Roughan, C. Lund, and D. Donoho, "An information-theoretic approach to traffic matrix estimation," in *Proc. Conf. Appl., Technol., Archit., Protocols Comput. Commun.*, 2003, pp. 301–312.
[35] N. Shlayan, P. Kachroo, and S. Wadoo, "Transportation reliability based on information theory," in *Proc. IEEE 14th Int. ITSC*, 2011, pp. 1415–1420.
[36] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Roy. Statist. Soc.*, vol. 67, pp. 301–320, 2005.
[37] S. Amari, "Natural gradient works efficiently in learning," *Neural Comput.*, vol. 10, no. 2, pp. 251–276, 1998.
[38] R. R. Wilcox, *Introduction to Robust Estimation and Hypothesis Testing*. New York, NY, USA: Academic, 2012.
[39] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Comput. Soc. Conf. CVPR*, 2001, vol. 1, pp. 511–518.
[40] P. Negri, X. Clady, S. M. Hanif, and L. Prevost, "A cascade of boosted generative and discriminative classifiers for vehicle detection," *EURASIP J. Adv. Signal Process.*, vol. 2008, p. 136, 2008.
[41] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.
[42] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, The PASCAL VOC2012 Results, 2012. [Online]. Available: http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html
[43] C. Papageorgiou and T. Poggio, "A trainable system for object detection," *Int. J. Comput. Vis.*, vol. 38, no. 1, pp. 15–33, 2000.
[44] T. Idé and S. Kato, "Travel-time prediction using Gaussian process regression: A trajectory-based approach," in *Proc. SIAM Int. Conf. SDM*, 2009, pp. 1185–1196.
[45] E. J. Gonzales, C. Chavis, Y. Li, and C. F. Daganzo, "Multimodal transport in Nairobi, Kenya: Insights and recommendations with a macroscopic evidence-based model," in *Proc. 90th Annu. Meet. Transp. Res. Board*, 2011, pp. 11–3045.
[46] Ministry of Land, Infrastructure, Transport and Tourism of Japan, 2010. [Online]. Available: http://www.mlit.go.jp/k-toukei/transportation_statistics.html
[47] N. Radjou, "Creative problem-solving in the face of extreme limits, 2014, [Online; accessed May 23, 2016]. [Online]. Available: https://www.ted.com/talks/navi_radjou_creative_problem_solving_in_the_face_of_extreme_limits
[48] T. Ehrlich and E. Fu, Fixing Traffic Congestion in Kenya: Twende Twende, 2015. [Online]. Available: http://www.forbes.com/sites/ehrlichfu/2015/03/03/fixing-traffic-congestion-in-kenya-twende-twende/
[49] A. Kinai, R. E. Bryant, A. Walcott-Bryant, E. Mibuari, K. Weldemariam, and O. Stewart, "Twende-Twende: A mobile application for traffic congestion awareness and routing," in *Proc. 1st Int. Conf. Mobile Softw. Eng. Syst.*, 2014, pp. 93–98.
[50] C. M. Bishop, *Pattern Recognition and Machine Learning*. Berlin, Germany: Springer-Verlag, 2006.
[51] Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Beta_distribution

Tsuyoshi Idé received the Ph.D. degree in theoretical physics from University of Tokyo, Tokyo, Japan, in 2000.

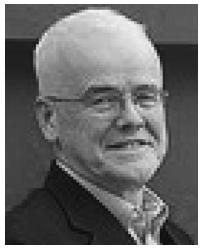
In 2000, he joined IBM Research—Tokyo where he led many research projects related to sensor data analytics and traffic modeling. In 2013, he moved to IBM Thomas J. Watson Research Center, IBM Research, Yorktown Heights, NY, USA, where he is currently a Senior Technical Staff Member of the Mathematical Science Department. His major research area is machine learning, especially from noisy time-series data.

Takayuki Katsuki received the M.E. degree in electrical engineering and bioscience from Waseda University, Tokyo, Japan, in 2012.

He is a Researcher with IBM Research—Tokyo, Tokyo. His research interests include machine learning, data mining, and their applications.

Tetsuro Morimura received the M.E. and Ph.D. degrees in engineering from Nara Institute of Science and Technology, Ikoma, Japan, in 2005 and 2008, respectively.

He is a Researcher with IBM Research—Tokyo. His research interests include machine learning, especially reinforcement learning and its applications.



Robert Morris (F'01) received the Ph.D. degree in computer science from University of California, Los Angeles, CA, USA.

He is the Vice President for Global Laboratories, IBM Research, Singapore, which includes laboratories in China, India, Japan, Australia, Brazil, and Africa. Before this current role, he was the Vice President for Services Research, IBM Thomas J. Watson Research Center; the Director of the IBM Almaden Research Center; and the Vice-President responsible for research on IBM's ThinkPad. He was Editor of IEEE TRANSACTIONS ON COMPUTERS.