A Study on Bayesian Optimal Estimation with Probabilistic Hidden Structure Modeling

確率的隠れ構造モデリングを用いた Bayes最適な推定に関する研究

February, 2017

Takayuki KATSUKI
勝木 孝行

A Study on Bayesian Optimal Estimation with Probabilistic Hidden Structure Modeling

確率的隠れ構造モデリングを用いた Bayes 最適な推定に関する研究

February, 2017

Waseda University
Graduate School of Advanced Science and Engineering
Department of Electrical Engineering and Bioscience
Research on Probabilistic Information Processing

Takayuki KATSUKI
勝木 孝行

Contents

Chapter.1

Introduction

This thesis presents a study on Bayesian optimal estimation with probabilistic models having a hidden structure. Using Bayesian optimal estimation with probabilistic hidden structure modeling, we tackle the problems of traffic flow estimation, image super resolution, and input selective regression. For each problem setting, we derive an efficient computational algorithm.

1.1 Probabilistic Information Processing for Data Analytics

The importance of data and information processing for analyzing data is growing. As the amount of data increases, the demand for information processing is becoming ubiquitous [1, 2]. A large amount of data is generated from devices with sensors that are connected to us and each other through the Internet [3, 4]. People can also be regarded as sensors generating information [5, 6]. We create a large amount of data through social networks and mobile devices. Through information processing, we can use these data for solving various real-world problems.

Most forms of information processing transform the observed data into a form that people can interpret as useful [7]. For example, there have been many studies on transforming sentences into meanings and topics [8, 9] and transforming images into what objects there are in the images [10, 11]. Transformation processes can be formulated mathematically and are defined along with what we want to do.

The transformation process can be explicitly expressed in the form of an estimation function. The input and output for this function are the observed data and the transformed information, respectively. When we design an estimation function based on the probabilistic distribution for representing the data, we call such information processing probabilistic information processing.

1.2 Information Processing as Estimation Function Based on Model and Evaluation Criterion

The estimation function is the result of optimization of an objective function consisting of a model and evaluation criterion [7,12,13]. In a situation in which unobservable values are to be estimated from observed data with an estimation function, the model represents the characteristics of the observed data, the values, and the relationship between them. The evaluation criterion shows what the estimates made by the estimation function should be; e.g., the estimate is better if the difference from the true value that follows the model is smaller.

The model is an assumption representing simplified and generalized characteristics of the data for helping people to understand it, and it is usually written in a mathematical form [14,15]. In probabilistic information processing, the model is defined as a probabilistic distribution, which can properly describe the stochastic variation of the data. The modeling is the process by which the model is constructed on the basis of prior knowledge about the data. Well-known examples of modeling are representing sentences consisting of a set of characters on the basis of a sequence of words [16–19] and representing an image consisting of a set of pixels on the basis of a set of local patterns [20, 21]. Such modeling makes it possible to interpret the data. In these examples of natural language processing and image processing, we can discriminate each of the extracted sequences of words and sets of local patterns through differences of frequencies of them.

The evaluation criterion is generally either a minimization or maximization of a function. The most successful examples are based on the squared error. The method of minimizing the sample mean of the squared differences between values estimated by the estimation function and the true values is called least squares; it has been applied to various problems [22,23]. In studies in which prediction is the main focus, such as on machine learning, the minimization of the generalization error is mostly used as the evaluation criterion [24]. The generalization error is the population mean of an error function, such as the squared difference. Since the evaluation criterion generally involves the model, it can be regarded as a functional of the model. We call the evaluation criterion involving the specific model the objective function to be optimized.

Since the model and evaluation criterion in the objective function are closely related, we need to construct and constrain the model in consideration of the balance between its complexity for sufficiently representing the data and the computational feasibility

and stability of the optimization of the objective function with the evaluation criterion [25–27]. For example, if we use a model that has extremely high flexibility, since the flexibility of the model almost corresponds to the number of model parameters, the number of parameters to be optimized is quite large in such a model. Since the computational cost increases as the number of parameters to be optimized increases, an optimization based on such a model is generally difficult. On the other hand, a model with less flexibility is not always better. Since such a model cannot represent data sufficiently, the estimation function as the result of the optimization based on the model will not have the desired performance.

1.3 Bayesian Optimal Estimation for Dealing with Data Insufficiency

Despite that the total amount of data has been rapidly increasing in recent years, the available data always seems relatively insufficient because the complexity of the purpose of information processing and the model used therein increase even faster. The purpose hence becomes ever more localized and personalized, and there is always a requirement for new data specialized for the task at hand. Even though we could use general-purpose datasets for the task, it would be biased against the task. The models used in recent studies, such as on deep learning [28–33], have huge numbers of parameters and require more training data than conventional common sense would suggest. In addition, data usually have many uncertainties associated with them and are unstructured. They may be noisy because of poor sensor quality or limited network bandwidth and may contain ambiguous expressions, such as colloquial expressions, and have poor photographic quality [3–6]. The amount of essential information that can be extracted from such data is not so large. All of these deficiencies as to quantity and quality mean that data often lack information essential for information processing.

For solving the data insufficiency problem, we study an estimation function derived from the optimization of the objective function using the Bayesian evaluation criteria. We call this estimation Bayesian optimal estimation. It is generally known that the Bayesian optimal estimation reaches a stable solution [13, 34] even in the case of insufficient data. This is because it considers the stochastic variation of the assumed model for optimization in the criteria. Since the model is a hypothesis based on data observed at random, when there is a limited amount of essential information in the data, to consider a only single model, like the maximum likelihood method, may cause unstable estimation. On the other hand, since Bayesian evaluation criteria consider multiple models that may fit the observed data and average over the different

models, Bayesian optimal estimation is more stable than other methods like maximum likelihood estimation.

1.4 Bayesian Optimal Estimation with Hidden Structure Modeling

In this thesis, we use Bayesian optimal estimation to solve problems in unsupervised estimation and hidden variables estimation in which the amount of data tends to be small for making an estimation and the solution tends to be unstable. In spite of the stability of its solution, Bayesian optimal estimation entails a large computational cost, and we can use only a limited class of models along with the criteria. Here, we propose a model that maintains the computational feasibility of the estimation and has appropriate complexity for representing the data in each problem setting.

We address the problems of traffic flow estimation, image super resolution, and input selective regression. The models used in each problem setting have similar hidden structures representing the data generating process. By averaging over the different possible models in Bayesian optimal estimation, we solve the inverse problem for estimating informative hidden variables and parameters in the model.

In the traffic flow estimation task, we estimate the traffic flow from features in traffic image sequences. We consider models that may be able to generate the observed features from the traffic flows, and then, by averaging over these possible models, we derive the Bayesian optimal estimation function for estimating the traffic flow from the features (i.e., solving the inverse problem). In the image super resolution task, we estimate a high-resolution image from a set of low-resolution images. Similar to the traffic flow task, we consider models that may be able to generate the observed low-resolution images, and then, by averaging over these possible models, we derive the Bayesian optimal estimation function for estimating the high-resolution image from the low-resolution images (i.e., solving the inverse problem). In the input selective regression task, we estimate the label for the new input data from sets of training samples that consists of pairs of data and labels. Again, we consider models may be able to generate the observed training samples from hidden variables selecting the effective part of the input data, and then, by averaging over the possible models, we derive the Bayesian optimal estimation function for estimating the label for the new input data from sets of training samples (i.e., solving the inverse problem).

1.5 Structure of Thesis

This thesis is organized as follows.

In Chapter 2, we explain the general framework of this study, which is Bayesian optimal estimation with probabilistic hidden structure modeling. We also derive several Bayesian optimal estimation functions from the specific evaluation criteria used throughout this thesis and derive computational algorithms for efficiently computing them as tools for accomplishing the tasks addressed in this thesis.

In Chapter 3, we address the traffic-flow-estimation problem for a novel lightweight approach to traffic monitoring using web-cameras as data sources [35–37]. We formulate the task as an unsupervised learning problem without the expensive steps of recognizing and tracking vehicles.

We tackle the super-resolution problem in Chapter 4 [38, 39]. We propose a novel model for super-resolution and derive an efficient estimation algorithm with a fully Bayesian treatment using image priors implementing both of smoothness and edges in images.

In Chapter 5, we describe a regression method for selecting the valuable parts of each training data instance with latent variable modeling [40].

We conclude the thesis in Chapter 6.

Every parameter or variable is independently defined in each chapter; i.e., the parameters and variables in a certain chapter have nothing to do with those in other chapters.

Chapter.2

General Framework of Bayesian Optimal Estimation with Hidden Structure Modeling

Here, we explain the general framework of Bayesian optimal estimation with hidden structure modeling.

We address a particular class of estimation problems in which we estimate the unobservable variables, either $\boldsymbol{y} \in \mathbb{R}^K$ or $\boldsymbol{y} \in \mathbb{N}^K$, from the observed variables, $\boldsymbol{x} \in \mathbb{R}^D$, through the use of an estimation function $\boldsymbol{y}^*(\boldsymbol{x})$, where $\mathbb{R}$ is the real number field, $\mathbb{N}$ is the set of natural numbers including zero, and K and D are the dimensions of $\boldsymbol{y}$ and $\boldsymbol{x}$, respectively. We cannot directly obtain the unobservable variables from the observed data, but we can directly obtain the observed variables from the observed data.

Based on the Bayesian perspective, we formalize the problem as a minimization of the population mean of the error function, $\mathrm{Error}(\boldsymbol{y}, \boldsymbol{y}^*(\boldsymbol{x}))$, that represents the difference between true unobservable variables $\boldsymbol{y}$ that follow the model and the estimates of $\boldsymbol{y}$ by the estimation function $\boldsymbol{y}^*(\boldsymbol{x})$ to which the observed variables $\boldsymbol{x}$ have been input:

$$\hat{\boldsymbol{y}}^*(\boldsymbol{x}) \equiv \underset{\boldsymbol{y}^*(\boldsymbol{x})}{\mathrm{argmin}} \ \left\langle \mathrm{Error}(\boldsymbol{y}, \boldsymbol{y}^*(\boldsymbol{x})) \right\rangle_{p(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})}, \tag{2.1}$$

where $p(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta})$ is the model which is represented as the probabilistic distribution for $\boldsymbol{y}$, $\boldsymbol{x}$, and the model parameters $\boldsymbol{\theta}$. It requires good estimation performance on average over various observations, $\boldsymbol{x}$, and the corresponding $\boldsymbol{y}$ and $\boldsymbol{\theta}$ and produces a stable estimation. In this thesis, we assume that the occurrence rate of $\boldsymbol{y}$, $\boldsymbol{x}$, and $\boldsymbol{\theta}$ exactly coincides with the distribution $p(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta})$. The specific form of the distribution is explained in the following chapters. The right-hand side of Eq. (2.1) is the Bayesian evaluation criterion which is characterized by $\mathrm{Error}(\boldsymbol{y}, \boldsymbol{y}^*(\boldsymbol{x}))$. For each

task addressed in the following chapters, we derive a Bayesian optimal estimation algorithm in the following steps:

1. Define the evaluation criterion for each problem setting and derive an **Bayesian optimal estimation function** for the criterion. From Eq. (2.1), we specifically define the error function for each problem. Since the evaluation criterion is a functional of the model, the Bayesian optimal estimation function is also derived as a functional of the model.
2. Design the **probabilistic model** $p(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta})$ by maintaining the computational feasibility of the derived estimation function involving the model and appropriate complexity for representing each problem.
3. Derive an efficient **computational algorithm** for the derived estimation function and designed model.

In the following sections in this chapter, we introduce several Bayesian evaluation criteria and derive $\hat{\boldsymbol{y}}^*(\boldsymbol{x})$ by optimizing each of them. After that, we explain the Markov chain Monte Carlo (MCMC) method and variational Bayes (VB) method, which are efficient computational algorithms for the estimation functions. We also propose an approximation method for the VB method based on a Taylor approximation. These are the tools for the tasks addressed in this thesis.

2.1 Probabilistic Models and Notations

Here, we give the definitions of the gamma, inverse gamma, beta, categorical, Bernoulli, uniform, Poisson, and Gaussian distributions used in this thesis:

$$\mathrm{Gamma}(x|a,b) \equiv \frac{b^a}{\Gamma(a)} x^{a-1} \mathrm{e}^{-bx} \quad (x>0, \quad a>0, \quad b>0), \tag{2.2}$$

$$\mathrm{InverseGamma}(x|a,b) \equiv \frac{b^a}{\Gamma(a)} \left(\frac{1}{x}\right)^{a+1} \mathrm{e}^{-\frac{b}{x}} \quad (x>0, \quad a>0, \quad b>0), \tag{2.3}$$

$$\mathrm{Beta}(x|a,b) \equiv \frac{1}{B(a,b)} x^{a-1}(1-x)^{b-1} \quad (0 \leq x \leq 1, \quad a>0, \quad b>0), \tag{2.4}$$

$$\mathrm{Categorical}(\boldsymbol{x}|\boldsymbol{\mu}) \equiv \prod_{d=1}^{D} \mu_d^{x_d} \tag{2.5}$$

$$(x_d \in \{0,1\}, \quad \sum_{d=1}^{D} x_d \equiv 1, \quad 0 \leq \mu_d \leq 1, \quad \sum_{d=1}^{D} \mu_d \equiv 1),$$

$$\mathrm{Bernoulli}(x|\mu) \equiv \mu^x (1-\mu)^{1-x} \quad (x \in \{0,1\}, \quad 0 \leq \mu \leq 1), \tag{2.6}$$

$$\mathcal{U}(x|a,b) \equiv \begin{cases} \frac{1}{b-a}, & a \leq x \leq b, \\ 0, & \text{otherwise}, \end{cases} \quad (x \in \mathbb{R}), \tag{2.7}$$

$$\mathrm{Poisson}(x|\mu) \equiv \frac{\mu^x \mathrm{e}^{-\mu}}{x!} \quad (x \in \mathbb{N}, \quad \mu>0), \tag{2.8}$$

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) \equiv |2\pi\boldsymbol{\Sigma}|^{-\frac{1}{2}} \mathrm{e}^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} \quad (\boldsymbol{x} \in \mathbb{R}^D, \quad \boldsymbol{\mu} \in \mathbb{R}^D, \quad \boldsymbol{\Sigma} \in \mathbb{R}^{D \times D}), \tag{2.9}$$

where Γ is the gamma function, B is the beta function, $|\bullet|$ denotes the determinant of a given matrix, superscript $\top$ denotes the transpose, and D is the dimensionality of $\boldsymbol{x}$. The sigmoid function and Kullback-Leibler (KL) divergence from distributions $p(\boldsymbol{x})$ to $q(\boldsymbol{x})$ are respectively defined as

$$\mathrm{Sigmoid}(x) \equiv \frac{1}{1+\mathrm{e}^{-x}}, \tag{2.10}$$

$$D_{\mathrm{KL}}(p(\boldsymbol{x}) \| q(\boldsymbol{x})) \equiv \left\langle \ln \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} \right\rangle_{p(\boldsymbol{x})}, \tag{2.11}$$

where the angle brackets $\langle \bullet \rangle_\circ$ denote the expectation of $\bullet$ with respect to a distribution $\circ$. Additionally, tr denotes the trace of a given matrix, diag denotes a diagonal matrix, $\boldsymbol{I}$ is an identity matrix of appropriate size, $\boldsymbol{i}$ represents a vector of all ones, and $\boldsymbol{0}$ is a zero vector or a zero matrix of appropriate size. All the vectors in this thesis are column vectors.

2.2 Bayesian Optimal Estimation Function

We introduce evaluation criteria based on the Bayesian perspective and derive the optimal estimation functions for them. All of the estimation functions that are derived in this section are related to the posterior distribution, in which all of the hidden variables and model parameters other than the target variables $\boldsymbol{y}$ and the observed variables $\boldsymbol{x}$ are marginalized out. This is the key ingredient in Bayesian evaluation criteria; that is, to marginalize out the hidden variables and model parameters, rather than to optimize. This marginalization corresponds to "averaging over the different models".

2.2.1 Posterior Mean as Bayesian Optimal Estimation Function for Mean Square Error

First, we consider the squared difference between $\boldsymbol{y}$ and $\boldsymbol{y}^*(\boldsymbol{x})$ as the error function $\mathrm{Error}(\boldsymbol{y}, \boldsymbol{y}^*(\boldsymbol{x}))$:

$$\mathrm{Error}(\boldsymbol{y}, \boldsymbol{y}^*(\boldsymbol{x})) \equiv \|\boldsymbol{y} - \boldsymbol{y}^*(\boldsymbol{x})\|_2^2, \tag{2.12}$$

where $\| \bullet \|_2$ denotes the $L2$-norm of a given vector $\bullet$.

The evaluation criterion is to minimize the population mean of the error function in Eq. (2.12), as

$$\underset{\boldsymbol{y}^*(\boldsymbol{x})}{\mathrm{argmin}} \left\langle \|\boldsymbol{y} - \boldsymbol{y}^*(\boldsymbol{x})\|_2^2 \right\rangle_{p(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})}. \tag{2.13}$$

We can then explicitly compute the optimal estimation function $\hat{\boldsymbol{y}}^*(\boldsymbol{x})$ as the posterior mean (PM):

$$\begin{aligned} \hat{\boldsymbol{y}}^*(\boldsymbol{x}) &= \underset{\boldsymbol{y}^*(\boldsymbol{x})}{\mathrm{argmin}} \left\langle \|\boldsymbol{y} - \boldsymbol{y}^*(\boldsymbol{x})\|_2^2 \right\rangle_{p(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})} \\ \langle \hat{\boldsymbol{y}}^*(\boldsymbol{x}) - \boldsymbol{y} \rangle_{p(\boldsymbol{y}|\boldsymbol{x})} &= 0 \\ \hat{\boldsymbol{y}}^*(\boldsymbol{x}) &= \langle \boldsymbol{y} \rangle_{p(\boldsymbol{y}|\boldsymbol{x})}, \end{aligned} \tag{2.14}$$

where the model parameters $\boldsymbol{\theta}$ are marginalized out and $\boldsymbol{x}$ are the observed variables. This is a well-known result in the Bayesian framework; that is, the PM coincides with the minimum mean square error estimator.

2.2.2 Maximum a Posteriori as Bayesian Optimal Estimation Function for Delta function Error

Next, we consider $\mathrm{Error}(\boldsymbol{y}, \boldsymbol{y}^*(\boldsymbol{x}))$ based on the Kronecker delta function for $\boldsymbol{y}$ and $\boldsymbol{y}^*(\boldsymbol{x})$, $\delta_{\boldsymbol{y},\boldsymbol{y}^*(\boldsymbol{x})}$, which takes 1 when $\boldsymbol{y} = \boldsymbol{y}^*(\boldsymbol{x})$ and 0 when $\boldsymbol{y} \neq \boldsymbol{y}^*(\boldsymbol{x})$:

$$\mathrm{Error}(\boldsymbol{y}, \boldsymbol{y}^*(\boldsymbol{x})) \equiv 1 - \delta_{\boldsymbol{y},\boldsymbol{y}^*(\boldsymbol{x})}. \tag{2.15}$$

The evaluation criterion is to minimize the population mean of the error function in Eq. (2.15), as

$$\underset{\boldsymbol{y}^*(\boldsymbol{x})}{\mathrm{argmin}} \left\langle 1 - \delta_{\boldsymbol{y},\boldsymbol{y}^*(\boldsymbol{x})} \right\rangle_{p(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})}. \tag{2.16}$$

We can then explicitly compute the optimal estimation function $\hat{\boldsymbol{y}}^*(\boldsymbol{x})$ as the maximum a posteriori (MAP):

$$\begin{aligned}
\hat{\boldsymbol{y}}^*(\boldsymbol{x}) &= \underset{\boldsymbol{y}^*(\boldsymbol{x})}{\mathrm{argmin}} \left\langle 1 - \delta_{\boldsymbol{y},\boldsymbol{y}^*(\boldsymbol{x})} \right\rangle_{p(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})} \\
&= \underset{\boldsymbol{y}^*(\boldsymbol{x})}{\mathrm{argmin}} \left\langle 1 - \delta_{\boldsymbol{y},\boldsymbol{y}^*(\boldsymbol{x})} \right\rangle_{p(\boldsymbol{y}|\boldsymbol{x})} \\
&= \underset{\boldsymbol{y}^*(\boldsymbol{x})}{\mathrm{argmax}} \left\langle \delta_{\boldsymbol{y},\boldsymbol{y}^*(\boldsymbol{x})} \right\rangle_{p(\boldsymbol{y}|\boldsymbol{x})} \\
&= \underset{\boldsymbol{y}^*(\boldsymbol{x})}{\mathrm{argmax}} \; p(\boldsymbol{y}^*(\boldsymbol{x})|\boldsymbol{x}) \\
&= \underset{\boldsymbol{y}}{\mathrm{argmax}} \; p(\boldsymbol{y}|\boldsymbol{x}),
\end{aligned} \tag{2.17}$$

where the model parameters $\boldsymbol{\theta}$ are marginalized out and $\boldsymbol{x}$ are the observed variables.

2.3 Computational Algorithms for Bayesian Optimal Estimation

The posterior distribution $p(\boldsymbol{y}|\boldsymbol{x})$ appearing in the optimal estimation functions derived in the above section is computationally feasible only when there are conjugate priors, that is, the prior is in the same distribution family as the posterior distribution, for all of the hidden variables and parameters other than observation variables in our designed model [13, 41]. This requirement is almost impossible to meet in most practical cases, wherein the models include multiple unknown parameters. Thus, below we introduce efficient approximate computational algorithms for $p(\boldsymbol{y}|\boldsymbol{x})$, i.e., the MCMC method and VB method.

2.3.1 Markov Chain Monte Carlo Method

The MCMC method [42–49] is a sampling-based approximation for the distribution. Given the joint distribution $p(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta})$ and observations $\boldsymbol{x}$, we can take T samples for $\boldsymbol{y}$, $\{\boldsymbol{y}^{(\tau)}\}_{\tau=1}^{T}$, from the posterior $p(\boldsymbol{y}|\boldsymbol{x})$ with the MCMC method, but without explicitly computing the posterior $p(\boldsymbol{y}|\boldsymbol{x})$. Using the samples $\{\boldsymbol{y}^{(\tau)}\}_{\tau=1}^{T}$, we can approximately compute the estimation function based on the posterior distribution $p(\boldsymbol{y}|\boldsymbol{x})$, e.g., by approximating the PM as the empirical mean of $\{\boldsymbol{y}^{(\tau)}\}_{\tau=1}^{T}$:

$$\int \boldsymbol{y}\ p(\boldsymbol{y}|\boldsymbol{x})\mathrm{d}\boldsymbol{y} \approx \frac{1}{T}\sum_{\tau=1}^{T}\boldsymbol{y}^{(\tau)}. \tag{2.18}$$

In this thesis, we use slice sampling [50] as the implementation of the MCMC method. Slice sampling is applicable to a wide variety of problems because it does not require an analytical computation of the conditional distributions like Gibbs sampling does or sensitive setting of the proposal distributions, as the Metropolis algorithm needs [50, 51].

Slice sampling takes samples from a distribution for some variable $\eta \in \mathbb{R}$. The distribution is proportional to some function $f(\eta)$ maintaining the computational feasibility. The key idea is to introduce an auxiliary variable $\xi \in \mathbb{R}$ and define a joint distribution over η and ξ. The joint distribution is designed such that the marginal density for η becomes the desired form that is proportional to $f(\eta)$:

$$p(\eta) = \int p(\eta, \xi)\mathrm{d}\xi = \frac{f(\eta)}{\int f(\eta)\mathrm{d}\eta}, \tag{2.19}$$

where the joint distribution $p(\eta, \xi)$ is uniform over the region $U = \{(\eta, \xi) \mid 0 < \xi < f(\eta)\}$ below the curve defined by $f(\eta)$, as

$$p(\eta, \xi) \equiv \begin{cases} \frac{1}{\int f(\eta)\mathrm{d}\eta}, & 0 < \xi < f(\eta), \\ 0, & \text{otherwise}, \end{cases} \tag{2.20}$$

The slice sampling is executed as follows: First, we sample ξ uniformly from $(0, f(\eta^{(\tau)}))$; thereby defining a horizontal *slice* $S = \{\eta^{(\tau+1)} \mid \xi < f(\eta^{(\tau+1)})\}$. Second, we sample a new point $\eta^{(\tau+1)}$ from part of the slice S. We repeat the first and second steps until a sufficient number of samples is obtained. Finally, to obtain samples only for η from $p(\eta)$, we ignore the samples for ξ.

The sampling scheme for η with the auxiliary variable ξ seems redundant compared with sampling only for η, but it is easier than Gibbs sampling and more efficient than the Metropolis algorithm.

When we sample multiple variables, such as $\boldsymbol{y}$, we alternately sample each variable by using slice sampling in the same manner as Gibbs sampling [13, 50, 51]. In our case,

we exploit the fact that the joint distribution $p(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta})$ is proportional to the posterior $p(\boldsymbol{y}, \boldsymbol{\theta}|\boldsymbol{x})$ for $\boldsymbol{y}$ and $\boldsymbol{\theta}$. This means that we can use the joint distribution $p(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta})$ as the objective $f(\boldsymbol{y}, \boldsymbol{\theta})$ of slice sampling for $\boldsymbol{y}$ and $\boldsymbol{\theta}$. We alternately sample each variable by using slice sampling. Then we use samples for $\boldsymbol{y}$ as the desired samples from the posterior $p(\boldsymbol{y}|\boldsymbol{x})$ of $\boldsymbol{y}$ given $\boldsymbol{x}$ simply by ignoring the samples for $\boldsymbol{\theta}$.

2.3.2 Variational Bayes

The VB method [52, 53] is a deterministic computational algorithm for finding a tractable distribution which approximates a intractable posterior distribution. If, for all of the hidden variables and parameters in the model, there are conjugate priors for some of the variables and parameters when the other variables and parameters are set to constant values, we can derive an efficient approximate computational algorithm for the posterior distribution using the VB method and its good performance is experimentally shown. In particular, for models such as the mixture model, hidden Markov model, Bayesian network, and fully-observed matrix factorization model, it has been reported that the VB method's approximation is sufficiently close to the true Bayesian optimal estimation in the KL-divergence sense [54–59]. The analysis of general cases, however, remains an open problem.

The starting point of the VB method is to introduce a trial distribution $q(\boldsymbol{z})$, where $\boldsymbol{z} = \{\boldsymbol{y}, \boldsymbol{\theta}\}$, that approximates the true posterior in a factorized form:

$$q(\boldsymbol{z}) \equiv \prod_i q(\boldsymbol{z}_i), \tag{2.21}$$

where $\boldsymbol{z}_i$ is the i-th subset of $\boldsymbol{z}$. Note that the distribution family of each factorized distribution is not limited. We then identify the optimal trial distribution that minimizes the KL divergence from the trial distribution $q(\boldsymbol{z})$ to the true posterior

distribution $p(\boldsymbol{z}|\boldsymbol{x})$, which becomes zero in the case of $q(\boldsymbol{z}) = p(\boldsymbol{z}|\boldsymbol{x})$, as

$$\hat{q}(\boldsymbol{z}) \equiv \underset{q(\boldsymbol{z})}{\operatorname{argmin}} D_{\mathrm{KL}}(q(\boldsymbol{z})\|p(\boldsymbol{z}|\boldsymbol{x})) \tag{2.22}$$

$$\begin{aligned}
&= \underset{q(\boldsymbol{z})}{\operatorname{argmin}} \left[- \left\langle \ln p(\boldsymbol{z}|\boldsymbol{x}) - \sum_i \ln q(\boldsymbol{z}_i) \right\rangle_{q(\boldsymbol{z})} \right] \\
&= \underset{q(\boldsymbol{z})}{\operatorname{argmin}} \left[- \left\langle \langle \ln p(\boldsymbol{z}|\boldsymbol{x}) \rangle_{\prod_{j\neq i} q(\boldsymbol{z}_j)} \right\rangle_{q(\boldsymbol{z}_i)} + \langle \ln q(\boldsymbol{z}_i) \rangle_{q(\boldsymbol{z}_i)} + \sum_{j\neq i} \langle \ln q(\boldsymbol{z}_j) \rangle_{q(\boldsymbol{z}_j)} \right] \\
&= \underset{q(\boldsymbol{z})}{\operatorname{argmin}} \left[- \langle \ln \tilde{p}_i(\boldsymbol{z}_i) - \ln q(\boldsymbol{z}_i) \rangle_{q(\boldsymbol{z}_i)} + \sum_{j\neq i} \langle \ln q(\boldsymbol{z}_j) \rangle_{q(\boldsymbol{z}_j)} \right] \\
&= \underset{q(\boldsymbol{z})}{\operatorname{argmin}} \left[D_{\mathrm{KL}}(q(\boldsymbol{z}_i)\|\tilde{p}_i(\boldsymbol{z}_i)) + \sum_{j\neq i} \langle \ln q(\boldsymbol{z}_j) \rangle_{q(\boldsymbol{z}_j)} \right],
\end{aligned} \tag{2.23}$$

where $\tilde{p}_i(\boldsymbol{z}_i)$ is defined as

$$\ln \tilde{p}_i(\boldsymbol{z}_i) \equiv \langle \ln p(\boldsymbol{z}|\boldsymbol{x}) \rangle_{\prod_{j\neq i} q(\boldsymbol{z}_j)} . \tag{2.24}$$

Since the above equations for $i = 1, 2, \ldots$ are a set of consistency conditions subject to the factorization of Eq. (2.21), the optimal trial distribution is

$$\hat{q}(\boldsymbol{z}) \propto \prod_i \tilde{p}_i(\boldsymbol{z}_i). \tag{2.25}$$

From the above result, they do not provide an explicit solution since i-th factor $\tilde{p}_i(\boldsymbol{z}_i)$ depends on the other factors $\{\tilde{p}_j(\boldsymbol{z}_j) \mid j \neq i\}$. Therefore, in a popular approach of VB [60], we solve the iterative updating equations as follows:

$$q^{(0)}(\boldsymbol{z}_i) \equiv p(\boldsymbol{z}_i), \tag{2.26}$$

$$q^{(t+1)}(\boldsymbol{z}_i) \propto \exp\langle \ln p(\boldsymbol{z}|\boldsymbol{x}) \rangle_{\prod_{j\neq i} q^{(t)}(\boldsymbol{z}_j)}, \tag{2.27}$$

where some $q^{(t+1)}(\boldsymbol{z}_j)$s are used instead of $q^{(t)}(\boldsymbol{z}_j)$s for the distribution on the right-hand side of (2.27). This depends on the hierarchical structure of the model. Similarly, some of the $q^{(0)}(\boldsymbol{z}_i)$s may not be necessary. We stop the VB iterations when a certain stopping condition is satisfied and take $\prod_i q^{(t+1)}(\boldsymbol{z}_i)$ at that time as the approximation of $p(\boldsymbol{z}|\boldsymbol{x})$. From Eq. (2.21), since $q(\boldsymbol{z})$ has been already factorized, we can simply use $q(\boldsymbol{z}_i)$ as the approximation of $p(\boldsymbol{z}_i|\boldsymbol{x})$, such as we can use $q(\boldsymbol{y})$ as the approximation of $p(\boldsymbol{y}|\boldsymbol{x})$.

2.3.3 Taylor Approximation for Variational Bayes

Although the VB method is a widely used general framework, its application is difficult in practice because it requires the conjugate modeling, that is, the modeling

of the distribution family of the prior distribution has to be the same as that of the posterior distribution, to reach an exact analytical solution in each step in Eq. (2.27). If we cannot obtain an exact analytical solution, the updating equations are computationally infeasible. This makes the framework of limited utility in many real-world situations, because it is difficult to appropriately design a model for complex real-world problems under such a modeling constraint.

We have found that simple Taylor approximations make the model conjugate and enable analytically exact expectations in each step of Eq. (2.27). To simplify the notation, we define the mean values of $\boldsymbol{z}_i$ over the trial distributions for the i-th subset of variables at step number t of the updates of VB as $\boldsymbol{\mu}_{\boldsymbol{z}_i}^{(t)}$. Specifically, when there are non-linear terms, $g(\boldsymbol{z})$, in the expectations in Eq. (2.27), whose specific form is revealed in the following chapters, we use a first-order Taylor approximation for $g(\boldsymbol{z})$. The non-linear term $g(\boldsymbol{z})$ is approximated around $\boldsymbol{z}_i = \boldsymbol{\mu}_{\boldsymbol{z}_i}^{(t)}$ as

$$g(\boldsymbol{z}) \approx g(\{\boldsymbol{z}_{\setminus i}, \boldsymbol{\mu}_{\boldsymbol{z}_i}^{(t)}\}) + (\boldsymbol{z}_i - \boldsymbol{\mu}_{\boldsymbol{z}_i}^{(t)})^\top \left.\frac{\partial g(\boldsymbol{z})}{\partial \boldsymbol{z}_i}\right|_{\boldsymbol{z}_i = \boldsymbol{\mu}_{\boldsymbol{z}_i}^{(t)}}. \tag{2.28}$$

Also, the non-linear term $g(\boldsymbol{z})$ can be approximated around $\ln z_i = \ln \mu_{z_i}^{(t)}$ as

$$g(\boldsymbol{z}) \approx g(\{z_{\setminus i}, \mu_{z_i}^{(t)}\}) + (\ln z_i - \ln \mu_{z_i}^{(t)}) \left.\frac{\partial g(\boldsymbol{z})}{\partial \ln z_i}\right|_{z_i = \mu_{z_i}^{(t)}}, \tag{2.29}$$

or approximated around $\ln(1 - z_i) = \ln(1 - \mu_{z_i}^{(t)})$ as

$$g(\boldsymbol{z}) \approx g(\{z_{\setminus i}, \mu_{z_i}^{(t)}\}) + (\ln(1 - z_i) - \ln(1 - \mu_{z_i}^{(t)})) \left.\frac{\partial g(\boldsymbol{z})}{\partial \ln(1 - z_i)}\right|_{z_i = \mu_{z_i}^{(t)}}. \tag{2.30}$$

Using these approximations, we can make the non-linear terms in the expectations in Eq. (2.27) linear or log-linear in the i-th subset of variables. This makes it so that many classes of model can be approximated as conjugate models in the VB method. For example, Eq. (2.28) is useful when we approximate models to be conjugate to Gaussian or Gamma prior and Eqs. (2.29) and (2.30) are useful when we approximate models to be conjugate to Gamma or Beta prior. The following chapters describe the specific derivations for each model of the tasks addressed in this thesis.

Note that Eq. (2.28) approximates $g(\boldsymbol{z})$ around the mean values of $\boldsymbol{z}_i$ over the trial distributions *in each step of the updates of the VB method*, $\boldsymbol{\mu}_{\boldsymbol{z}_i}^{(t)}$, as the expansion points. This results in a better approximation than by expanding around fixed points, such as prior mean $\boldsymbol{\mu}_{\boldsymbol{z}_i}^{(0)}$, because the Taylor approximation requires that the approximation point be close to the expansion point and $\boldsymbol{\mu}_{\boldsymbol{z}_i}^{(t)}$ is closer to $\boldsymbol{\mu}_{\boldsymbol{z}_i}^{(t+1)}$ than $\boldsymbol{\mu}_{\boldsymbol{z}_i}^{(0)}$ on average.

Chapter.3

Bayesian Traffic Flow Estimation

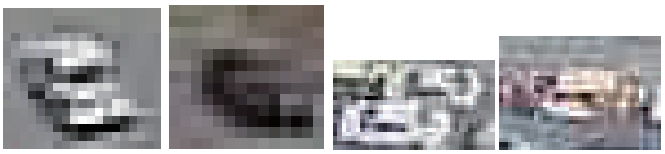
3.1 Introduction

Efficient control of traffic and city planning for better traffic flow are keys to economic growth and improving our lives. Intelligent Transportation Systems (ITS) offer such solutions by using the technologies based on machine learning, artificial intelligence, and data mining [61, 62]. The autonomous self-driving car [63] is a recent well-known example. Traffic volume and speed modeling is also a traditional and challenging research topic [62, 64–68]. There have been several studies on travel-time prediction [69, 70] and modeling of incidents and anomalies in traffic [71–73].

Systems for traffic monitoring, which is a fundamental part of ITS, sense the principle variables representing traffic, that is, flow, volume, and velocity, and if we can obtain two of them, the remaining one is uniquely determined, by using specialized hardware such as a Global Positioning System (GPS) and inductive loop sensors embedded in roads. They are the most trustworthy means to acquire the data.

Hardware sensors, however, are often bottlenecks blocking the introduction of ITS because they are immovable and expensive. Therefore, increasing attention is being paid to approaches using "non-intrusive" sensors because of their higher flexibility. Video surveillance is a natural approach to non-intrusive traffic monitoring. A number of cities (in advanced countries) have started using video cameras as non-intrusive traffic monitoring tools with high flexibility. For example, automatic license plate recognition is a recent successful application [74]. Special-purpose cameras producing high-quality images are used in most scenarios

In most of cities in developing countries, attention is increasingly being paid to the use of less expensive and more scalable Internet-linked cameras for city-wide traffic monitoring since special-purpose cameras are as costly as the roadside sensors [35–37, 76–79]. However, unlike the mature technologies of inductive loop sensors and special-purpose cameras, the use of such web-cameras is still challenging. Figure 3.1



(a) Example 1 (b) Example 2 (c) Example 3 (d) Example 4

Fig. 3.1 Vehicles in web-camera images [75].

shows vehicles in typical web-camera images [75]. The viewing angles tend to be quite poor and varied, image resolution is limited, and there are many occlusions. Even worse, the geometric configuration of these cameras differs significantly from camera to camera, requiring customized analysis for each camera. In addition, their poor frame-rates, due to limited network bandwidth, does not allow analyzing the time dependency of successive images for mitigating such inconveniences.

In this chapter, we tackle fundamental tasks of traffic monitoring, that are, extracting the traffic volume and velocity, by using the low-quality web-camera images. Traffic flow can be computed from them. For video-based traffic monitoring, two types of approaches have been proposed: (1) individual vehicle recognition, and (2) qualitative analysis. The first approach attempts to recognize individual vehicles in the images. Examples of current approaches include vehicle and non-vehicle classification [80, 81], and template matching [82–85]. Once all vehicles are identified in an image, counting and tracking vehicles for sensing traffic volume and velocity, respectively, are trivial tasks. However, this approach (vehicle recognition) is of limited utility in most real-world situations because the classifiers are quite sensitive to the training data sets. Also, the quality and frame-rates of web-camera images are generally lower than the assumptions made with current technologies. To address these shortcomings, the second approach (qualitative analysis) attempts to directly extract traffic-relevant metrics from images by skipping the expensive step of vehicle recognition, such as the local variance of pixels [76] and the total area that may correspond to moving objects [35–37, 77, 78, 86, 87]. However, most current approaches provide only qualitative metrics, such as a relative level of congestion, and are not capable of estimating absolute values for traffic volume, velocity, and flow [76, 86, 87]. When absolute values are required, we need to translate the obtained qualitative metrics into absolute values with a regression model and labeled training dataset. The dataset involves labeling a large amount of training data, which is time-consuming and costly. For the city-wide traffic monitoring services, we need to handle many web-cameras [35, 36, 77, 78]. The geometric configurations of these cameras differ; thus, customized labeled training data for every camera is required. This scenario moti-

vates us to use the unsupervised formulation, rather than conventional supervised approaches, due to the costs of preparing of the labeled training data.

We proposes an novel unsupervised approach for inexpensive traffic monitoring systems using only low-quality web-cameras. This approach estimates traffic volume and velocity only from the web-camera images without any labeled training data or the expensive steps of recognizing and tracking vehicles for the task. Thanks to the framework of Bayesian optimal estimation framework, our approach is quite robust against low-quality observations. We will show the sufficient accuracy and robustness of our approach in our experiments.

In Section 3.2, we discuss related work. We respectively derive the traffic-volume-estimation method and the traffic-velocity-estimation method in Sections 3.3 and 3.4. In Section 3.5, we evaluate these methods using artificial and real-world datasets. We discuss these methods in Section 3.6 and summarize our approach in Section 3.7.

3.2 Related Work

As mentioned in the Introduction to this chapter, current video-based-traffic monitoring systems proposed to date are categorized into two approaches, depending on whether or not they use individual vehicle recognition.

In the first approach (vehicle recognition) for traffic-volume estimation, once all of the vehicles are identified in an image, vehicle-counting is a trivial task. For this approach, previous studies used either image patch classification based on vehicle/non-vehicle classification [80, 81] or template matching. For template matching, examples include feature tracking for edges and lines characteristic of vehicles [82, 83], as well as targeted recognition of windshields [84] or headlights [84, 85]. These approaches clearly differ from our unsupervised approach in that they require a training dataset based on costly manual vehicle-counting and generally require high image quality and high frame rates.

For the traffic velocity estimation, the first approach, which is based on explicit vehicle recognition, attempts to directly sense velocity. Most previous studies on this approach tracked vehicles or track feature points in time-series of observations that correspond to the vehicle movements by using techniques for determining identical vehicles; not only video-based vehicle classification [80, 81], video-based feature tracking including tracking of edges and lines characteristic of vehicles [82, 83] and optical flow [88–90], and targeted recognition of windshields [84] and headlights [84, 85], but also the GPS [91, 92]. Once the vehicles have been tracked in the temporal sequences, we can obtain their moving distances and compute the traffic velocity by dividing the distance by the elapsed time. Moreover, there are variants of this approach based on

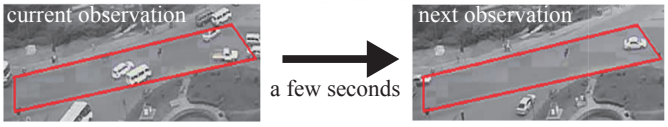


Fig. 3.2 At low sampling rates, only few vehicles appear in consecutive observations [75].

matching between consecutive frames, e.g., tracking pixels in consecutive observed images using cross-correlation of an image feature [93,94] or using matching of intensity profiles [95]. These variants are robust against variations in the expected conditions of a roadway scene and are computationally relatively inexpensive. In most cases, such video-based techniques require camera calibrations, as they are required to find the correct coordinate transform for obtaining traffic velocity. An algorithm to detect scene changes has been proposed [96], which can determine whether a camera has to be re-calibrated for video-based velocity estimation. On the other hand, some algorithms can use un-calibrated cameras for velocity estimation. They use parameters derived from distributions of known vehicle lengths [97,98], an estimation of the camera's position relative to the roadway [99], or a spatio-temporal map [100].

The second approach (qualitative analysis) for traffic monitoring, which does not rely on vehicle recognition in its first step, leads to a robust framework for traffic analysis. These approaches abandon deriving the exact number and velocity of vehicles to tackle the problem of image quality and to skip the preparation of the training images. They use image features such as the local variance of pixels [76] and the total area that may correspond to moving objects [35–37,77,78,86,87]. We then simply compute a relative level of congestion by dividing by the maximum available values of these features. Extraction of image features is easier than vehicle recognition and tracking, and it can work on images whose quality is lower than what would be required with the vehicle-recognition approach. However, when we need an absolute value for the number of vehicles as the input of possible applications, such as traffic simulators, we need to translate the features into the number and velocity of vehicles with a regression model and a labeled training dataset. Many traffic-velocity regression models have been proposed, such as linear model [101,102], log-linear model [103], exponential model [104–106], bell-shaped curve model [107], and stochastic model [108,109]. The task of velocity regression from the relative traffic density is a one-shot estimation for a single observation and does not use sequences of consecutive observations. Accordingly, this approach works for any sampling rate.

The requirements of these two types of approaches are often costly. This is particularly the case in cities in developing countries [77–79,110,111]. The first approach (vehicle recognition) requires that many vehicles be identified in consecutive observa-

tions. Their feasibilities are sensitive to the quality and sampling rate of the sensor. When using sensors with low sampling rates, such as web cameras, instead of expensive infrastructures, such as the special-purpose close-view cameras used for vehicle recognition, the number of vehicles that appear in consecutive observations is small, as shown in Fig. 3.2 [75]. Also, in a web-camera-based city-wide traffic monitoring scenario, it is unrealistic to assume a reasonable amount of training images for vehicle recognition since the configurations of web-cameras, which are typically represented by angles to and distance from the area of interest, relative directions to light sources, etc., differ from camera to camera. The second approach (qualitative analysis) enables us to use a variety of sensors whose sensor qualities are far too low for direct sensing of traffic, including inexpensive and non-intrusive ones such as web-cameras or mobile phones equipped with video and audio sensors [77–79], but we need to translate the obtained traffic density into traffic volume and velocity with a regression model and labeled training dataset. The dataset involves labeling a large amount of training data, which is time-consuming and costly. Therefore, we need a lightweight approach for traffic monitoring.

3.3 Bayesian Traffic Volume Estimation

In this section, we tackle the task that is extracting the current volume of traffic with inexpensive web-cameras [35, 36]. We proposes a probabilistic formulation on this problem by interpreting the problem as an unsupervised density estimation problem [35, 36].

Our concept is simple. We assume that the input observation is represented by a scalar feature, x. For the feature x, we learn a Gaussian mixture whose mixture index is equated with the count of the vehicles, d, in the observation (see Figure 3.3). To find the count, we pick the cluster of the highest likelihood given x. One technical challenge is how to associate the clusters with the count without any label information as to the count d. This is indeed not a trivial task because the cluster indexes are interchangeable in nature in the original Gaussian mixture. Our key contribution is to show that the stick-breaking process (SBP) [112] elegantly solves this challenge. Thanks to a VB formulation [52], the learning procedure is reduced to a simple iterative formula. Our formulation naturally can be applied to general object counting problem.

We will demonstrate that the accuracy and robustness of our approach without any labeled training data are comparable to those of supervised alternatives in experiments.

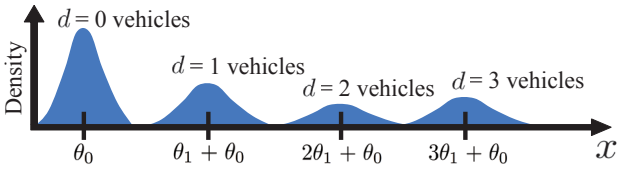


Fig. 3.3 Illustration of the key idea of mixture-based vehicle counting. The probability density for a feature x is represented as a Gaussian mixture which has the sequence of the mean of the feature restricted by the linear function of the count d.

3.3.1 Problem Setting

Suppose we are given N image features $\boldsymbol{x} \equiv \{x_1, x_2, \ldots, x_N\}$, which is the set of a scalar feature $x_n \in \mathbb{R}$ extracted from a observed image. Our task is to estimate the numbers of vehicles for the N images, given $\boldsymbol{x}$ without any labeled training data.

To represent the count of vehicles, we define an indicator vector $\boldsymbol{h}$ based on the 1-of-K notation. For example, if $\boldsymbol{h} = [1, 0, 0, 0, ...]^\top$ and $[0, 0, 1, 0, ...]^\top$, the number of vehicles are zero and two, respectively. Since we do not know the maximum number of vehicles in advance, in spite of the general term of "1-of-K", we assume that the dimension of $\boldsymbol{h}$ is infinity, $\boldsymbol{h} \in \{0, 1\}^\infty$, $\sum_{d=0}^{\infty} h_d = 1$. Let us denote the number of vehicles in the n-th image as $\boldsymbol{h}_n \in \{0, 1\}^\infty$, which is not directly observed. Then the set of the number of vehicles for the N images is represented as $\boldsymbol{H} \equiv \{\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_N\} \in \{0, 1\}^{\infty \times N}$. Now our goal is to estimate $\boldsymbol{H}$ from $\boldsymbol{x}$.

3.3.2 Maximum a Posteriori Estimation for Traffic Volume

We formalize this estimation problem as the estimates of $\boldsymbol{h}_n$ by the estimation function, $\boldsymbol{h}_n^*(\boldsymbol{x})$, to which the observed variables $\boldsymbol{x}$ have been input.

We first consider the evaluation criterion based on the Bayesian perspective for this task. Since the number of vehicles is a natural number, we define the error function $\mathrm{Error}(\boldsymbol{h}_n, \boldsymbol{h}_n^*(\boldsymbol{x}))$ for the task as the Kronecker delta function between $\boldsymbol{h}_n$ and the estimates by the estimation function $\boldsymbol{h}_n^*(\boldsymbol{x})$:

$$\mathrm{Error}(\boldsymbol{h}_n, \boldsymbol{h}_n^*(\boldsymbol{x})) \equiv 1 - \delta_{\boldsymbol{h}_n, \boldsymbol{h}_n^*(\boldsymbol{x})}, \tag{3.1}$$

Using the model parameters $\boldsymbol{\phi}_{\boldsymbol{x}}$ and $\boldsymbol{\phi}_{\boldsymbol{H}}$ which are explicitly defined later, we define the evaluation criterion for this task as the minimization of the population mean of the error function Eq. (3.1):

$$\underset{\boldsymbol{h}_n^*(\boldsymbol{x})}{\mathrm{argmin}} \left\langle 1 - \delta_{\boldsymbol{h}_n, \boldsymbol{h}_n^*(\boldsymbol{x})} \right\rangle_{p(\boldsymbol{H}, \boldsymbol{x}, \boldsymbol{\phi}_{\boldsymbol{H}}, \boldsymbol{\phi}_{\boldsymbol{x}})} \tag{3.2}$$

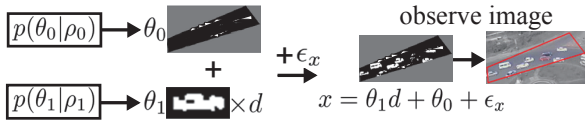


Fig. 3.4 The observation process.

Then, we can derive the optimal estimation function using the result in Eq. (2.17) as the MAP

$$\begin{aligned}\hat{\boldsymbol{h}}_n^*(\boldsymbol{x}) &= \underset{\boldsymbol{h}_n^*(\boldsymbol{x})}{\operatorname{argmin}} \left\langle 1 - \delta_{\boldsymbol{h}_n, \boldsymbol{h}_n^*(\boldsymbol{x})} \right\rangle_{p(\boldsymbol{H}, \boldsymbol{x}, \boldsymbol{\phi}_{\boldsymbol{H}}, \boldsymbol{\phi}_{\boldsymbol{x}})} \\ &= \underset{\boldsymbol{h}_n}{\operatorname{argmax}} \; p(\boldsymbol{h}_n|\boldsymbol{x}) \\ &= \underset{\boldsymbol{h}_n}{\operatorname{argmax}} \int p(\boldsymbol{x}|\boldsymbol{H}, \boldsymbol{\phi}_{\boldsymbol{x}}) p(\boldsymbol{H}|\boldsymbol{\phi}_{\boldsymbol{H}}) p(\boldsymbol{\phi}_{\boldsymbol{x}}, \boldsymbol{\phi}_{\boldsymbol{H}}) \mathrm{d}\boldsymbol{\phi}_{\boldsymbol{x}} \mathrm{d}\boldsymbol{\phi}_{\boldsymbol{H}} \prod_{m \neq n} \mathrm{d}\boldsymbol{h}_m,\end{aligned} \tag{3.3}$$

where the posterior distribution $p(\boldsymbol{h}_n|\boldsymbol{x})$ represents the probability distribution of the number of vehicles $\boldsymbol{h}_n$ given $\boldsymbol{x}$. As shown in Eq. (3.3), the posterior distribution consists of the observation model $p(\boldsymbol{x}|\boldsymbol{H}, \boldsymbol{\phi}_{\boldsymbol{x}})$ and the prior models $p(\boldsymbol{H}|\boldsymbol{\phi}_{\boldsymbol{H}})p(\boldsymbol{\phi}_{\boldsymbol{x}}, \boldsymbol{\phi}_{\boldsymbol{H}})$. In the following subsection, we propose them.

Notice that this formulation requires no training data. This is extremely useful in practice, since we can avoid the quite time-consuming and costly step of manual vehicle-counting and labeling.

3.3.3 Probabilistic Hidden Structure Modeling for Traffic Volume

We assume that each image is represented by a scalar feature called Vehicle Pixel Area (VPA). In this case, $\boldsymbol{x} \in \mathbb{R}^N$. The VPA feature of an image is computed using the following steps: First, as a pre-process, we use a median filter for noise reduction and also subtract from the pixel values of each image the median of its pixels to handle the variation in luminance, such as the considerable luminance difference between rainy and clear days or between day and night. Then we binarize the image using a discriminant analysis technique [113], which is a traditional method but still used as a state-of-the-art of image-binarization [114,115], and count the number of white pixels which may correspond to vehicles. This is a raw score for the image feature. Finally, the raw score is normalized to be in $[-1, 1]$ by dividing by half of the maximum raw score in the N images and subtracting 1. This feature extraction algorithm will work for any frame-rate, even for still images, and is quite robust against poor image quality.

Figure 3.4 illustrates the proposed observation process for the VPA. As shown in Figure 3.4, apart from the additive Gaussian noise represented by ϵ_x, we assume that the VPA feature x is a linear function of the number of vehicles d as $\theta_1 d + \theta_0$. The

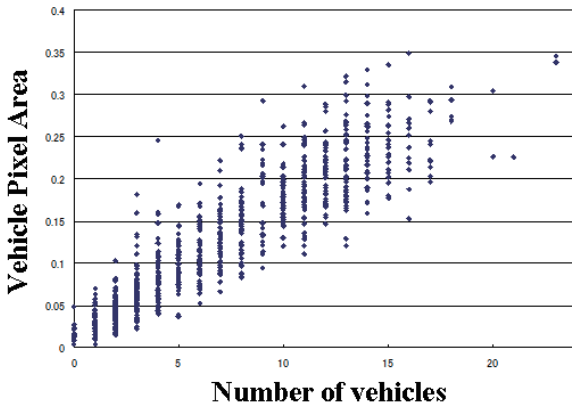


Fig. 3.5 The validity of the linear model for VPA.

precision parameter of the Gaussian noise, $\beta > 0$, and two parameters, $\theta_0 \in \mathbb{R}$ and $\theta_1 \in \mathbb{R}$, are to be learned from the data. From these assumption, the observation model $p(x|\boldsymbol{h}, \boldsymbol{\theta})$ is defined as a Gaussian mixture model (GMM), whose d-th mixture component is responsible for x having d number of objects through a restriction on its mean parameter as shown in Figure 3.3:

$$p(x|h_d = 1, \boldsymbol{\theta}, \beta) \equiv \mathcal{N}\left(x\middle|\theta_1 d + \theta_0, \beta^{-1}\right). \tag{3.4}$$

Since the count for the observation can take on any arbitrary *natural number*, the proposed GMM has an infinite number of mixture components as

$$\begin{aligned} p(x|\boldsymbol{h}, \boldsymbol{\theta}, \beta) &\equiv \prod_{d=0}^{\infty} \mathcal{N}\left(x\middle|\theta_1 d + \theta_0, \beta^{-1}\right)^{h_d} \\ &= \frac{\exp\left(-\frac{\beta}{2}\sum_{d=0}^{\infty} h_d (x - \theta_1 d - \theta_0)^2\right)}{(2\pi\beta^{-1})^{\frac{1}{2}}}. \end{aligned} \tag{3.5}$$

This is an infinite GMM with the specific restriction on its mean value given by the linear function of the count d.

The joint observation model over all of the N images is

$$\begin{aligned} p(\boldsymbol{x}|\boldsymbol{H}, \theta_0, \theta_1, \beta) &\equiv \prod_{n=1}^{N}\prod_{d=0}^{\infty} \mathcal{N}(x_n|\theta_1 d + \theta_0, \beta^{-1})^{h_{n,d}} \\ &= \frac{\exp(-\frac{\beta}{2}\sum_{n=1}^{N}\sum_{d=0}^{\infty} h_{n,d}(x_n - \theta_1 d - \theta_0)^2)}{(2\pi\beta^{-1})^{\frac{N}{2}}}. \end{aligned} \tag{3.6}$$

The linear assumption we made is based on the observation shown in Figure 3.5. This obviously shows that a nonlinear function does not give any significant improvement. This comes from the fact that we focused on images taken with cameras located far away from the roads to allow city-wide traffic monitoring. In this scenario, we do

not have to explicitly take account of the effect of the nonlinear relationship between x and d, such as the effect of perspective projection.

In the observation model defined above, we have four parameters, $\boldsymbol{H}$, θ_0, θ_1, and β. We now define prior distributions for these parameters according to Eq. (3.3).

First, we introduce an SBP prior [112] for $\boldsymbol{H}$ using an additional parameter $\boldsymbol{v}$ ($0 \leq v_d \leq 1$) as

$$p(\boldsymbol{H}|\boldsymbol{v}) \equiv \prod_{n=1}^{N} \prod_{d=0}^{\infty} \left(v_d \prod_{k=0}^{d-1} (1 - v_k) \right)^{h_{n,d}}. \tag{3.7}$$

In general, SBPs have the property of automatic determination of model complexity. In our context, the SBP is useful to remove the redundant clusters, so that we can obtain the simplest model that fits the data best.

From Eq. (3.7), we see that, for each component with $h_{n,d} = 1$, the probability is given by successively breaking a unit length stick into an infinite number of pieces. The size of each piece is the product of the rest of the stick and an independent generating value v_d.

Regarding the SBP parameter $\boldsymbol{v}$, we use the hyperprior distribution [116, 117]

$$p(\boldsymbol{v}|\alpha) \equiv \prod_{d=0}^{\infty} \mathrm{Beta}(v_d|1, \alpha), \tag{3.8}$$

where α (> 0) is a hyperparameter controlling the degree of sparseness of SBP and also to be learned. Note that in the SBP formulation with the VB method the infinite dimension of the model is replaced with a finite (large) dimension when implementing the algorithm (see Section 3.3.4).

Regarding prior distributions for θ_0, θ_1, and β, we simply use the conjugate priors:

$$\begin{aligned} &p(\theta_0|\rho_0) \equiv \mathcal{N}(\theta_0|\mu_{\theta_0}^{(0)}, \rho_0), \quad p(\theta_1|\rho_1) \equiv \mathcal{N}(\theta_1|\mu_{\theta_1}^{(0)}, \rho_1), \quad \text{and} \\ &p(\beta) \equiv \mathrm{Gamma}(\beta|a_\beta^{(0)}, b_\beta^{(0)}), \end{aligned} \tag{3.9}$$

where the parameters $\mu_{\theta_0}^{(0)}$, $\mu_{\theta_1}^{(0)}$, $a_\beta^{(0)}$, and $b_\beta^{(0)}$ are treated as input parameters given as a part of the model (see the section on the experimental results). Here the superscript (0) indicates that these parameters are used for the initial values of the VB procedure.

Finally, we define the hyperprior distributions for α, ρ_0, and ρ_1 using the conjugate priors:

$$\begin{aligned} p(\rho_0, \rho_1, \alpha) \equiv &\mathrm{Gamma}(\rho_0|a_{\rho_0}^{(0)}, b_{\rho_0}^{(0)}) \\ &\times \mathrm{Gamma}(\rho_1|a_{\rho_1}^{(0)}, b_{\rho_1}^{(0)}) \, \mathrm{Gamma}(\alpha|a_\alpha^{(0)}, b_\alpha^{(0)}), \end{aligned} \tag{3.10}$$

where $a_{\rho_0}^{(0)}$, $b_{\rho_0}^{(0)}$,$a_{\rho_1}^{(0)}$, $b_{\rho_1}^{(0)}$,$a_\alpha^{(0)}$, and $b_\alpha^{(0)}$ are input parameters. For the input parameters we actually used, see the section on the experimental results.

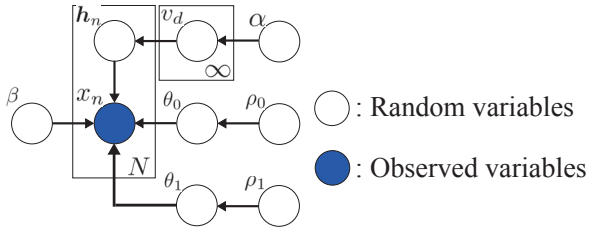


Fig. 3.6 The generative model.

The joint distribution for all of the random variables $\boldsymbol{z} \equiv \{\boldsymbol{H}, \theta_0, \theta_1, \beta, \rho_0, \rho_1, \boldsymbol{v}, \alpha\}$ as well as $\boldsymbol{x}$ can now be explicitly given as

$$\begin{aligned} p(\boldsymbol{x}, \boldsymbol{z}) &= p(\boldsymbol{x}|\boldsymbol{H}, \theta_0, \theta_1, \beta) p(\boldsymbol{H}|\boldsymbol{v}) \\ &\quad \times p(\theta_0|\rho_0) p(\theta_1|\rho_1) p(\beta) p(\boldsymbol{v}|\alpha) p(\rho_0, \rho_1, \alpha). \end{aligned} \tag{3.11}$$

We can derive the relevant marginal and conditional distributions such as the posterior distribution $p(\boldsymbol{z}|\boldsymbol{x})$ in terms of this joint distribution.

Figure 3.6 summarizes the proposed generative model including all of the parameters. First, α, ρ_0, ρ_1, and β are generated, after which $\boldsymbol{v}$, θ_0, and θ_1 are generated using α, ρ_0, and ρ_1, and then the number of vehicles $\boldsymbol{h}_n$ is generated using $\boldsymbol{v}$. Finally, the observation variable x_n is generated according to the observation process using $\boldsymbol{h}_n$, β, θ_0, and θ_1. The dimension of the parameters of the proposed infinite GMM is *not* infinity, which is in contrast to previous nonparametric Bayesian formulations [112, 116, 118, 119]. This makes it possible to give a special meaning to the value of the individual cluster centers.

3.3.4 Variational Bayes Algorithm for Maximum a Posteriori Estimation

As mentioned earlier, our goal is to obtain $\hat{\boldsymbol{h}}_n^*(\boldsymbol{x})$ in Eq. (3.3). While it is not possible to obtain an exact analytical solution for $p(\boldsymbol{h}_n|\boldsymbol{x})$ in $\hat{\boldsymbol{h}}_n^*(\boldsymbol{x})$, an approximated analytic solution can be found through a VB algorithm [52].

According to the formulation in Chapter 2, we assume a trial distribution $q(\boldsymbol{z})$ that approximates the true posterior in a factorized form:

$$q(\boldsymbol{z}) \equiv q(\boldsymbol{H}) q(\theta_0, \theta_1) q(\beta, \rho_0, \rho_1, \boldsymbol{v}) q(\alpha). \tag{3.12}$$

Then we identify the optimal trial distribution that minimizes the KL divergence from the trial distribution to the true posterior distribution $D_{\mathrm{KL}}(q(\boldsymbol{z}) \| p(\boldsymbol{z}|\boldsymbol{x}))$. Here, for efficient implementation of the VB algorithm with SBP, we replace the infinite dimension of the model with the training data size N, which is the maximum resolution of the observations [116].

From Eqs. (2.26), (2.27), and (3.6) - (3.11), the trial distribution of $\boldsymbol{H}$ is given as

$$q^{(t+1)}(\boldsymbol{H}) = \prod_{n=1}^{N} q^{(t+1)}(\boldsymbol{h}_n) = \prod_{n=1}^{N} \text{Categorical}(\boldsymbol{h}_n | \boldsymbol{\mu}_{\boldsymbol{h}_n}^{(t+1)}), \tag{3.13}$$

$$\text{where} \quad \boldsymbol{\mu}_{\boldsymbol{h}_n}^{(t+1)} \equiv [\mu_{h_{n,1}}^{(t+1)}, \mu_{h_{n,2}}^{(t+1)}, ...\mu_{h_{n,\infty}}^{(t+1)}]. \tag{3.14}$$

The trial distributions of θ_0, θ_1, β, ρ_0. ρ_1, $\boldsymbol{v}$, and α are given as:

$$q^{(t+1)}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} | \boldsymbol{\mu}_{\boldsymbol{\theta}}^{(t+1)}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{(t+1)}), \quad \text{where} \quad \boldsymbol{\theta} \equiv [\theta_0, \theta_1], \tag{3.15}$$

$$\begin{aligned} &q^{(t+1)}(\beta, \rho_0, \rho_1, \boldsymbol{v}) = \\ &\text{Gamma}(\beta | a_\beta^{(t+1)}, b_\beta^{(t+1)}) \, \text{Gamma}(\rho_0 | a_{\rho_0}^{(t+1)}, b_{\rho_0}^{(t+1)}) \\ &\quad \times \text{Gamma}(\rho_1 | a_{\rho_1}^{(t+1)}, b_{\rho_1}^{(t+1)}) \left[\prod_{d=0}^{\infty} \text{Beta}(v_d | a_{v_d}^{(t+1)}, b_{v_d}^{(t+1)}) \right], \text{ and} \end{aligned} \tag{3.16}$$

$$q^{(t+1)}(\alpha) = \text{Gamma}(\alpha | a_\alpha^{(t+1)}, b_\alpha^{(t+1)}). \tag{3.17}$$

Using the mean values of the hyperparameters β, ρ_0, ρ_1, and α over the trial distributions $q^{(t)}(\beta, \rho_0, \rho_1, \alpha)$, $\mu_\beta^{(t)} = \frac{a_\beta^{(t)}}{b_\beta^{(t)}}$, $\mu_{\rho_0}^{(t)} = \frac{a_{\rho_0}^{(t)}}{b_{\rho_0}^{(t)}}$, $\mu_{\rho_1}^{(t)} = \frac{a_{\rho_1}^{(t)}}{b_{\rho_1}^{(t)}}$, $\mu_\alpha^{(t)} = \frac{a_\alpha^{(t)}}{b_\alpha^{(t)}}$, we can analytically compute the parameters at step $t+1$ in Eqs. (3.13) - (3.17):

$$\mu_{h_{n,d}}^{(t+1)} = \frac{\exp\left[[\boldsymbol{c_v}]_d - \frac{1}{2}\mu_\beta^{(t)} c_{n,d}\right]}{\sum_{m=0}^{N-1} \exp\left[[\boldsymbol{c_v}]_m - \frac{1}{2}\mu_\beta^{(t)} c_{n,m}\right]}, \tag{3.18}$$

$$\begin{aligned} c_{n,d} \equiv &\left(x_n - d[\boldsymbol{\mu}_{\boldsymbol{\theta}}^{(t)}]_1 - [\boldsymbol{\mu}_{\boldsymbol{\theta}}^{(t)}]_0\right)^2 + d^2[\Sigma_{\boldsymbol{\theta}}^{(t)}]_{1,1} + [\Sigma_{\boldsymbol{\theta}}^{(t)}]_{0,0} \\ &+ d[\Sigma_{\boldsymbol{\theta}}^{(t)}]_{0,1} + d[\Sigma_{\boldsymbol{\theta}}^{(t)}]_{1,0}, \end{aligned} \tag{3.19}$$

$$\begin{aligned} [\boldsymbol{c_v}]_d \equiv &\psi(a_{v_d}^{(t)}) - \psi(a_{v_d}^{(t)} + b_{v_d}^{(t)}) \\ &+ \left[\sum_{k=0}^{d-1} \psi(b_{v_k}^{(t)}) - \psi(a_{v_k}^{(t)} + b_{v_k}^{(t)})\right], \end{aligned} \tag{3.20}$$

where ψ is the digamma function.

$$\begin{aligned} \boldsymbol{\mu}_{\boldsymbol{\theta}}^{(t+1)} &= \boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{(t+1)} \left[\mu_\beta^{(t)} \boldsymbol{c_x} + \boldsymbol{c}_{\boldsymbol{\mu_\theta}}\right], \\ \boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{(t+1)} &= \left[\mu_\beta^{(t)} \boldsymbol{C_h} + \boldsymbol{C}_\rho\right]^{-1}, \end{aligned} \tag{3.21}$$

$$[\boldsymbol{c_x}]_0 \equiv \sum_{d=0}^{N-1} \sum_{n=1}^{N} \mu_{h_{n,d}}^{(t)} x_n, \qquad [\boldsymbol{c_x}]_1 \equiv \sum_{d=0}^{N-1} \sum_{n=1}^{N} d\mu_{h_{n,d}}^{(t)} x_n, \tag{3.22}$$

$$[\boldsymbol{c}_{\boldsymbol{\mu}_{\boldsymbol{\theta}}}]_0 \equiv \mu_{\rho_0}^{(t)}\mu_{\theta_0}^{(0)}, \quad [\boldsymbol{c}_{\boldsymbol{\mu}_{\boldsymbol{\theta}}}]_1 \equiv \mu_{\rho_1}^{(t)}\mu_{\theta_1}^{(0)}, \tag{3.23}$$

$$[\boldsymbol{C}_{\boldsymbol{h}}]_{0,0} \equiv \sum_{d=0}^{N-1}\sum_{n=1}^{N}\mu_{h_{n,d}}^{(t)}, \quad [\boldsymbol{C}_{\boldsymbol{h}}]_{1,1} \equiv \sum_{d=0}^{N-1}\sum_{n=1}^{N}d^2\mu_{h_{n,d}}^{(t)}, \tag{3.24}$$

$$[\boldsymbol{C}_{\boldsymbol{h}}]_{0,1} \equiv \sum_{d=0}^{N-1}\sum_{n=1}^{N}d\mu_{h_{n,d}}^{(t)}, \quad [\boldsymbol{C}_{\boldsymbol{h}}]_{1,0} \equiv \sum_{d=0}^{N-1}\sum_{n=1}^{N}d\mu_{h_{n,d}}^{(t)}, \tag{3.25}$$

$$[\boldsymbol{C}_{\rho}]_{0,0} \equiv \mu_{\rho_0}^{(t)}, \quad [\boldsymbol{C}_{\rho}]_{1,1} \equiv \mu_{\rho_1}^{(t)}, \quad [\boldsymbol{C}_{\rho}]_{0,1}, [\boldsymbol{C}_{\rho}]_{0,1} \equiv 0, \tag{3.26}$$

$$a_{\beta}^{(t+1)} = a_{\beta}^{(0)} + \frac{1}{2}N, \;\; b_{\beta}^{(t+1)} = b_{\beta}^{(0)} + \frac{1}{2}\sum_{d=0}^{N-1}\sum_{n=1}^{N}\mu_{h_{n,d}}^{(t)}c_{n,d}, \tag{3.27}$$

$$\begin{aligned} a_{\rho_0}^{(t+1)} &= a_{\rho_0}^{(0)} + \frac{1}{2}, \\ b_{\rho_0}^{(t+1)} &= b_{\rho_0}^{(0)} + \frac{1}{2}\left(\left([\mu_{\boldsymbol{\theta}}^{(t)}]_0 - \mu_{\theta_0}^{(0)}\right)^2 + [\Sigma_{\boldsymbol{\theta}}^{(t)}]_{0,0}\right), \end{aligned} \tag{3.28}$$

$$\begin{aligned} a_{\rho_1}^{(t+1)} &= a_{\rho_1}^{(0)} + \frac{1}{2}, \\ b_{\rho_1}^{(t+1)} &= b_{\rho_1}^{(0)} + \frac{1}{2}\left(\left([\mu_{\boldsymbol{\theta}}^{(t)}]_1 - \mu_{\theta_1}^{(0)}\right)^2 + [\Sigma_{\boldsymbol{\theta}}^{(t)}]_{1,1}\right), \end{aligned} \tag{3.29}$$

$$a_{v_d}^{(t+1)} = 1 + \sum_{n=1}^{N}\mu_{h_{n,d}}^{(t)}, \;\; b_{v_d}^{(t+1)} = \mu_{\alpha}^{(t)} + \sum_{n=1}^{N}\sum_{k=d+1}^{N-1}\mu_{h_{n,k}}^{(t)}, \tag{3.30}$$

$$\begin{aligned} a_{\alpha}^{(t+1)} &= a_{\alpha}^{(0)} + N, \\ b_{\alpha}^{(t+1)} &= b_{\alpha}^{(0)} - \sum_{d=0}^{N-1}\psi(b_{v_d}^{(t)}) - \psi(a_{v_d}^{(t)} + b_{v_d}^{(t)}). \end{aligned} \tag{3.31}$$

To solve the updating equations, (3.13) - (3.17), we first compute $q^{(t+1)}(\boldsymbol{H}, \theta_0, \theta_1, \beta)$ using $q^{(t)}(\rho_0, \rho_1, \boldsymbol{v}, \alpha)$. Then we compute $q^{(t+1)}(\rho_0, \rho_1, \boldsymbol{v})$ using $q^{(t)}(\boldsymbol{H}, \theta_0, \theta_1, \beta)q^{(t)}(\alpha)$. Finally we compute $q^{(t+1)}(\alpha)$ using $q^{(t+1)}(\boldsymbol{H}, \theta_0, \theta_1, \beta, \rho_0, \rho_1, \boldsymbol{v})$. Here, we simply compute only the parameters of these distributions, thanks to conjugate modeling. For the initial parameters for θ_0, θ_1, β, ρ_0, ρ_1, $\boldsymbol{v}$, and α, we use the same values as those of the corresponding priors. In practice, we stop the VB iterations when this condition is satisfied:

$$\frac{(D_{\mathrm{KL}}(p(\boldsymbol{z})\|q^{(t+1)}(\boldsymbol{z})) - D_{\mathrm{KL}}(p(\boldsymbol{z})\|q^{(t)}(\boldsymbol{z})))^2}{D_{\mathrm{KL}}(p(\boldsymbol{z})\|q^{(t)}(\boldsymbol{z}))^2} < 10^{-10}. \tag{3.32}$$

After the above stopping condition is satisfied, we obtain the final outcome $q(\boldsymbol{h}_n)$ directly, which corresponds to an approximation of the learned posterior $p(\boldsymbol{h}_n|\boldsymbol{x})$ since the trial distribution q has been factorized as shown in Eq. (3.12). From Eq.(3.3), using the learned $q(\boldsymbol{h}_n)$, we can estimate the numbers of vehicles as

$$\hat{\boldsymbol{h}}_n^* \approx \underset{\boldsymbol{h}_n}{\operatorname{argmax}} \; q(\boldsymbol{h}_n). \tag{3.33}$$

Although the proposed method is formulated as the batch algorithm for N observed images, we can extend it to an online estimation model of $\boldsymbol{h}$ for a new observation x approximately as

$$\begin{aligned}\hat{\boldsymbol{h}} = \underset{\boldsymbol{h}}{\operatorname{argmax}} \;\; & \exp\langle \ln p(x|\boldsymbol{h}, \theta_0, \theta_1, \beta) p(\boldsymbol{h}|\boldsymbol{v}) \\ & \times p(\theta_0|\rho_0) p(\theta_1|\rho_1) p(\beta) p(\boldsymbol{v}|\alpha) p(\rho_0, \rho_1, \alpha) \rangle_{q^{(\infty)}(\theta_0, \theta_1, \beta, \boldsymbol{v}, \rho_0, \rho_1, \alpha)}.\end{aligned} \tag{3.34}$$

3.4 Bayesian Traffic Velocity Estimation

In this section, we propose a new approach in which the traffic velocity is estimated only from observed temporal-sequences of the numbers of vehicles, which can be obtained from web-camera images by using the method in the above section [37]. The proposed method does not require tracking any vehicles or using any labeled training data.

We use the fact that the some proportion of vehicles in two or more consecutive observations will be the same vehicles. The proportion will increase as the traffic velocity v decreases, and it directly represents the correlation between the numbers of vehicles in the consecutive observations. On the basis of the above fact, we first propose an observation model for observations conditioned on the traffic velocity v. Then, we estimate the traffic velocity through the density estimation of the model given the observations. This estimation task is an unsupervised one without using any labeled training data. Since our method does not need to track any vehicles, it can work with low quality and inexpensive sensors with low sampling rates, such as one observation every several seconds. Our approach naturally can be applied to general traffic velocity estimation problem using any data sources, such as inductive loop and GPS.

We will demonstrate that the accuracy and robustness of our approach without tracking any vehicles or using any labeled training data are good enough for our traffic monitoring application using real-world data in experiments.

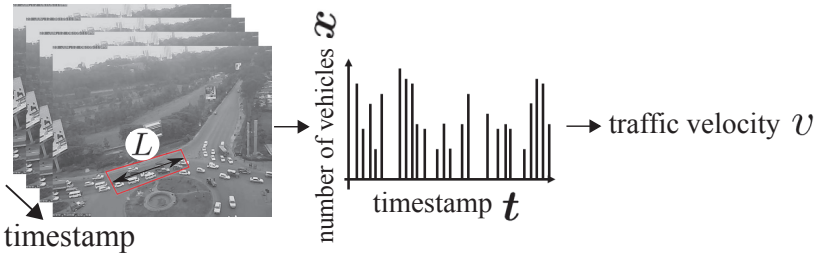


Fig. 3.7 Outline of the traffic velocity estimation problem.

3.4.1 Problem Setting

Let us define the task. We repeatedly observe the numbers of vehicles $\boldsymbol{x} \equiv [x_1, x_2, ..., x_N]^\top \in \mathbb{N}^N$ on a certain road area at time $\boldsymbol{t} \equiv [t_1, t_2, ..., t_N]^\top \in \mathbb{R}^N$ $(t_1 < t_2 < ... < t_N)$, as shown in Fig. 3.7. Note that the time intervals of $\boldsymbol{t}$ generally differ. For the length of the road area in which $\boldsymbol{x}$ is observed, $L > 0$ is known, and the road area has no intersections or branches. We refer to the road area as the observation area.

Our goal is to estimate the average traffic velocity $v \geq 0$ throughout the observations only from the available data $\boldsymbol{x}$ without tracking vehicles or using any labeled training data.

3.4.2 Posterior Mean Estimation for Traffic Velocity

We formalize this estimation problem as the estimates of v by the estimation function, $v^*(\boldsymbol{x})$, to which the observed variables $\boldsymbol{x}$ have been input.

We first consider the evaluation criterion based on the Bayesian perspective for this task. Since the velocity is a positive real number, the use of an all-or-none type error, such as the Dirac delta or Kronecker delta function, is nonsensical, whereas the squared L2-norm error (mean square error) is a conventional way of doing so. We define the error function $\mathrm{Error}(v, v^*(\boldsymbol{x}))$ for the task as the squared difference between v and the estimate by the estimation function $v^*(\boldsymbol{x})$:

$$\mathrm{Error}(v, v^*(\boldsymbol{x})) \equiv \|v - v^*(\boldsymbol{x})\|_2^2, \tag{3.35}$$

Using the model parameters $\boldsymbol{\phi}_{\boldsymbol{x}}$ and $\boldsymbol{\phi}_v$ which are explicitly defined later, we define the evaluation criterion as the minimization of the population mean of the error function Eq. (3.35):

$$\underset{v^*(\boldsymbol{x})}{\mathrm{argmin}} \left\langle \|v - v^*(\boldsymbol{x})\|_2^2 \right\rangle_{p(v, \boldsymbol{x}, \boldsymbol{\phi}_{\boldsymbol{x}}, \boldsymbol{\phi}_v)}. \tag{3.36}$$

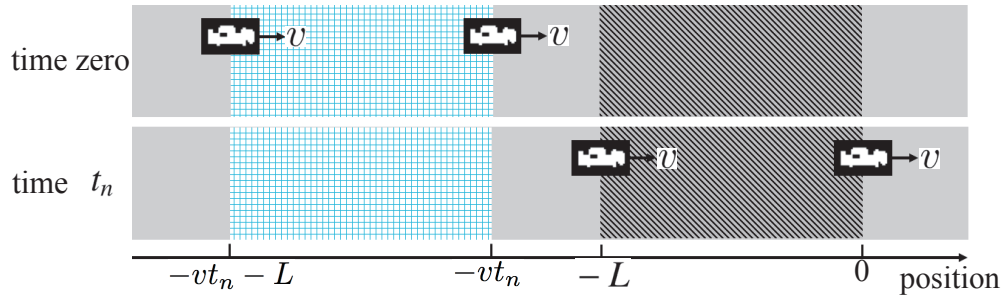


Fig. 3.8 Positions of observed vehicles at time zero and t_n.

Then, we can derive the optimal estimation function using the result in Eq. (2.14) as the PM,

$$\begin{aligned}\hat{v}^*(\boldsymbol{x}) &= \underset{v^*(\boldsymbol{x})}{\operatorname{argmin}} \left\langle \|v - v^*(\boldsymbol{x})\|_2^2 \right\rangle_{p(v,\boldsymbol{x},\boldsymbol{\phi}_{\boldsymbol{x}},\boldsymbol{\phi}_v)} \\ &= \int v\, p(v|\boldsymbol{x}) \mathrm{d}v,\end{aligned} \tag{3.37}$$

where the posterior distribution $p(v|\boldsymbol{x})$ represents the probability distribution of the traffic velocity v given the numbers of vehicles $\boldsymbol{x}$.

3.4.3 Probabilistic Hidden Structure Modeling for Traffic Velocity

The estimation function of the velocity v is found through the posterior distribution $p(v|\boldsymbol{x})$ from Eq. (3.37). The posterior $p(v|\boldsymbol{x})$ for v can be decomposed into an observation model for the number of vehicles $p(\boldsymbol{x}|v)$ that is conditioned on the average traffic velocity v and the prior model for the velocity $p(v)$:

$$p(v|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|v)p(v)}{\int p(\boldsymbol{x}|v)p(v)\mathrm{d}v}. \tag{3.38}$$

In this subsection, we define the observation model $p(\boldsymbol{x}|v)$ conditioned on v and the prior model $p(v)$.

We derive the observation model $p(\boldsymbol{x}|v)$ for $\boldsymbol{x}$ conditioned on the average traffic velocity v by considering the proportion of the number of vehicles that are in consecutive observations. If the time interval between t_n and t_{n+1} is not too large and the length of the observation area L is not too small, some of the vehicles will be the same in consecutive observations. In particular, the proportion of identical vehicles is supposed to become large when the average traffic velocity v is small. Conversely, it becomes small when the velocity is large. Also, the strength of the correlation between the consecutive observations in $\boldsymbol{x}$ will increase with this proportion. We use these relationships between the proportion of identical vehicles, the average traffic velocity v, and the correlation in $\boldsymbol{x}$ to derive the observation model.

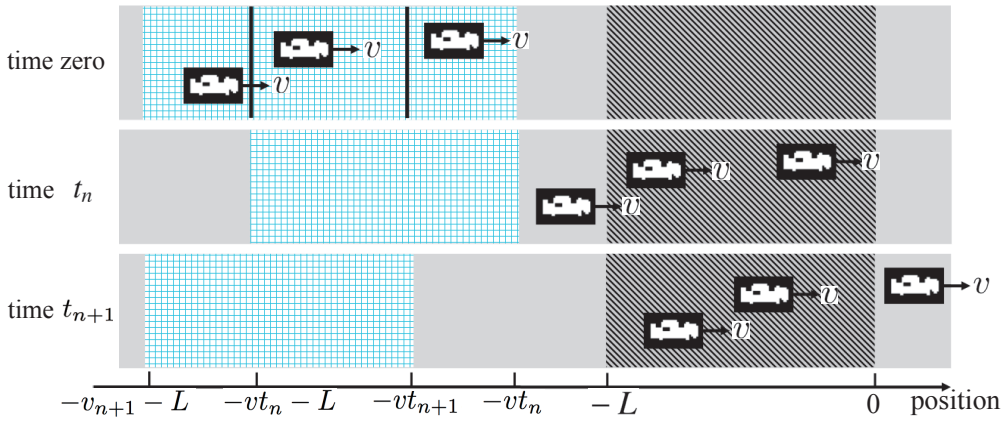


Fig. 3.9 Positions of observed vehicles at time zero, t_n and t_{n+1}. In this case, there are overlapping area between the consecutive n-th and n+1-st observations in $[-vt_n - L, -vt_{n+1})$.

First, we place two assumptions on the observed traffic during the period of N observations, $t_N - t_1$: (i) all the vehicles have a common constant velocity, v, toward the observation area, and (ii) the positions of the vehicles at time zero are independent and identically distributed. These assumptions might be strong but can be applicable for many cases when the period $t_N - t_1$ is not so long. We will examine and discuss these assumptions in experiments and discussions.

Here, we define the positions of the vehicles as the distance from the front of the observation area, which is regarded as the zero position (see Fig. 3.8). The observation area is defined as $[-L, 0)$, as shown in the hatched area in Fig. 3.8, where $[\bullet]$ denotes a closed interval and $(\bullet)$ denotes an open interval. If a vehicle is located at the position $-y$ at time zero, the assumption (i) indicates that the vehicle is observed during time $[(y-L)/v, y/v)$. Accordingly, the vehicles that will be observed at time t_n in the n-th observation should be located in the area $[-vt_n - L, -vt_n)$ at time zero, as shown in the check-pattern area in Fig. 3.8.

Since the area in which the vehicles in the $n+1$-st observation can exist at time zero is $[-vt_{n+1} - L, -vt_{n+1})$, when $-vt_n - L < -vt_{n+1}$, the areas for the consecutive n-th and n+1-st observations partially overlap each other. This overlapping area can be defined as

$$[\max(-vt_{n+1}, -vt_n - L), \min(-vt_{n+1}, -vt_n - L)), \tag{3.39}$$

where vehicles that are in this area at time zero are observed at both times in the n-th and $n+1$-st observations and *are not observed individually in each of the n-th and n+1-st observations*. Note that if there is no overlapping area between the consecutive n-th and $n+1$-st observations, Eq. (3.39) becomes an empty set. We denote the overlapping area as $[-vt_n - L, -vt_{n+1})$ in Fig. 3.9. From Eq. (3.39), the

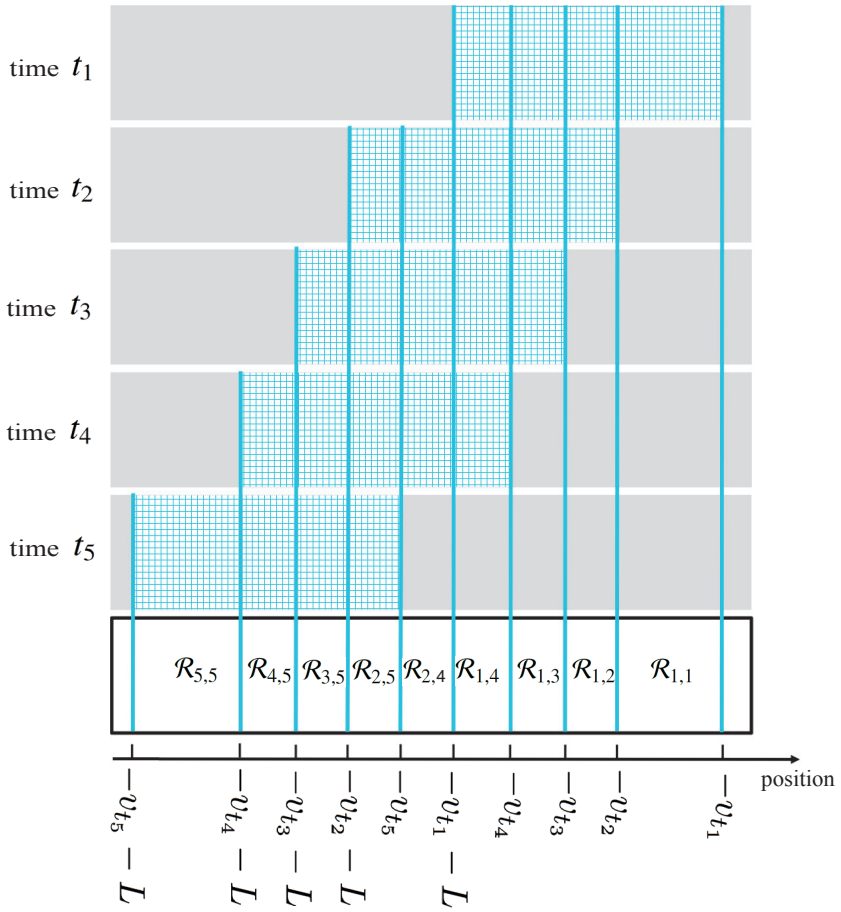


Fig. 3.10 Example of $\mathcal{R}_{j,k}$s and $L_{j,k}(v)$s in the case of $N = 5$. For example, $L_{3,5}(v) = -vt_2 - L - (-vt_3 - L)$. The check-pattern areas represent the areas at which the vehicles that will be observed at time t_n in n-th observation should be located at time zero.

length of the overlapping area can be written by v as

$$\begin{aligned} &|[\max(-vt_{n+1}, -vt_n - L), \min(-vt_{n+1}, -vt_n - L))| \\ &\quad = \max\Big(0, \min(-vt_{n+1}, -vt_n - L) - \max(-vt_{n+1}, -vt_n - L)\Big), \end{aligned} \tag{3.40}$$

where we make the length to be zero for when Eq. (3.39) is the empty set.

While the above overlapping area and its length is defined between the consecutive n-th and n+1-st observations, it can be naturally generalized to the overlapping area between the j-th to k-th consecutive observations in $\boldsymbol{x}$, $\mathcal{R}_{j,k}$ ($j, k \in 1, 2, ..., N,$ and $j \leq k$). The vehicles, which are in this area at time zero, are in all of the j-th to k-th observations and are not observed in the other observations in $\boldsymbol{x}$. This area $\mathcal{R}_{j,k}$ can

be defined as

$$\mathcal{R}_{j,k} \equiv \bigcap_{n=1}^{N} \begin{cases} [-vt_n - L, -vt_n) & (\text{if } j \le n \le k) \\ \overline{[-vt_n - L, -vt_n)} & (\text{elsewhere}) \end{cases}, \tag{3.41}$$

where $\bigcap_{n=1}^{N}$ denotes intersection over $n = 1$ to $n = N$ and $\overline{\bullet}$ denotes exclusion of $\bullet$. Figure 3.10 shows an example of $\mathcal{R}_{j,k}$s in the case of $N = 5$. In Eq. (3.41), since $[-vt_n - L, -vt_n)$ represents the check-pattern area in Fig. 3.10 at each time, $\mathcal{R}_{j,k}$ can be computed as the intersection over the corresponding check-pattern areas:

$$\mathcal{R}_{j,k} = [\max(-vt_{k+1}, -vt_j - L), \min(-vt_k, -vt_{j-1} - L)). \tag{3.42}$$

where $t_0 \equiv -\infty$, $t_{N+1} \equiv \infty$. Note that, from the definition in Eqs. (3.41) and (3.42), some of the intervals $\mathcal{R}_{j,k}$ may also be the empty set, such as $\mathcal{R}_{1,5}$, $\mathcal{R}_{2,2}$, $\mathcal{R}_{2,3}$, $\mathcal{R}_{3,3}$, $\mathcal{R}_{3,4}$, and $\mathcal{R}_{4,4}$ in Fig. 3.10. The area $\mathcal{R}_{n,n}$ represents the area in which the vehicles observed only in the n-th observation exist at time zero. When $N = 2$, Eq. (3.39) and Eq. (3.42) are identical. Additionally, all the ranges $\mathcal{R}_{j,k}$ are mutually exclusive. The length of the overlapping area can be written by v as

$$\begin{aligned} L_{j,k}(v) &\equiv |\mathcal{R}_{j,k}| \\ &= \max\Big(0, \min(-vt_k, -vt_{j-1} - L) - \max(-vt_{k+1}, -vt_j - L)\Big), \end{aligned} \tag{3.43}$$

where we also make the length $L_{j,k}(v)$ to be zero for when Eq. (3.42) is the empty set.

Then, we introduce a random variable $c_{j,k}$ ($j, k \in 1, 2, ..., N$, and $j \le k$), which denotes the number of vehicles in the mutually exclusive area $\mathcal{R}_{j,k}$ at time zero and is a decomposition of $\boldsymbol{x}$. Figure 3.11 shows an example where $c_{2,4} = 1$ and the other components in $\boldsymbol{c}$ are 0 in the case of $N = 5$. The observation variables $\boldsymbol{x}$ can be determined uniquely in terms of $\boldsymbol{c} \equiv [c_{1,1}, ..., c_{1,N}, c_{2,2}, ..., c_{2,N}, ..., c_{N,N}]^\top$:

$$x_n = \sum_{1 \le j \le n \le k \le N} c_{j,k}. \tag{3.44}$$

For simplicity, we introduce an $N \times \frac{1}{2}N(N+1)$ matrix $\boldsymbol{D}$ that corresponds to the above summation and satisfies

$$\boldsymbol{x} = \boldsymbol{D}\boldsymbol{c}. \tag{3.45}$$

Since the position of a vehicle at time zero is random from assumption (ii), the probability, which is that the vehicle appears in the area $\mathcal{R}_{j,k}$ at time zero when the velocity is v, is given by the length of the area $L_{j,k}(v)$ divided by the total length of the roads

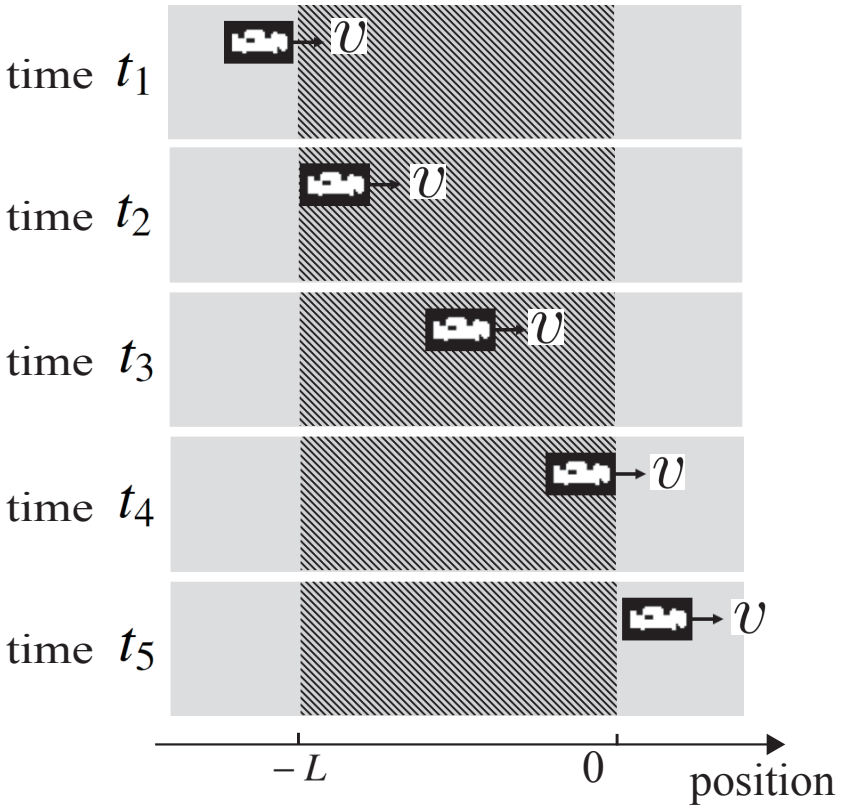


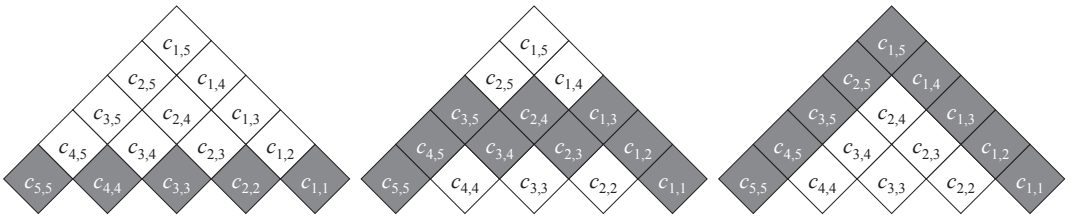
Fig. 3.11 Example in which $c_{2,4} = 1$ and the other components in $\boldsymbol{c}$ are 0 in the case of $N = 5$.

L_{total} in the possible road area, *i.e.*, $L_{j,k}(v)/L_{\text{total}}$. Because all the areas $\mathcal{R}_{j,k}$ are mutually exclusive, the event that this vehicle does not appear among N observations is a complementary event, and its probability is given by $1 - \sum_{1 \le j \le k \le N} L_{j,k}(v)/L_{\text{total}}$. Thus, the random vector $\boldsymbol{c}$ obeys a multinomial distribution using these probabilities and the total number of vehicles M_{total} in the possible road area:

$$
\begin{aligned}
&p(\boldsymbol{c}|v, M_{\text{total}}) \qquad (3.46)\\
&= \frac{M_{\text{total}}}{\left(M_{\text{total}} - \sum_{1 \le j \le k \le N} c_{j,k}\right) \prod_{1 \le j \le k \le N} c_{j,k}} \\
&\quad \times \left(1 - \sum_{1 \le j \le k \le N} \frac{L_{j,k}(v)}{L_{\text{total}}}\right)^{M_{\text{total}} - \sum_{1 \le j \le k \le N} c_{j,k}} \prod_{1 \le j \le k \le N} \left(\frac{L_{j,k}(v)}{L_{\text{total}}}\right)^{c_{j,k}}.
\end{aligned}
$$

Figure 3.12 shows examples of $p(\boldsymbol{c}|v, M_{\text{total}})$, which illustrate typical cases of v. The $c_{j,k}$s depicted as the gray regions have a non-zero probability for each of the high, medium, and low velocity cases. We can see that this distribution $p(\boldsymbol{c}|v, M_{\text{total}})$ for $\boldsymbol{c}$ dramatically changes with the velocity.

Since L_{total} and M_{total} are usually huge, we consider the large limit of them while keeping their ratio constant, *i.e.*, $M_{\text{total}}/L_{\text{total}} = M/L$, where M is a newly introduced



(a) High velocity (b) Medium velocity (c) Low velocity

Fig. 3.12 Examples of $p(\boldsymbol{c}|v, M_{\text{total}})$ where v is low, medium, and high. The gray regions have non-zero probability.

parameter. Both $M_{\text{total}}/L_{\text{total}}$ and M/L mean the vehicle density per unit road length. After taking this limit, $p(\boldsymbol{c})$ becomes

$$p(\boldsymbol{c}|v, M) = \prod_{1 \le j \le k \le N} \text{Poisson}\left(c_{j,k}|q_{j,k}\right), \tag{3.47}$$

$$\text{where} \quad q_{j,k} \equiv \frac{L_{j,k}(v)}{L} M. \tag{3.48}$$

Note that by taking the limit, the $c_{j,k}$s become independent from one another. The v and M are unknown parameters to be estimated. Within our Bayesian framework, we introduce their prior distributions later.

Finally, through marginalization over $\boldsymbol{c}$ using Eqs. (3.45) and (3.47), the observation model for $\boldsymbol{x}$ can be written as

$$p(\boldsymbol{x}|v, M) \equiv \sum_{\boldsymbol{c}} \delta_{\boldsymbol{x}, \boldsymbol{D}\boldsymbol{c}}\, p(\boldsymbol{c}|v, M), \tag{3.49}$$

where δ denotes the Kronecker delta function. However, in this complicated case, the discrete marginalization in Eq. (3.49) is computationally infeasible. As an alternative, $p(\boldsymbol{x}|v, M)$ is approximated as a Gaussian distribution in the following subsection.

We introduce prior distributions for v and M:

$$p(v, M) \equiv \text{InverseGamma}(v|a_v, b_v)\text{InverseGamma}(M|a_M, b_M), \tag{3.50}$$

where the parameters a_v, b_v, a_M, and b_M are treated as input parameters given as part of the model. See the Experimental Results section for these parameters we actually used. The reason we chose inverse gamma distributions for v and M is that they are defined as positive variables and play a role similar to that of the variance parameter in a Gaussian distribution, where the inverse gamma distribution is widely used as the conjugate prior distribution for the variance parameter.

From Eqs. (3.49) and (3.50), we explicitly construct the joint distribution of all random variables:

$$p(\boldsymbol{x}, v, M) \equiv p(\boldsymbol{x}|v, M)p(v, M). \tag{3.51}$$

Algorithm 1 Sampling Procedure for Velocity Estimation

1: Initialize the values of v and M with their prior distributions
2: **repeat**
3: $\xi_v \leftarrow \mathcal{U}(\xi_v|0, p(\boldsymbol{x}, v^{(\tau-1)}, M^{(\tau-1)}))$
4: Sample $v^{(\tau)}$ uniformly from part of the slice $S_v = \{v^{(\tau)}|\xi_v < p(\boldsymbol{x}, v^{(\tau)}, M^{(\tau-1)}))\}$
5: $\xi_M \leftarrow \mathcal{U}(\xi_M|0, p(\boldsymbol{x}, v^{(\tau)}, M^{(\tau-1)}))$
6: Sample $M^{(\tau)}$ uniformly from part of the slice $S_M = \{M^{(\tau)}|\xi_M < p(\boldsymbol{x}, v^{(\tau)}, M^{(\tau)}))\}$
7: **until** a stopping condition is met.
8: **return** $v^{(1)}, v^{(2)}, ..., v^{(T)}$

All marginal and conditional distributions including the posterior $p(v|\boldsymbol{x})$ can be derived in terms of this joint distribution.

3.4.4 Slice Sampling Algorithm for Posterior Mean Estimation

Although the estimation function, the PM $v^*(\boldsymbol{x})$, in Eq. (3.37) is derived from the joint distribution in Eq. (3.51), an exact analytical solution is computationally infeasible. We will thus use an approximate inference method to compute the PM.

Since the discrete marginalization in Eq. (3.49) is computationally infeasible, as stated in the above subsection, and we cannot compute even the joint distribution in Eq. (3.51), $p(\boldsymbol{x}|v, M)$ is approximated as a Gaussian distribution with the same mean and covariance ignoring cumulants higher than second-order:

$$p(\boldsymbol{x}|v, M) \approx \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}), \quad \text{where} \tag{3.52}$$

$$\boldsymbol{\mu} \equiv \mathrm{E}_{p(\boldsymbol{x}|v,M)}[\boldsymbol{x}] \quad \text{and} \quad \boldsymbol{\Sigma} \equiv \mathrm{Var}_{p(\boldsymbol{x}|v,M)}[\boldsymbol{x}]. \tag{3.53}$$

$\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ can be exactly calculated as

$$\boldsymbol{\mu} = \mathrm{E}_{p(\boldsymbol{c}|v,M)}[\boldsymbol{D}\boldsymbol{c}] = \boldsymbol{D}\mathrm{E}_{p(\boldsymbol{c}|v,M)}[\boldsymbol{c}] = M\boldsymbol{i}, \tag{3.54}$$

$$\boldsymbol{\Sigma} = \mathrm{Var}_{p(\boldsymbol{c}|v,M)}[\boldsymbol{D}\boldsymbol{c}] = \boldsymbol{D}\mathrm{Var}_{p(\boldsymbol{c}|v,M)}[\boldsymbol{c}]\boldsymbol{D}^\top = \boldsymbol{D}(\mathrm{diag}\ \boldsymbol{q})\boldsymbol{D}^\top, \tag{3.55}$$

where $\boldsymbol{q} \equiv [q_{1,1}, ..., q_{1,N}, q_{2,2}, ..., q_{2,N}, ..., q_{N,N}]^\top$ and $(\mathrm{diag}\ \boldsymbol{q})$ denotes a diagonal matrix whose diagonal elements are $\boldsymbol{q}$. We will examine this approximation in the Discussion section. Note that $\boldsymbol{D}$ is defined as the operation $\sum_{1\le j\le n\le k\le N}$ for each element of an $\frac{1}{2}N(N+1)$-dimensional vector, as shown in Eq. (3.45). In Eq. (3.54), each element of the vector $\boldsymbol{D}\mathrm{E}_{p(\boldsymbol{c}|v,M)}[\boldsymbol{c}]$ represents the sum of all expectations of the corresponding $c_{j,k}$, which means $\sum_{1\le j\le n\le k\le N} \mathrm{E}_{p(c_{j,k}|v,M)}[c_{j,k}]$ and is always M. Thus, the mean value of the model has no information on the velocity, but the covariance matrix does.

We can hence obtain the likelihood of the traffic velocity from the covariance matrix, which represents the correlation between elements of $\boldsymbol{x}$.

Our goal is to obtain the PM $\hat{v}^*(\boldsymbol{x})$ from the above joint distribution in Eq. (3.51) with the model of Eqs. (3.52) and (3.50) given the observations $\boldsymbol{x}$. However, although we can compute the joint distribution $p(\boldsymbol{x}, v, M)$ thanks to the approximation in Eq. (3.52), we cannot analytically compute the posterior $p(v|\boldsymbol{x})$.

Instead, we derive a sampling-based approximation of the PM $\hat{v}^*(\boldsymbol{x})$ by using the MCMC method. Given the joint distribution $p(\boldsymbol{x}, v, M)$ and observations $\boldsymbol{x}$, we can take T samples for v, $\{v^{(\tau)}\}_{\tau=1}^{T}$, from the posterior $p(v|\boldsymbol{x})$ with the MCMC method, but without explicitly computing the posterior $p(v|\boldsymbol{x})$. Then, we use the empirical mean of $\{v^{(\tau)}\}_{\tau=1}^{T}$ as an approximation of $\hat{v}^*(\boldsymbol{x})$:

$$\hat{v}^*(\boldsymbol{x}) = \int v\ p(v|\boldsymbol{x})\mathrm{d}v \approx \frac{1}{T}\sum_{\tau=1}^{T} v^{(\tau)}. \tag{3.56}$$

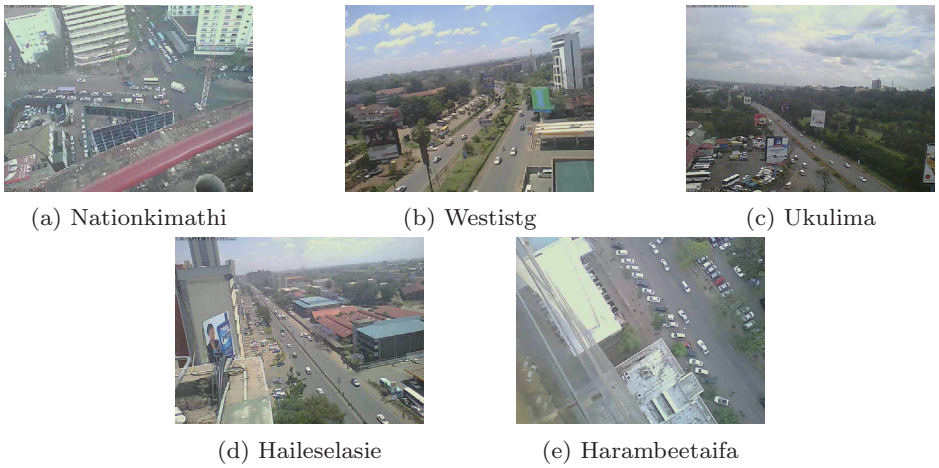
For sampling $\{v^{(\tau)}\}_{\tau=1}^{T}$ in Eq. (3.56), we use slice sampling [50] introduced in Chapter 2. Since we have two different random variables v and M, we repeatedly sample v and M in turn in the same manner as Gibbs sampling [13, 50, 51] and obtain samples only for v from $p(v|\boldsymbol{x})$ by ignoring samples for M. For the stopping condition, we simply use the number of iterations. See the Experimental Results section for the iteration number we actually used.

Algorithm 1 shows the sampling procedure for the velocity estimation task. Here, ξ_v and ξ_M are auxiliary variables for v and M, respectively, which are required in the sampling scheme of slice sampling, and $\bullet \leftarrow \circ$ denotes that a sample from a distribution $\circ$ is substituted into $\bullet$. For efficiency, we use the "stepping out procedure" in Steps 3 and 5 and use the "shrinkage procedure" in Steps 4 and 6, as described in [50].

3.5 Experimental Results

3.5.1 Experimental Results for Traffic Volume Estimation using Real-world Web-camera Images

We tested our proposed method for traffic volume estimation using real-world web-camera images captured in Nairobi, Kenya [35, 36, 75, 77, 78], as shown in Figure 3.13. These images were captured from roads at five different locations with the same size of 640×480 pixels. As shown in Figure 3.1, on average only several hundred of these pixels are occupied by individual vehicles. Also, the frame-rate is one image per six seconds. These are far too poor for the assumptions of the existing vehivle recognition approaches [120–122]. The number of images was $N = 100$ for each location.



(a) Nationkimathi (b) Westistg (c) Ukulima

(d) Haileselasie (e) Harambeetaifa

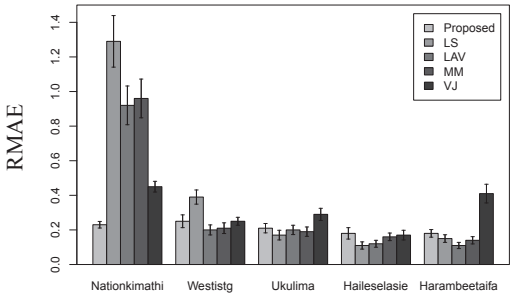
Fig. 3.13 Traffic monitoring web-camera images [75].

Also, we used the hyperparameter values of $a_\beta^{(0)} = a_{\rho_0}^{(0)} = a_{\rho_1}^{(0)} = a_\alpha^{(0)} = 1$, $b_\beta^{(0)} = b_{\rho_0}^{(0)} = b_{\rho_1}^{(0)} = b_\alpha^{(0)} = 10^{-10}$, and $\mu_{\theta_0}^{(0)} = -1$ and $\mu_{\theta_1}^{(0)} = 0.3$. We chose them to be as *non-informative* as possible in a fully Bayesian framework and to have a quite flat distribution. Also, preliminary experiments showed the accuracy of the algorithm is insensitive to changes in the values of the hyperparameters.

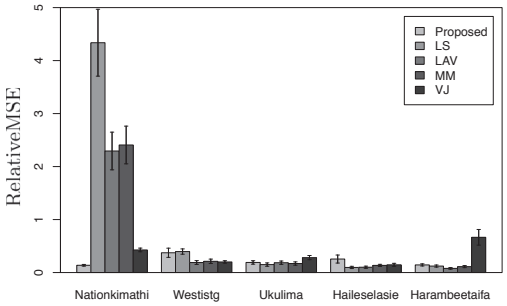
Figure 3.14 compares our unsupervised approach with several *supervised* alternatives. To train those, we used the true count labels in addition to the VPA, and hence the comparison is extremely preferable to the alternatives. We used least squares linear regression (LS), least absolute values (LAV), and MM estimator (MM). See [123] for details of the algorithms. We also compared our unsupervised approach with a widely used vehicle recognition approach by Viola and Jones (VJ) [121] as another baseline method using features other than from VPA.

Notice that these supervised alternatives, LS, LAV, and MM, *require labeled training data customized for each camera location*, which is in fact impractical in city-wide traffic monitoring scenarios. We gave these methods 100 *labeled* data *for each location*. In the VJ training, we prepared 2000 *labeled* images for positive and negative examples. They consisted of popular image databases that include vehicles [122, 124–126] and several hundred manually labeled images that came from our training data set. For the training of the supervised alternatives, manual vehicle-counting and labeling were used to create the labeled data, and they took several days to complete. In contrast, the computational time for our VB inference took only a few seconds on a moderately capable laptop computer and the time complexity is $O(N)$.

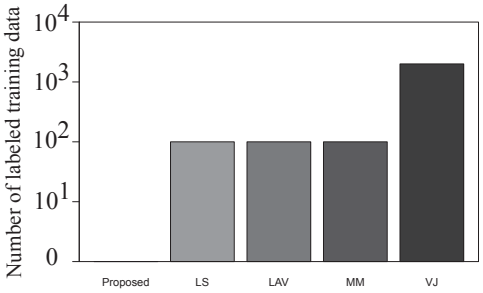
The goal of this experiment was to see if our *unsupervised* method is comparable in performance to these *supervised* alternatives. For each location, we evaluated the results with regard to the relative mean absolute error (RMAE) and the relative



(a) RMAEs for all of the camera locations (smaller is better). Error bars represent the standard error.

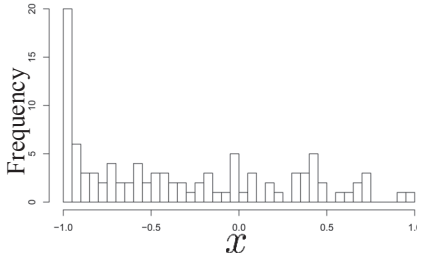


(b) RelativeMSEs for all of the camera locations (smaller is better). Error bars represent the standard error.

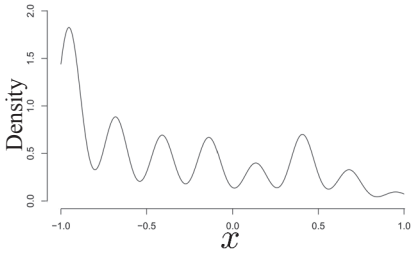


(c) Number of labeled training data (smaller is better). Note that the proposed method requires no labeled training data.

Fig. 3.14 Comparison of the proposed unsupervised method and supervised alternatives for all of the camera locations



(a) Observed histogram of x



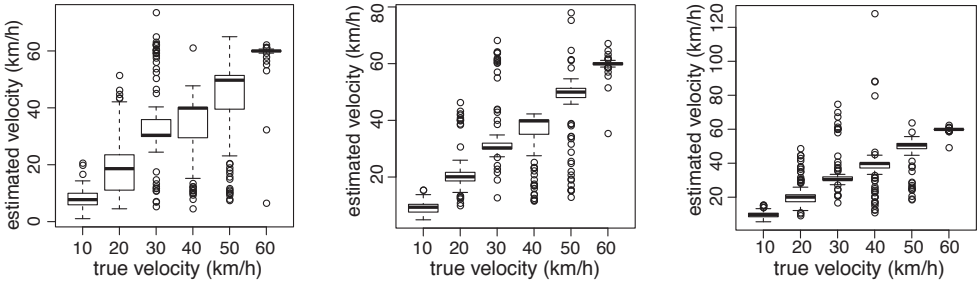
(b) Estimated predictive posterior of x

Fig. 3.15 Observed histogram and estimated predictive posterior of $\boldsymbol{x}$.

mean square error (RelativeMSE) over $M = 100$ images. RMAE and RelativeMAE are defined as

$$\mathrm{RMAE} = \frac{1}{M}\sum_{m=1}^{M}\frac{\left|d_{\mathrm{true}}^{(m)} - d_{\mathrm{estimate}}^{(m)}\right|}{d_{\mathrm{true}}^{(m)} + 1} \quad \text{and} \tag{3.57}$$

$$\mathrm{RelativeMSE} = \frac{1}{M}\sum_{m=1}^{M}\frac{\left|d_{\mathrm{true}}^{(m)} - d_{\mathrm{estimate}}^{(m)}\right|^2}{(d_{\mathrm{true}}^{(m)})^2 + 1}, \tag{3.58}$$



(a) Average number of vehicles $M_{\text{art}} = 1$

(b) Average number of vehicles $M_{\text{art}} = 10$

(c) Average number of vehicles $M_{\text{art}} = 100$

Fig. 3.16 True velocity and velocity estimated using the proposed method for the time interval $\Delta t = 1$ (second) in the artificial instance. Note that we have used the Tukey boxplot [127].

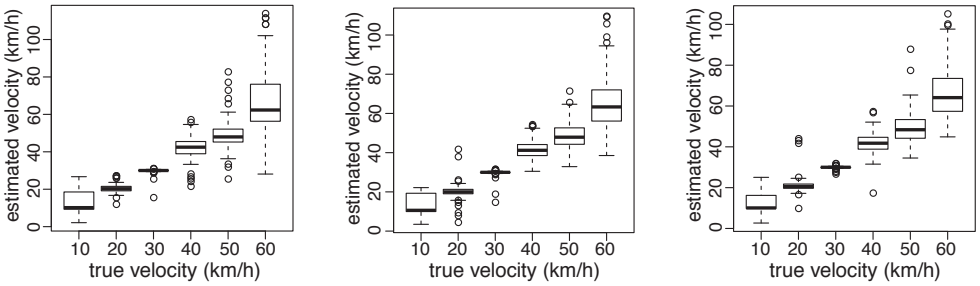
where $d_{\text{true}}^{(m)}$ is the true number of vehicles in the m-th image, and $d_{\text{estimate}}^{(m)}$ is the estimated number of vehicles for the m-th image. We computed the standard error of the relative absolute error and relative square error (the error bars in Fig. 3.14).

From Fig. 3.14, we can see that the overall performance of our method is comparable to or even better than those of the supervised alternatives. This is rather surprising, because our method does *not* use any labeled training data. Our method gives quite stable RMAE scores for the various camera locations in contrast to most of the supervised alternatives, which have significantly worse scores at the Nationkimathi location due to outliers and occlusions. These results demonstrate the robustness of our approach against the image conditions.

Finally, for a reality check of the VB inference, Fig. 3.15 compares the predictive posterior distribution of x, $p(x|\boldsymbol{x})$, with the histogram created from the data. To get $p(x|\boldsymbol{x})$, we marginalized all of the parameters except for x using the variational posterior q. The result confirms that the estimated predictive posterior is consistent with the true observed histogram. The predictive posterior $p(x|\boldsymbol{x})$ resembles the distribution in Figure 3.3. It has equally-spaced clusters having variances depending on the values of d.

3.5.2 Experimental Results for Traffic Velocity Estimation on Artificial Traffic

We examined the proposed approach for traffic velocity estimation in numerical experiments. First, we generated artificial datasets to study the performance of the approximate inference method in this subsection and to see if it could deal with actual traffic, we applied it to the temporal-sequences of the numbers of vehicles extracted from real-world web-camera images and to publicly available traffic datasets in the



(a) Average number of vehicles $M_{\text{art}} = 1$

(b) Average number of vehicles $M_{\text{art}} = 10$

(c) Average number of vehicles $M_{\text{art}} = 100$

Fig. 3.17 True velocity and estimated velocity using the proposed method for the time interval $\Delta t = 4$ (second) in the artificial instance. Note that we have used the Tukey boxplot [127].

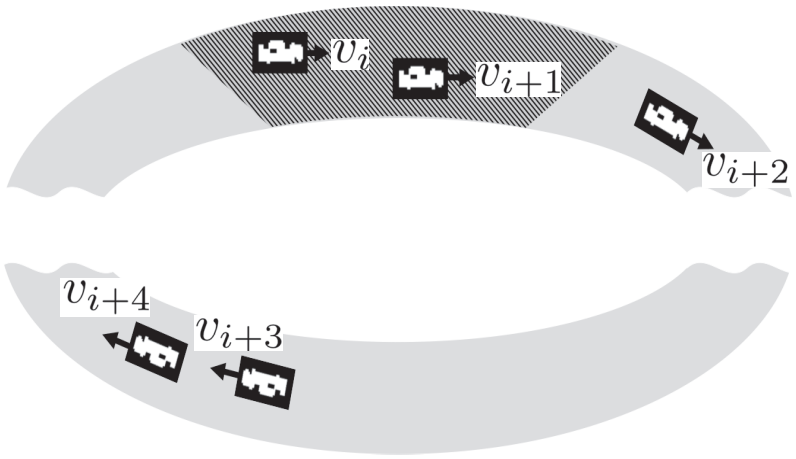


Fig. 3.18 Traffic simulation for validation.

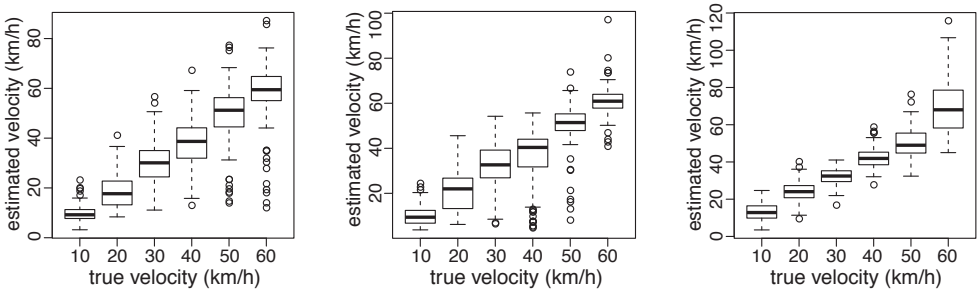
next subsection.

For the Bayesian inference, we set the hyperparameter values in Eq. (3.50) to be as non-informative as possible and to have a quite flat distribution:

$$a_v = a_M = 10^{-4},\ b_v = v_{\text{legal}} \times 10^{-4},\ b_M = \mu_{\boldsymbol{x}} \times 10^{-4}, \tag{3.59}$$

$$\text{where} \quad \mu_{\boldsymbol{x}} \equiv \frac{1}{N} \sum_{n=1}^{N} x_n. \tag{3.60}$$

For the prior means of v and M, we respectively used the legal speed limit v_{legal} on each road and the naively computed a sample mean of $\boldsymbol{x}$. The number of effective prior observations of the inverse gamma distribution within the Bayesian framework is equal to twice the value of parameter a. These settings were considered sufficiently non-informative. For fairness, we used this hyperparameter setting for all of the following instances. Also, the number of iterations of the slice sampling was 1000.



(a) Time interval $\Delta t = 1$ (second)

(b) Time interval $\Delta t = 2$ (second)

(c) Time interval $\Delta t = 4$ (second)

Fig. 3.19 True average velocity and velocity estimated using the proposed method in the simulation instance. Note that we have used the Tukey boxplot [127].

A preliminary analysis indicated that using more iterations, such as 100000, did not improve the accuracy much.

In preparing the artificial validation dataset, we randomly generated $\boldsymbol{c}$ at constant time intervals from the model in Eq. (3.47) and computed the observations $\boldsymbol{x}$ as the sum of the corresponding $\boldsymbol{c}$. Using the model in Eq. (3.47) for generating the dataset and using the approximate model in Eq. (3.52) for estimating the velocity, we studied the validity of the Gaussian approximation for the approximate model in Eq. (3.52) and the performance of the approximate PM inference.

Using Eq. (3.47), we generated $\boldsymbol{x}$ with the following settings: time intervals of $\boldsymbol{t}$, $\Delta t \in \{1, 4\}$ (second), length of the observation area $L = 100$ (m), average number of vehicles on the road $M_{\text{art}} \in \{1, 10, 100\}$, traffic velocity $v \in \{10, 20, 30, 40, 50, 60\}$ (km/h), and number of input observations $N = 50$. We set the legal speed limit to the one in Japan, $v_{\text{legal}} = 60$ (km/h). For each of these settings, we repeatedly evaluated the proposed method in 100 experiments, where we used a different random seed in each experiment. Thus, the total number of experiments was $2 \times 3 \times 6 \times 100 = 3600$.

Figures 3.16 and 3.17 compare the true velocity with the estimated velocity using the proposed method. We can see that the overall performance of our method is good. Our method performed consistently for different values of M, even $M = 1$, which is a difficult setting for this kind of Gaussian approximation (Eq. (3.52)). The results show that the Gaussian approximation and approximate MCMC inference method worked well. They are also non-trivial because the density-velocity regression approach cannot work in a scenario where the average number of vehicles is independent from the traffic velocity.

Next, we evaluated the proposed method on simulated traffic data, where the observations $\boldsymbol{x}$ were obtained from our simple traffic simulation. We examined its ro-

Fig. 3.20 Real-world web-camera image used in our experiment.

bustness in a situation where each vehicle had a different velocity.

We simulated the traffic as follows (see Fig. 3.18). First, we distributed M_{sim} vehicles at random positions on a virtual circuit whose total length was L_{sim}. The parameters M_{sim} and L_{sim} were respectively equal to one-thousandth of the total number of vehicles in Japan and the total road length in Japan. The vehicles moved at different velocities that were generated from a uniform distribution $\mathcal{U}(x|\tilde{v}-10, \tilde{v}+10)$, where the average of the true velocities was set as $\tilde{v} \in \{10, 20, 30, 40, 50, 60\}$ (km/h). This means the vehicles had their own velocities. We repeatedly obtained the numbers of vehicles $\boldsymbol{x}$ from a certain road in the virtual circuit at constant time intervals. We experimented with time intervals of $\boldsymbol{t}$ $\Delta t \in \{1, 2, 4\}$ (second), a length of the observation area $L = 100$ (m), and $N = 50$ input observations. We set the legal speed limit in this experiment to the legal speed limit in Japan, *i.e.*, $v_{\mathrm{legal}} = 60$ (km/h). For each of these settings, we evaluated the proposed method in 100 experiments, where we used a different random seed in each experiment. Thus, the total number of experiments was $3 \times 6 \times 100 = 1800$.

Figure 3.19 compares the true average velocity with the estimated velocity using the proposed method. Even when each vehicle had a different velocity, we can see that the proposed method could estimate the average traffic velocity.

3.5.3 Experimental Results for Traffic Velocity Estimation using Real-world Dataset

We demonstrated the utility of our approach by using it to estimate the traffic velocity from web-camera images captured in Tokyo, Japan for the city traffic monitoring scenario [36, 77, 78] (see Fig. 3.20). The specific location is at 35.651054N, 139.799986E. The image size was 640×480 pixels. The legal speed limit on this road was $v_{\mathrm{legal}} = 60$ (km/h). The frame rate of the web camera was about one image

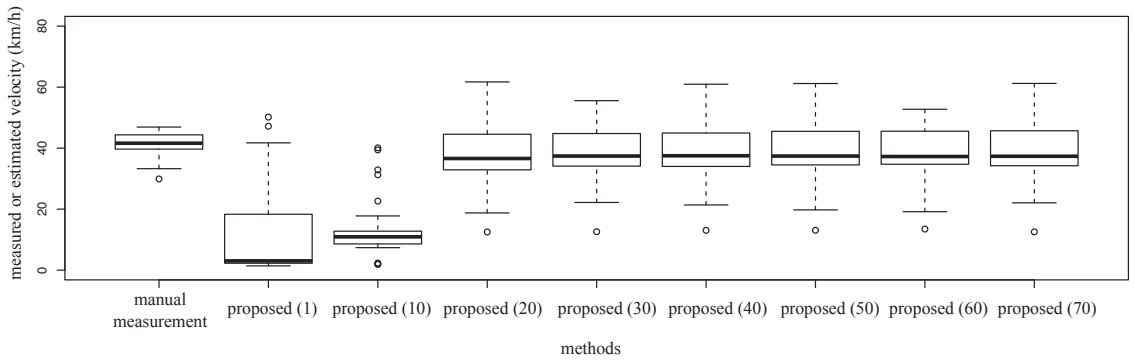
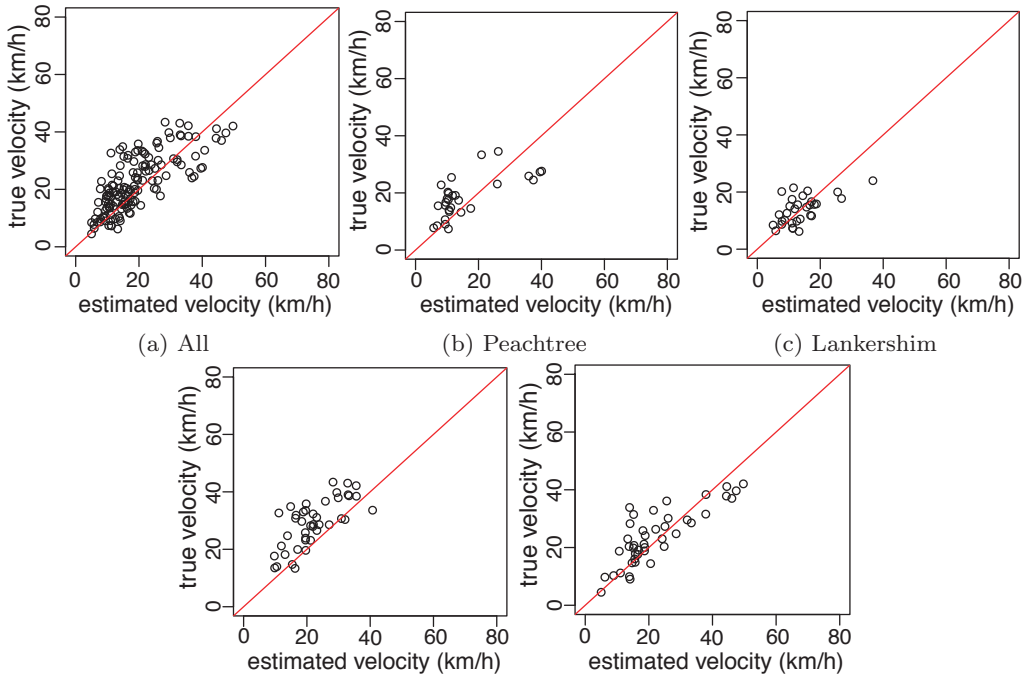


Fig. 3.21 Manually measured velocity and velocity estimated using the proposed method with different prior means. In this figure, proposed (1) means the proposed method with the prior mean 1 (km/h) for the traffic velocity. Note that we have used the Tukey boxplot [127].

per second. Since the sampling rate of the web camera was low, this was a good application for our approach. The dataset contained images captured for 17 minutes, and the images in the set totaled about 1000.

For $\boldsymbol{x}$, we simply used the temporal-sequences of the numbers of vehicles in the images, which were extracted from the images by using the method described in [35, 36, 77, 78]. This method works for any frame rate, even for still images, is robust even when image quality is poor and is almost calibration-free because it does not recognize individual vehicles, but rather estimates their number from the image features. The road length L can easily be obtained because we can estimate the size of the vehicles and the length of the road in the images by using the methods described in [35, 36, 77, 78]. In particular, L was computed by referring to the typical sizes of vehicles in the real world. We used the timestamps attached to the images as $\boldsymbol{t}$ and input $N = 60$ consecutive images (corresponding to about one minute) for each estimation. We estimated the traffic velocities about 17 times. To create the validation test dataset, we manually measured traffic velocities each minute by using a radar speed gun at the roadside. The average traffic velocity was 41 (km/h).

Since we have only one traffic situation in this real-world traffic scenario, in which the average traffic velocity was 41 (km/h), we checked to ensure that the hyperparameter setting for the traffic velocity does not accidentally become the best for estimating this velocity. We tested the following hyperparameter settings: $b_v \in \{1, 10, 20, 30, 40, 50, 60, 70\} \times 10^{-4}$ and $a_v = 10^{-4}$ for the traffic velocity v. This means that the prior mean b_v/a_v for v takes over $\{1, 10, 20, 30, 40, 50, 60, 70\}$ (km/h). In our Bayesian framework, the traffic velocity estimation is generally difficult when this prior mean is significantly different from the true velocity.



(a) All (b) Peachtree (c) Lankershim

(d) US-101 (e) I-80

Fig. 3.22 Relationship between true NGSIM velocity and velocity estimated by the proposed method. The estimate is good if the dot is close to the identity line (red line).

Figure 3.21 compares our estimation results of each prior mean with the manually measured velocities in the 17 minutes of observations. We can see that our method can estimate a reasonable velocity over the different prior mean settings of v, except for the settings in which the prior means are 1 and 10 (km/h). In these cases (in which the prior means are 1 and 10 (km/h)), the true velocity is more than four times larger than the prior mean. Since it is not difficult to set the prior mean in a range that is the true velocity plus or minus 30 (km/h), this result shows that the dependence of the hyperparameters is small enough.

Finally, we evaluated the proposed method using the publicly available Next Generation Simulation (NGSIM) datasets collected by the United States Department of Transportation Federal Highway Administration [128]. The NGSIM datasets consist of real-world vehicle trajectory data collected using digital video cameras at several locations in the United States. *The proposed method only used the number of vehicles every second in the NGSIM datasets; it did not use the original speed information.* The original speed information was used only for evaluating the estimation accuracy of the proposed method.

We used four vehicle trajectory datasets: Peachtree, Lankershim, US-101, and I-80. Each of them was collected at different locations. The Peachtree and Lankershim

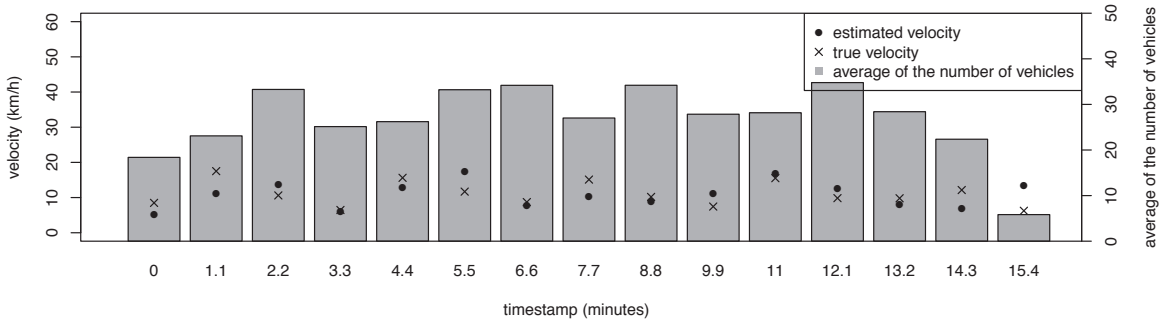


Fig. 3.23 Time-series of velocities estimated by the proposed method and the corresponding true velocities. The average number of vehicles for the corresponding timestamp is indicated by the bar chart.

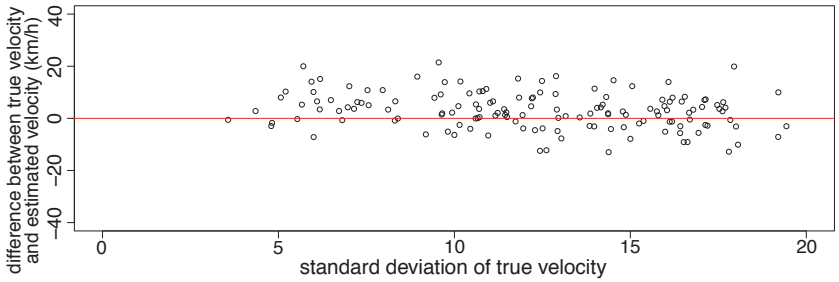


Fig. 3.24 Relationship between the estimation error and standard deviation of velocity. The estimate is good if the dot is close to the line, which means the difference is zero (red line).

datasets were collected on local roads, which means that vehicles had relatively lower velocities and the legal speed limit was $v_{\text{legal}} = 56.3269$ (km/h). The US-101 and I-80 datasets were collected on freeways, which means that vehicles had relatively higher velocities and the legal speed limit was $v_{\text{legal}} = 88.5137$ (km/h).

We used the initial 100 (m) areas without intersections or branches at each location as the observation area, which means the length of the observation area was $L = 100$ (m). We used the numbers of vehicles in the trajectories on the observation area at each time and *thinned them to one observation per second for making the observations* $\boldsymbol{x}$ *in our problem setting*. We input $N = 60$ (corresponding to about one minute) consecutive observations for each estimation. Since the Peachtree and Lankershim datasets contain vehicle trajectories for about 30 minutes, the number of estimations for each of them was about 30. Similarly, since the US-101 and I-80 datasets contain vehicle trajectories for 45 minutes, the number of estimations for each of them was about 45. Thus, the total number of estimations for these datasets was about 150. We used the timestamps attached to the trajectories as $\boldsymbol{t}$.

We compared the estimated velocities with the true ones for each location, as shown in Fig. 3.22. We can see that the proposed method can estimate a reasonable velocity at every location with the different true velocity. For the US-101 dataset, the estimation results seem to have bias. We will discuss the bias in the Discussion section. Additionally, Fig. 3.23 shows typical time-series of the true and estimated velocities, together with average number of vehicles. We can see that the proposed method can estimate a reasonable velocity regardless of the average number of vehicles. In addition, Fig. 3.24 shows the relationship between the estimation error and standard deviation of vehicle velocity. Although we assume that all vehicles have a common velocity, the standard deviation of the velocity, which reflects variation of the velocities over vehicles and time, including situations such as overtaking and the existence of multiple lanes, did not affect the quality of the estimation of the real-world NGSIM data, as well as, the simulated traffic.

3.6 Discussion

We discuss the proposed unsupervised approach in this section. First, we discuss the robustness in choosing features in our traffic-volume-estimation method. Second, we give the details of image-binarization method for extracting the image feature VPA. Third, we discuss the traffic-volume-estimation model with Gaussian mixture and SBP. We then discuss the validity of our velocity-estimation approach. Next, we discuss the validation of the limitations of this approach. Finally, we discuss other applications of the approach.

3.6.1 Robustness against Choice of Features for Traffic Volume Estimation

One of the advantage of our approach is the robustness against the choice of features. In the traffic-volume estimation, we may use other scalar features, such as the number of edges or corners in the image, as long as those are believed to be linearly correlated with the vehicle count.

Figure 3.25 shows the relationship between two features, where V1 is the number of vehicles, V2 is the VPA, V3 is the number of edges by first-order Sobel filter, V4 is the number of edges by second-order Sobel filter, V5 is the number of edges by Laplacian filter, and V6 is the number of corners by Harris corner detector. These methods are explained in [129]. From the figure, all of these features are believed to be linearly correlated with the vehicle count, and we can use them as the input of our method. However, we also confirmed that the other features do not provide significant improvements as compared to the VPA. Also, we use the VPA for its simplicity and

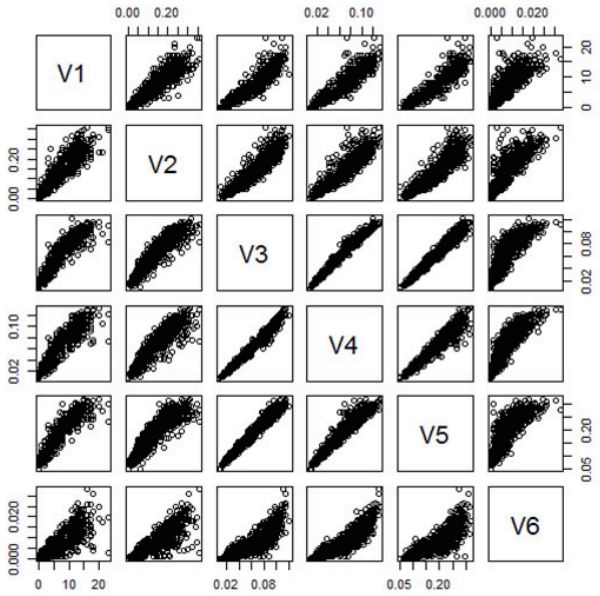


Fig. 3.25 Pairwise scatterplots for pairs of the features

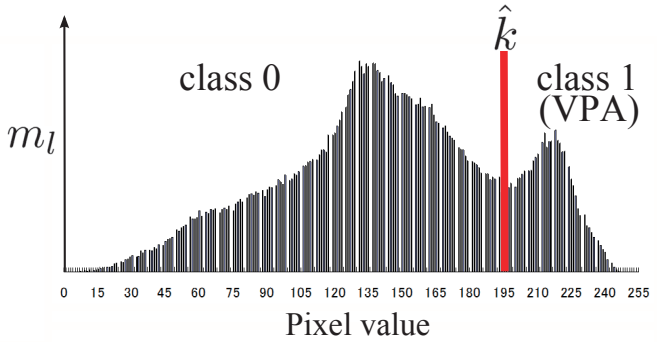


Fig. 3.26 Otsu's binarization method

clarity. Unlike other image-based features, such as the number of edges, VPA has no ambiguity in the definition.

3.6.2 Details of Image Binarization Method

We show the concrete procedure of Otsu's binarization method [113] used for extracting the VPA from images.

We partition the pixels of the focus area into two classes based on Otsu's binarization method. Each of the pixels in the focus area is represented 256 gray-levels [0, 1, ..., 255], and Otsu's method binarizes this image with a single threshold k, which takes a natural number within the range of 256 gray-levels $[1, 254]$. We define

the darker pixels by the class label 0, and the brighter ones by 1. Our goal is to find an optimal threshold $\hat{k}$ so that the brighter class (class 1) correspond to vehicles. Let $\omega_0(k)$ and $\omega_1(k)$ be relative weights for the class 0 and 1, respectively:

$$\omega_0(k) \equiv \frac{\sum_{l=0}^{k} m_l}{\sum_{l=0}^{255} m_l}, \text{ and } \omega_1(k) \equiv \frac{\sum_{l=k+1}^{255} m_l}{\sum_{l=0}^{255} m_l}, \tag{3.61}$$

where $\omega_0(k) + \omega_1(k) = 1$, and m_l denotes the number of pixels at level l. Also, we define that $\mu_0(k)$ and $\mu_1(k)$ as the mean values of pixel values for the class 0 and 1, respectively:

$$\mu_0(k) \equiv \frac{\sum_{l=0}^{k} m_l l}{\sum_{l=0}^{k} m_l}, \text{ and } \mu_1(k) \equiv \frac{\sum_{l=k+1}^{255} m_l l}{\sum_{l=k+1}^{255} m_l}, \tag{3.62}$$

Then, in the spirit of the discriminant analysis [130], the optimal threshold $\hat{k}$ is chosen so that a between-class variance $\sigma_B^2(k)$ is maximized as

$$\hat{k} \equiv \underset{k}{\text{argmax}} \ \sigma_B^2(k), \text{ where} \tag{3.63}$$

$$\sigma_B^2(k) \equiv \omega_0(k)\big(\mu_0(k) - \mu_T\big)^2 + \omega_1(k)\big(\mu_1(k) - \mu_T\big)^2, \tag{3.64}$$

and μ_T represents the grand mean of the pixel value defined as

$$\mu_T \equiv \frac{\sum_{l=0}^{255} m_l l}{\sum_{l=0}^{255} m_l}. \tag{3.65}$$

The number of pixels assigned to class 1 is a raw score for the image feature. This maximization of $\sigma_B^2(k)$ is equivalent to a minimization of a intra-class variance, which is a weighted sum of variances of the pixels in the class 0 and 1 [113]. The optimal threshold $\hat{k}$ is determined in a reasonable manner, which is the mean values $\mu_0(k)$ and $\mu_1(k)$ for the class 0 and 1 differ from the grand mean μ_T as much as possible, or the variances of the pixels in the class 0 and 1 are as low as possible. Figure 3.26 shows the example of thresholding by the method for 256 gray-level histogram. This method was developed many years ago, but still used as a state-of-the-art image-binarization technique [114, 115].

3.6.3 Gaussian Mixture as a Counting Model and Bayesian Nonparametrics

From Eq. (3.4), the GMM formulation without any labeled training data does not give a unique solution. The likelihood of the count $\boldsymbol{h}$ in Eq. (3.4) is invariant with respect to the simultaneous translation of x and θ_0, as well as the simultaneous scaling between d and θ_1. This means that the counting results of the proposed GMM without any additional constraint will become *linearly proportional to the true count*.

To remove this indistinguishability, we essentially used a minimum assumption for the count; that is, the assigned count values for the observations are consecutive natural numbers from zero, which is realistic for most counting problems. This means that we choose the smallest (simplest) one from possible count sets in the training data set $\boldsymbol{X}$. For example, when we have hundreds of observations and the possibilities for the corresponding count sets $\{0, 1, 2, \ldots, 99\}$, $\{100, 101, 102, \ldots, 199\}$, and $\{0, 10, 20, \ldots, 990\}$ are equivalent, we choose the smallest one $\{0, 1, 2, \ldots, 99\}$. This rather ad hoc introduced constraint for the count $\boldsymbol{h}$ was mathematically represented using the SBP prior [112] commonly used in nonparametric Bayes models [112] as the prior for the count $\boldsymbol{h}$ in our generative model. From Eq. (3.7), the probability of the counts decreases in ascending order of the count on average, and this can solve the above issues of the proposed GMM.

In traditional Bayesian nonparametric literature, this nature is known not only as a useful tool to automatically determine the number of mixture components, but also as a drawback; that is, it can cause the solution to get stuck at a local minimum in practical use [131, 132]. This is because the biased ordering of the expected components' probabilities means that a permutation of the component indexes changes the probability distribution, and each component is always associated with the same index. Interestingly, this drawback becomes a natural constraint for the count in the proposed model.

3.6.4 Validity of Velocity Estimation Method from Temporal-sequences of Vehicle Counts

From our experiments for the traffic velocity estimation, without tracking any vehicles or using any labeled training data, we confirmed that the proposed observation model for the temporal-sequences of the numbers of vehicles $\boldsymbol{x}$ can properly represent the likelihood of v given $\boldsymbol{x}$. Also, we showed that our approximate estimation method performs consistently and stably well on the simulation and real-world datasets.

We used slice sampling in the implementation of our method. Gibbs sampling, which is one of the most common MCMC methods, is not applicable to our model because we cannot analytically compute the conditional distributions required in its sampling procedure. Also, an efficient implementation of the Metropolis algorithm, which is also a common MCMC methods, is difficult because it is almost impossible to prepare appropriate proposal distributions for both of the random variables v and M. We prefer slice sampling because it does not require such analytical modeling or sensitive setting of the proposal distributions [50, 51].

With regard to the initial sample value for the traffic velocity in our slice sampling

procedure, we used the prior mean of the traffic velocity. Here, we recommend that the initial sample value for the traffic velocity be set sufficiently high regardless of the prior mean value. The legal speed limit is good enough for the recommended setting. To make sure, by using the real-world dataset in Section 3.5.3, we examined whether the proposed method worked well when we set the initial sample value to the legal speed limit regardless of the prior mean value. We tested it in settings in which the prior means were 1 and 10 (km/h) and the initial sample value was always the legal speed limit, 60 (km/h). Although we failed to estimate the correct traffic velocity by using the prior mean as the initial sample value in these settings for the prior mean in the experiment in Section 3.5.3, by using the legal speed limit as the initial sample value, we could estimate the correct traffic velocity 41 (km/h), as shown in Fig. 3.27.

With regard to the number of observations N for $\boldsymbol{x}$, we used $N = 50$ for the artificial and simulation instances and $N = 60$ for the real-world instance. Roughly speaking, the proposed method required more than $N = 10$ observations to get a good estimation. A preliminary analysis indicated that the accuracy did not improve much when more observations N, such as $N = 100$, were used. On the other hand, when we used fewer observations, such as $N = 5$, the proposed method could not estimate an appropriate traffic velocity.

With regard to the calculation cost, the proposed algorithm requires $\mathcal{O}(N^3)$. This cost is determined by the matrix inversion: $\boldsymbol{\Sigma}$ in Eq. (3.52). Also, since we use the MCMC-based approach, slice sampling, the proposed algorithm requires a large number of iterations for the inference. The use of deterministic algorithms, such as VB [52], seems to be a promising way of making a more efficient inference.

3.6.5 Validation of Limitations of Traffic Velocity Estimation

Here, we discuss the limitations of the proposed model, which is caused by our assumptions on the time interval (sampling rate) between observations, the quality of the observation, and the positions of the vehicles at time zero.

When the time interval between observations is quite large, since all of the $L_{j,k}$s for $j, k \in j < k$ are zero in Eq. (3.43), the likelihood of v given $\boldsymbol{x}$ (Eq. (3.52)) always takes same value for any v. In this situation, because the posterior distribution becomes equivalent to the prior distribution, the estimated traffic velocity converges to the prior mean and we cannot estimate the reasonable traffic velocity. From the definition shown in Eq. (3.43), the condition under which we can estimate the traffic velocity is

$$\max_{j,k \in j<k} L_{j,k}(v) > 0. \tag{3.66}$$

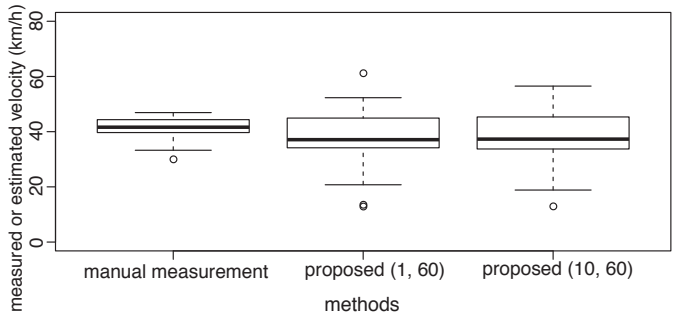


Fig. 3.27 Manually measured velocity and velocity estimated using the proposed method with different prior means. The initial sample value was always the legal speed limit, 60 (km/h). In this figure, proposed (1, 60) means the proposed method with the prior mean 1 (km/h) and the initial sample value 60 (km/h) for the traffic velocity. Note that we have used the Tukey boxplot [127].

From Eq.(3.66), we can also derive an upper bound of the traffic velocity that the proposed method can deal with:

$$v < \frac{L}{\min_{j\in\{1,2,\ldots,N-1\}}(t_{j+1} - t_j)}. \tag{3.67}$$

The upper bound is determined by the time interval of $\boldsymbol{t}$ and the road length L. We have seen that the proposed model can estimate a reasonable traffic velocity in most practical settings, including the real-world case study in Section 3.5.3, where one can obtain, e.g., one image per second and the length of the observed area is $L = 24$ (m). In this subsection, we examine the limitation of the proposed model in terms of the time interval by using the artificial validation dataset in Section 3.5.2 with an extreme setting, that is, the time interval $\Delta t = 10$ (second) and road length $L = 100$ (m). Figure 3.28 compares the true velocity with the estimated velocity. We can see that the estimated velocity converges to the prior mean, 60 (km/h), from around the upper bound velocity, 36 (km/h), in this extreme setting.

With regard to the quality of the observation variable, that is, the vehicle count $\boldsymbol{x}$, we modeled it probabilistically by taking into consideration statistical noise, as shown in Eqs. (3.46) - (3.49). Here, let us examine the influence of the counting error. Using the NGSIM data in Section 3.5.3, we tested the proposed method when some vehicles were constantly missing or double-counted through all the observations; this situation corresponds to one in which we cannot find some vehicles because they blend in with the background or one in which some vehicles, such as busses or trucks, are double-counted because they are larger than ordinary vehicles. Figure 3.29 shows the mean absolute error for the settings in which vehicles are (a) missing (negative noise) in a particular proportion, (b) double-counted (positive noise) in a particular proportion, and (c) missing or double-counted (mixed noise) in a particular proportion. They

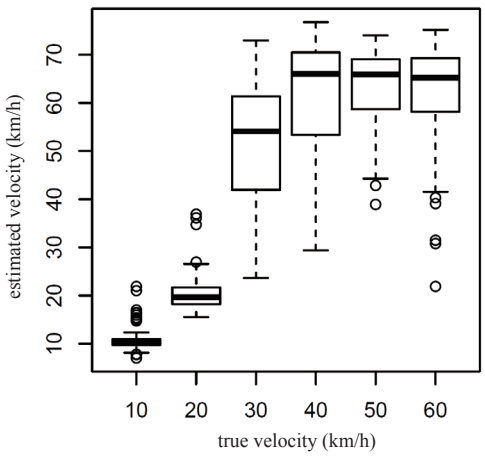
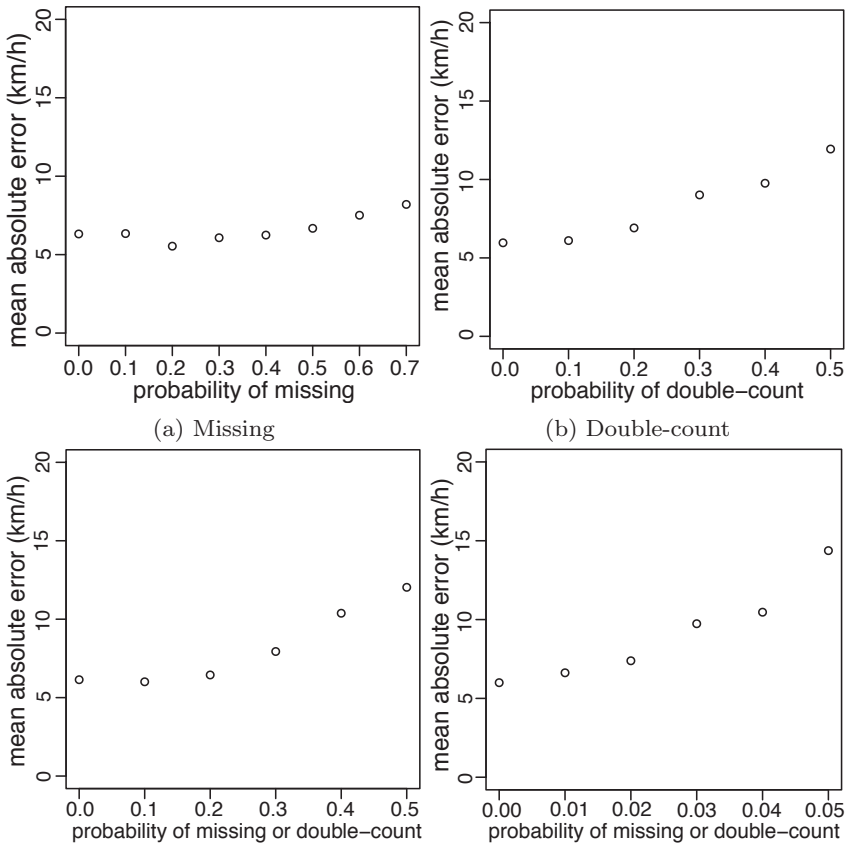


Fig. 3.28 An extreme setting: true velocity and velocity estimated by the proposed method for a time interval $\Delta t = 10$ (second) in an artificial instance. The average number of vehicles $M_{\mathrm{art}} = 10$. Note that we have used the Tukey boxplot [127]

show that our method is robust against the counting error even when 40 percent of the vehicles are missing or double-counted. The counting error for the double-count has a greater effect on the accuracy since the double-counted vehicles do not become independent. We also tried a more extreme (but unrealistic) setting, where vehicles are miscounted independently in every second during the one minute of the N observations. As shown in Figure 3.29 (d), the proposed method cannot achieve good accuracy when the noise proportion is more than 5 percent. This is because this type of noise makes the correlation between the observation sequences quite low. Since the velocity estimated by the proposed model is high when the correlation is low, it overestimates the velocity when it is given such fluctuating observations.

To derive the observation model, we assume that the positions of the vehicles at time zero are random and independent from each other. In real world traffic data, the positions of the vehicles are not exactly random and the number of vehicles in the current observation depends on and is correlated to the one of the former observations even if the observations do not have an overlapping area between them. This correlation may cause that our estimation results have an underestimation bias since the strong correlation means larger overlapping area and slower velocity in the proposed model. The experiments using the real-world datasets indicated that such a bias occurred. While the proposed method can estimate velocities with considerable accuracy, the estimated mean values of the velocities are slightly lower than the means of the true velocities in four of the five real-world datasets described in Section 3.5.3. However, since this bias is only -2.7 (km/h) on average and is statistically signifi-



(a) Missing (b) Double-count

(c) Missing or double-count (d) Missing or double-count every second (extreme setting)

Fig. 3.29 Relationship between probability of artificially added noise to the count and mean absolute error.

cant (paired t-test, $p \leq 0.05$) in only one of the five datasets (US-101 dataset in the NGSIM datasets), we conclude that the bias is not so large. Solving this problem will be part of our future work.

3.6.6 Other Applications of Velocity Estimation

While we assume for the proposed model that v is the same for all of the vehicles during the timestamps, our experiment showed that the model is robust enough for situations where each vehicle has a different velocity. Additionally, because the definition in Eqs. (3.41) and (3.43) is *invariant with respect to the movement direction of the vehicles*, we can use it in situations where the observations $\boldsymbol{x}$ include vehicles moving inbound as well as outbound without any changes to the model or algorithm. It estimates a common absolute value of the velocity for both lane directions. The directional invariance is preserved even in other multi-directional cases, such as at intersections, as shown in Fig. 3.30, which shows an image captured in

Fig. 3.30 Traffic at intersection captured by web camera in Nairobi [75].

Nairobi, Kenya [75]. The above capabilities also mean that we can use the proposed method for estimating the velocities of crowds, molecules, etc. It can also be used to estimate the velocity separately for each of the three legs of that intersection if we can separately obtain the number of vehicles in each leg.

The Gaussian approximation has another advantage in that we can straightforwardly extend the model so that it can use low-level features by taking $\boldsymbol{x}$ to be real numbers, where the features need to be such that larger values correspond to larger vehicle counts. For example, in the case of analyzing web-camera images, we can use the total area that may correspond to moving objects in an image (TAM) [35, 36, 77, 78, 86, 87] as $\boldsymbol{x}$. Since we can usually obtain such low-level features more easily than the number of vehicles, this ability is quite useful when we do not have any way to determine the explicit number of vehicles from the raw input data. Here, we examined this extension by using the web-camera dataset in Section 3.5.3. We tested the proposed method by inputting TAM as $\boldsymbol{x}$. Figure 3.31 compares manually measured velocity, velocity estimated using the proposed method with TAM as input, and velocity estimated using the proposed method with the vehicle count as input. We can see that the degradation in the estimation accuracy is small when we use the TAM as input of the proposed method.

3.7 Summary

This study tackled the novel task of *traffic-flow estimation from images without recognizing any vehicles or using any labeled training data* for non-intrusive lightweight traffic monitoring systems. We formulated the task as a Bayesian density estimation problem by deriving a new model for traffic-volume estimation and one for traffic-velocity estimation.

For the traffic volume estimation, we use a new variant of an infinite GMM, where each of the components has a particular interpretation of the vehicle count. We showed that the SBP prior works well to regularize the solution. Surprisingly, our completely unsupervised approach without any training data was comparable to or

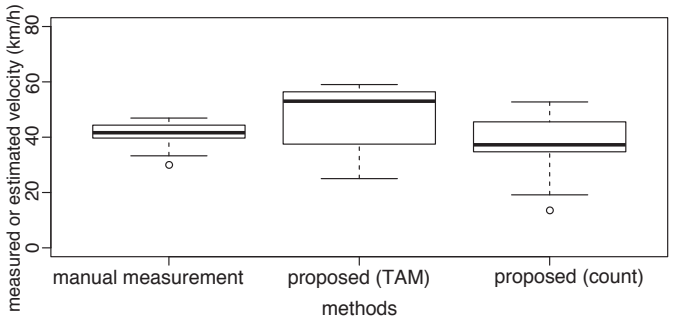


Fig. 3.31 Manually measured velocity, velocity estimated using the proposed method with the low-level image feature, TAM, as input, and velocity estimated using the proposed method with the vehicle count as input. Note that we have used the Tukey boxplot [127].

even better than the supervised alternatives. The proposed method does not rely on any knowledge or labeled training data tailored to the objects being counted, which constitutes a clear advantage in practice. Thus, it would be interesting to use our approach in other applications, such as on crowd or microscopic images. Moreover, our traffic-volume-estimation method can be applied to other tasks of object counting in a variety of literature, such as counting the number of a specific word in a text [133–136] and the number of times a specific pattern appears in time-series data [137–139].

For the traffic velocity estimation, we derive a new model that represents the likelihood of the traffic velocity given the traffic volumes. For this model, we propose an efficient approximation method of estimating the PM of the traffic velocity by using slice sampling. In the experiments, the proposed approach was good enough for our applications. Our approach can naturally be applied to the traffic-velocity-estimation problem using other data sources, such as inductive loop and GPS.

Chapter.4

Bayesian Image Super Resolution

4.1 Introduction

Super-resolution (SR) is a promising technology that is expected to be applied to microscope time series images, satellite photographs, and sequential video frames. It is an image-processing technique that makes it possible to estimate a spatially high-resolution (HR) image of a scene from corresponding multiple low-resolution (LR) images. In SR, we assume that the LR images are affected by warping, blurring, and noise, and we solve the registration problem of LR images in addition to the image-restoration problem of the registered LR images. Since the earliest work by Tsai and Huang [140], SR has been achieved using various methods [141–149] and good overviews of these methods are given in [150–155].

Generally, SR is an ill-posed inverse problem because inverting the blur process without amplifying the effect of the noise is difficult [151]. In other words, the degrees of freedom of the HR image and pixel-wise observation noise are always higher than the dimensionality of the observed LR images, so complete determination of an HR image is impossible.

In this chapter, we tackle the SR problem within the framework of Bayesian optimal estimation. We solve the ill-posed inverse problem by using image prior appropriately regularizing the degrees of freedom of the HR image and the stable nature of Bayesian optimal estimation. For the image prior in SR with Bayesian formulation [141, 146], various Markov random field (MRF) priors [38, 39, 143, 144, 148, 156–158], the total variation (TV) prior [142, 147], Huber prior [145], and patch-based priors [159] have been used in image processing. These can well represent image properties and have good performance in SR, image restoration, and other applications. As the estimation function for SR, we believe the PM is suitable because we usually evaluate the accuracy of SR methods by mean square error, and PM is the optimal estimation function when using that as the evaluation criterion. To determine the exact PM of

the HR image, all parameters other than the HR image should be marginalized out over the joint posterior distribution without using any point estimation. According to this meaning, the previous methods [141–144, 146, 147] are not optimal.

We propose an SR methods that use a novel unified warping, blurring, and downsampling model for SR and "causal" and "compound" Gaussian MRF priors with VB to calculate the Bayesian optimal estimation function, PM [38, 39]. This is a straightforward approach, but it was not proposed earlier possibly because an important limitation of VB is that a conjugate prior is needed. We solve this problem through simple Taylor approximations introduced in Chapter 2. In experiments, we evaluate the proposed method by comparing it with existing methods.

4.2 Related Work

In the SR problem, to deal with warping, blurring, and downsampling, a linear transformation model is frequently used [141–144]. Warping is usually limited with planar rotation and parallel translation. Blurring is defined by using a point spread function (PSF); a square or Gaussian type PSF is common. Downsampling denotes sampling from an HR image to construct an LR image. Downsampling sometimes includes anti-aliasing. Since these three transformations are linear, they can be combined into a single transformation matrix. As for the noise model, pixel-independent additive white Gaussian noise (AWGN) is usually used.

The Bayesian framework, especially the HR image prior, is quite useful for SR. The HR image prior provides appropriate smoothness between neighboring pixel luminances. A common type of HR image prior imposes an L2-norm penalty on differences between horizontally and vertically adjacent pixel luminances (the first derivative). The L1-norm of the first derivative is sometimes used, and it has the advantage of robust inference against outliers. The TV prior [142] uses the L1-norm of the gradient vector. The Huber prior [145] is a mixture prior of L1- and L2-norms. The SAR model [146, 147, 160] uses the response of a two-dimensional Laplacian filter (the second derivative). The Gaussian process prior [141] has neighboring pixels spread according to a Gaussian distribution. Besides the degree of smoothness between neighboring pixels, information regarding the discontinuity, or equivalently, the edges or line process, is also useful for inference. A common type of prior implementing edges is the "compound" Gaussian MRF prior that was introduced by Geman & Geman [156] and is widely used [39, 148, 157, 158]. With respect to the "compound" Gaussian MRF prior, the normalizing constant, or equivalently, the partition function, is usually computationally infeasible because it has an exponential calculation cost with respect to the dimensionality of the line process. Recently, Kanemura *et*

al. [143,144] introduced a "causal" type of Gaussian MRF prior [38] as an approximation of the "compound" Gaussian MRF prior whose calculation cost is polynomial. We try to improve the prior and approximation in this study.

The estimation function should be derived from an objective function. As the objective function, a posterior distribution has been widely used. Since the posterior distribution usually includes both the HR image and registration parameters, the joint MAP solution [146] is a suitable estimation function for this objective function. Other than the joint MAP, the use of the marginalized maximum likelihood (ML) [141, 143] or marginalized MAP [145] has been proposed. Tipping *et al.* [141] and Kanemura *et al.* [143, 144] determine the registration parameters by using ML inference, where the HR image is marginalized out, and determine the HR image by using the MAP inference. Pickup *et al.* [145] determines the HR image by using the MAP inference, wherein the registration uncertainties are marginalized out, and it is assumed that the registration parameters are pre-registered by using standard registration techniques. Marginalized ML is also called type-II ML, evidence approximation, or empirical Bayes. Marginalized ML has no registration prior, unlike marginalized MAP. Pickup *et al.* [145] reported that marginalized MAP is superior to both joint MAP and marginalized ML. We evaluate the accuracy of SR methods in terms of mean square error. Therefore, we believe it is natural to use it as the evaluation criterion. For the objective function based on mean square error, PM is an optimal estimation function. The VB approach [142] seems to approximately determine the PM of the HR image, although these authors assume some registration parameters are known and use point-estimate model parameters obtained by ML inference. To determine the exact PM of the HR image, all parameters other than the HR image should be marginalized out over the joint posterior distribution.

The type of computational algorithm to use is not as substantial a problem as the choice of model and evaluation criterion, but it is still important. Since almost all good estimation functions cannot be exactly determined because of difficult analytical integration or an exponential calculation cost, some approximation methods need to be introduced. Also, parameter tuning is necessary with many numerical optimization methods; e.g., of the initial value and the step-width settings in gradient methods. Specifically, in early work done on image restoration, an annealing method was used for the joint MAP solution [156, 161]. For marginalized ML and marginalized MAP solutions, the scaled conjugate gradients algorithm was used [141, 145]. In recent studies, the variational expectation-maximization (EM) algorithm has been applied, which includes the gradient method in the M step [143, 144]. The VB approach has also been applied [142]. This method includes nested optimization of the majorization-minimization approach. This approach seems to affect both the HR im-

age prior and estimation function. Specifically, it modifies the TV prior to include a discontinuity parameter (called local spatial activity). In addition, this parameter is point-estimated when the HR image is inferred.

4.3 Problem Setting

Our task is to estimate an HR grayscale image, $\boldsymbol{x} \in \mathbb{R}^{N_{\boldsymbol{x}}}$, from the observed multiple LR grayscale images, $\boldsymbol{Y} \equiv \{\boldsymbol{y}_l\}_{l=1}^{L}$, $\boldsymbol{y}_l \in \mathbb{R}^{N_{\boldsymbol{y}}}$. Images $\boldsymbol{y}_l$ and $\boldsymbol{x}$ are regarded as lexicographically stacked vectors. The number of pixels for each LR image, $N_{\boldsymbol{y}}$, is assumed to be less than that of the HR image, $N_{\boldsymbol{x}}$; i.e., $N_{\boldsymbol{y}} < N_{\boldsymbol{x}}$. We conduct this estimation using an SR technique whose resolution enhancement factor is $\alpha \equiv \sqrt{N_{\boldsymbol{x}}/N_{\boldsymbol{y}}}$ (> 1). Although we define the range of a pixel luminance value as infinite, we use -1 for black, $+1$ for white, and values between -1 and $+1$ for gradual gray.

4.4 Posterior Mean Estimation for Super Resolution

We formalize this estimation problem as the estimates of $\boldsymbol{x}$ by the estimation function, $\boldsymbol{x}^*(\boldsymbol{Y})$, to which the observed variables $\boldsymbol{Y}$ have been input.

One of the most commonly used error functions $\mathrm{Error}(\boldsymbol{x}, \boldsymbol{x}^*(\boldsymbol{Y}))$ for evaluating the estimated image quality is the L2-norm (mean square error). We define the error function $\mathrm{Error}(\boldsymbol{x}, \boldsymbol{x}^*(\boldsymbol{Y}))$ for the task as the squared difference between $\boldsymbol{x}$ and the estimates by the estimation function $\boldsymbol{x}^*(\boldsymbol{Y})$:

$$\mathrm{Error}(\boldsymbol{x}, \boldsymbol{x}^*(\boldsymbol{Y})) \equiv \|\boldsymbol{x} - \boldsymbol{x}^*(\boldsymbol{Y})\|_2^2, \tag{4.1}$$

Using the model parameters $\boldsymbol{\theta}$, which are explicitly defined later, we define the evaluation criterion as the minimization of the population mean of the error function Eq. (4.1):

$$\underset{\boldsymbol{x}^*(\boldsymbol{Y})}{\mathrm{argmin}} \left\langle \|\boldsymbol{x} - \boldsymbol{x}^*(\boldsymbol{Y})\|_2^2 \right\rangle_{p(\boldsymbol{Y}, \boldsymbol{x}, \boldsymbol{\theta})}. \tag{4.2}$$

Then, we can derive the optimal estimation function using the result in Eq. (2.14) as the PM,

$$\begin{aligned} \hat{\boldsymbol{x}}^*(\boldsymbol{Y}) &= \underset{\boldsymbol{x}^*(\boldsymbol{Y})}{\mathrm{argmin}} \left\langle \|\boldsymbol{x} - \boldsymbol{x}^*(\boldsymbol{Y})\|_2^2 \right\rangle_{p(\boldsymbol{Y}, \boldsymbol{x}, \boldsymbol{\theta})} \\ &= \int \boldsymbol{x}\, p(\boldsymbol{x}|\boldsymbol{Y}) \mathrm{d}\boldsymbol{x}, \end{aligned} \tag{4.3}$$

where the posterior distribution $p(\boldsymbol{x}|\boldsymbol{Y})$ represents the probability distribution of the HR image $\boldsymbol{x}$ given the observed multiple LR images $\boldsymbol{Y}$. Note that $p(\boldsymbol{x}|\boldsymbol{Y})$ requires marginalization of all parameters other than $\boldsymbol{x}$ over $p(\boldsymbol{\theta}|\boldsymbol{Y})$.

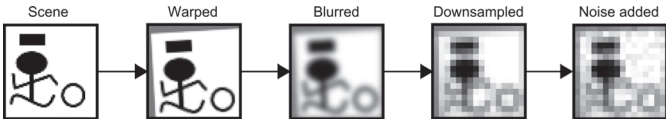


Fig. 4.1 An illustration of the image observation process

4.5 Probabilistic Hidden Structure Modeling for Super Resolution

The estimation function of the HR image $\hat{\boldsymbol{x}}^*(\boldsymbol{Y})$ is found through the posterior distribution $p(\boldsymbol{x}|\boldsymbol{Y})$ from Eq. (4.3). The posterior $p(\boldsymbol{x}|\boldsymbol{Y})$ for $\boldsymbol{x}$ can be decomposed into an observation model for the LR images $p(\boldsymbol{Y}|\boldsymbol{x})$ that is conditioned on the HR image $\boldsymbol{x}$ and the HR image prior $p(\boldsymbol{x})$:

$$p(\boldsymbol{x}|\boldsymbol{Y}) = \frac{p(\boldsymbol{Y}|\boldsymbol{x})p(\boldsymbol{x})}{\int p(\boldsymbol{Y}|\boldsymbol{x})p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}}. \tag{4.4}$$

We define the observation model $p(\boldsymbol{Y}|\boldsymbol{x})$ conditioned on $\boldsymbol{x}$ and the prior model $p(\boldsymbol{x})$ in the following subsections. In this chapter, we introduce two kind of HR image prior, that is "causal" Gaussian MRF prior and "compound" Gaussian MRF prior. We derive two algorithms based on these priors as Algorithm 1 and Algorithm 2, respectively, and compare their performance through experiments.

4.5.1 Observation Model

The image observation process is modeled as shown in Fig. 4.1; the HR image $\boldsymbol{x}$ is geometrically warped, blurred, downsampled, and corrupted by noise $\boldsymbol{\epsilon}_l$ to form the observed LR image $\boldsymbol{y}_l$:

$$\boldsymbol{y}_l \equiv \boldsymbol{W}(\boldsymbol{\phi}_l)\boldsymbol{x} + \boldsymbol{\epsilon}_l, \tag{4.5}$$

or, more strictly,

$$p(\boldsymbol{Y}|\boldsymbol{x}, \beta, \boldsymbol{\Phi}) \equiv \prod_{l=1}^{L} \mathcal{N}(\boldsymbol{y}_l|\boldsymbol{W}(\boldsymbol{\phi}_l)\boldsymbol{x}, \beta^{-1}\boldsymbol{I}). \tag{4.6}$$

The $\boldsymbol{\epsilon}_l \in \mathbb{R}^{N_{\boldsymbol{y}}}$ is AWGN with precision (inverse variance) β (> 0). Here, $\boldsymbol{W}(\boldsymbol{\phi}_l)$ is the $N_{\boldsymbol{y}} \times N_{\boldsymbol{x}}$ transformation matrix that is simultaneously used for warping, blurring,

and downsampling. It is defined as

$$\boldsymbol{W}(\boldsymbol{\phi}_l)_{j,i} \equiv \frac{\mathcal{N}\left(\vec{\chi}(\theta_l, \vec{o}_l, \vec{\zeta}_j, \vec{\xi}_i)|0, \gamma_l^{-1}\boldsymbol{I}\right)}{\sum_{i' \in \mathcal{I}} \mathcal{N}\left(\vec{\chi}(\theta_l, \vec{o}_l, \vec{\zeta}_j, \vec{\xi}_{i'})|0, \gamma_l^{-1}\boldsymbol{I}\right)}, \quad \text{where} \tag{4.7}$$

$$\vec{\chi}(\theta, \vec{o}, \vec{\zeta}, \vec{\xi}) \equiv \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \left(\alpha\vec{\zeta} - \vec{o}\right) - \vec{\xi}, \tag{4.8}$$

where $\mathcal{I}$ represents the extent of the summation (explained in the next paragraph), and the vectors $\vec{\xi}_i$ and $\vec{\zeta}_j$ respectively denote the two-dimensional positions of the i-th pixel of the original HR image and the j-th pixel of the observed LR image. We define the center of each image as the origin and the size of each pixel is 1 by 1. For example, regarding an HR image with 40×40 pixels, each $\vec{\xi}$ represents $[-19.5, -19.5]^\top, [-18.5, -19.5]^\top, ..., [19.5, 19.5]^\top$. θ_l and $\vec{o}_l$ represent the warping parameters of the l-th LR image: the rotational motion parameter and translational motion parameter. The Gaussian distribution in (4.7) represents a Gaussian PSF that defines the blur, and γ_l (> 0) represents its precision parameter. In this study, we assume γ_l also differs for each observed image. These transformation parameters are packed into $\boldsymbol{\phi}_l$, which is defined as

$$\boldsymbol{\Phi} \equiv \{\boldsymbol{\phi}_l\}_{l=1}^{L},\ \boldsymbol{\phi}_l \equiv [\phi_{l,k}]_{k=1}^{4} \equiv [\theta_l, [\vec{o}_l]_h, [\vec{o}_l]_v, \gamma_l]^\top, \tag{4.9}$$

where subscripts h and v, respectively, denote horizontal and vertical positions on the image.

In previous works [141, 143, 144], the extent of $\mathcal{I}$ was defined as the extent of the HR image. According to this definition, however, the shape of the PSF is no longer Gaussian. For example, at the corner of the HR image, the shape is not omnidirectional but limited in a way such as that of a quadrant. In this study, the extent of $\mathcal{I}$ is defined as infinite, and the luminance values outside the HR image are defined as 0 (middle gray). This normalization term faithfully represents the Gaussian PSF. We also found that this normalization term is exactly given by using the elliptic theta function ϑ_3, and we can rewrite $\boldsymbol{W}(\boldsymbol{\phi}_l)$ as

$$\boldsymbol{W}(\boldsymbol{\phi}_l)_{j,i} = \frac{\mathcal{N}\left(\vec{\chi}(\theta_l, \vec{o}_l, \vec{\zeta}_j, \vec{\xi}_i)|0, \gamma_l^{-1}\boldsymbol{I}\right)}{\vartheta_3\left(\left[\vec{\chi}(\theta_l, \vec{o}_l, \vec{\zeta}_j, \vec{\xi}_i)\right]_h, \mathrm{e}^{-\frac{2\pi^2}{\gamma_l}}\right)\vartheta_3\left(\left[\vec{\chi}(\theta_l, \vec{o}_l, \vec{\zeta}_j, \vec{\xi}_i)\right]_v, \mathrm{e}^{-\frac{2\pi^2}{\gamma_l}}\right)}, \quad \text{where} \tag{4.10}$$

$$\vartheta_3(u, q) \equiv 1 + 2\sum_{n=1}^{\infty} q^{n^2} \cos 2n\pi u. \tag{4.11}$$

The elliptic theta function includes an infinite series, but it is easily determined numerically because the convergence is quite fast. In (4.10), the normalization term

(the denominator of the right-hand side) seems to depend on i because $\vec{\chi}\big(\theta_l, \vec{o}_l, \vec{\zeta}_j, \vec{\xi}_i\big)$ includes $\vec{\xi}_i$, but this is not true. Because the elliptic theta function is a periodic function with respect to the argument u with period 1, and $\vec{\chi}\big(\theta_l, \vec{o}_l, \vec{\zeta}_j, \vec{\xi}_i\big)$ can only take discrete values with step size 1 for the horizontal and vertical directions, the normalization term has the same value with respect to i.

4.5.2 Causal Gaussian MRF prior

For Algorithm 1, we introduce a "causal" Gaussian MRF prior [38, 143, 144] for the HR image and additional latent variables. These latent variables are called the line process that controls the local correlation among pixel luminances. The introduction of the latent variables enables explicit expression of the possible discontinuity in the HR image. The line process, $\boldsymbol{\eta}$, consists of binary variables $\eta_{i,j} \in \{0, 1\}$ for all adjacent pixel pairs i and j. Its size equals $N_{\boldsymbol{\eta}} \equiv 2N_{\boldsymbol{x}} -$ [number of HR image's horizontal pixels] $-$ [number of HR image's vertical pixels]. We define the prior as

$$
\begin{aligned}
&p(\boldsymbol{x}, \boldsymbol{\eta}|\lambda, \rho, \kappa) \equiv p(\boldsymbol{x}|\boldsymbol{\eta}, \rho, \kappa)p(\boldsymbol{\eta}|\lambda) \qquad (4.12)\\
&= \exp\Bigg[-\lambda \sum_{i\sim j}(1-\eta_{i,j}) - \frac{\rho}{2}\sum_{i\sim j}\eta_{i,j}(x_i - x_j)^2 - \frac{\kappa}{2}\|\boldsymbol{x}\|_2^2 \\
&\qquad + \frac{1}{2}\ln\left|\frac{\boldsymbol{A}(\boldsymbol{\eta}, \rho, \kappa)}{2\pi}\right| + N_{\boldsymbol{\eta}} \ln\ \mathrm{Sigmoid}(\lambda)\Bigg],
\end{aligned}
$$

where

$$
p(\boldsymbol{\eta}|\lambda) \equiv \prod_{i\sim j} \mathrm{Bernoulli}\left(\eta_{i,j}|\mathrm{Sigmoid}(\lambda)\right), \qquad (4.13)
$$

$$
p(\boldsymbol{x}|\boldsymbol{\eta}, \rho, \kappa) \equiv \mathcal{N}(\boldsymbol{x}|\boldsymbol{0}, \boldsymbol{A}(\boldsymbol{\eta}, \rho, \kappa)^{-1}), \qquad (4.14)
$$

$$
\boldsymbol{A}(\boldsymbol{\eta}, \rho, \kappa)_{i,j} \equiv \begin{cases} \rho\sum_{k\sim i}\eta_{i,k} + \kappa, & i = j, \\ -\rho\eta_{i,j}, & i \sim j, \\ 0, & \text{otherwise}, \end{cases} \qquad (4.15)
$$

where the summation $\sum_{i\sim j}$ is taken over all pairs of adjacent pixels. The notation $i \sim j$ means that the i-th and j-th pixels are adjacent in the upward, downward, leftward, and rightward directions. The line process $\boldsymbol{\eta}$ switches the local characteristics of the prior. It indicates whether two adjacent pixels take similar values or independent values. When $\eta_{i,j} = 1$, the i-th and the j-th pixels are strongly smoothed according to the quadratic penalty, whereas there is no smoothing when $\eta_{i,j} = 0$. The hyperparameter λ (> 0) is an edge penalty parameter that prevents $\eta_{i,j}$ from excessively taking edges. Note that λ is restricted to positive values because a negative λ leads to a reward rather than a penalty for taking edges. ρ (> 0) is a smoothness

parameter that prevents the differences in adjacent pixel luminances from becoming large, and κ (> 0) is a contrast parameter that prevents $\boldsymbol{x}$ from taking an improperly large absolute value. On the other hand, in previous works [143,144], κ is assumed to be 0, which results in an improper normalizing constant (see Discussion). $\boldsymbol{A}(\boldsymbol{\eta}, \rho, \kappa)$ is the $N_{\boldsymbol{x}} \times N_{\boldsymbol{x}}$ precision matrix of $\boldsymbol{x}$.

We have defined the introduced causal Gaussian MRF prior in the joint distribution form of $\boldsymbol{x}$ and $\boldsymbol{\eta}$, i.e., $p(\boldsymbol{\eta})p(\boldsymbol{x}|\boldsymbol{\eta})$. We call such a model "causal" because $\boldsymbol{\eta}$ seems to cause $\boldsymbol{x}$. The MRF model is defined as having the property

$$p(x_i|\boldsymbol{x}\backslash x_i, \boldsymbol{\eta}) = p(x_i|\boldsymbol{x}_{\mathcal{L}(i)}, \boldsymbol{\eta}_{i,\mathcal{L}(i)}) \tag{4.16}$$

in this case; i.e., the conditional distribution of a random variable, x_i, given all other variables, $\boldsymbol{x}\backslash x_i$ and $\boldsymbol{\eta}$, equals the conditional distribution of the random variable, x_i, given its "neighboring" variables, $\boldsymbol{x}_{\mathcal{L}(i)}$ and $\boldsymbol{\eta}_{i,\mathcal{L}(i)}$. If this conditional distribution is a Gaussian distribution, such an MRF is called a Gaussian MRF.

4.5.3 Compound Gaussian MRF prior

For Algorithm 2, we use a "compound" Gaussian MRF prior for the HR image and the latent variables $\boldsymbol{\eta}$ representing the edges, called a line process, which is same to "causal" one. It is a compounded distribution of the Gaussian MRF model and the line process proposed by [156], which is widely used [39, 148, 157, 158] and can simultaneously represent smoothness and discontinuity of the image. It is defined as

$$\begin{aligned}
&p(\boldsymbol{x}, \boldsymbol{\eta}|\lambda, \rho, \kappa) \\
&\equiv \frac{\exp\left[-\lambda\sum_{i\sim j}(1-\eta_{i,j}) - \frac{\rho}{2}\sum_{i\sim j}\eta_{i,j}(x_i - x_j)^2 - \frac{\kappa}{2}\|\boldsymbol{x}\|_2^2\right]}{\sum_{\boldsymbol{\eta}}\int \exp\left[-\lambda\sum_{i\sim j}(1-\eta_{i,j}) - \frac{\rho}{2}\sum_{i\sim j}\eta_{i,j}(x_i - x_j)^2 - \frac{\kappa}{2}\|\boldsymbol{x}\|_2^2\right]\mathrm{d}\boldsymbol{x}} \\
&= \exp\left[-\lambda\sum_{i\sim j}(1-\eta_{i,j}) - \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{A}(\boldsymbol{\eta}, \rho, \kappa)\boldsymbol{x}\right. \\
&\quad \left. -\ln\sum_{\boldsymbol{\eta}}\exp\left\{-\lambda\sum_{i\sim j}(1-\eta_{i,j}) - \frac{1}{2}\ln\left|\frac{1}{2\pi}\boldsymbol{A}(\boldsymbol{\eta}, \rho, \kappa)\right|\right\}\right],
\end{aligned} \tag{4.17}$$

where the definitions for $\boldsymbol{\eta}$, λ, ρ, κ, and $\boldsymbol{A}$ are same to "causal" one.

As shown in the previous subsection, the "causal" Gaussian MRF prior is defined as the joint distribution of $\boldsymbol{x}$ and $\boldsymbol{\eta}$ in the form of $p(\boldsymbol{\eta})p(\boldsymbol{x}|\boldsymbol{\eta})$, and it differs from the "compound" one in that it is not simultaneously normalized about both $\boldsymbol{x}$ and $\boldsymbol{\eta}$ like Eq. (4.17). A "causal" one is an approximation of the "compound" one, and it is easier to use than the "compound" one because simultaneous normalization of a "compound" one has an exponential-order calculation cost with respect to the

dimensionality of the line process; the calculation cost of a "causal" one is polynomial. Additionally, though both $\boldsymbol{x}$ and $\boldsymbol{\eta}$ of a "compound" one has a Markov property, in the "causal" one only $\boldsymbol{x}$ has a Markov property. However, in Eq. (4.17), ignoring $\ln|\boldsymbol{A}|$ as in [143, 144] makes them take the same form and breaks either property. Therefore, in Section 4.6.2, we propose a new approximation that does not ignore $\ln|\boldsymbol{A}|$.

4.5.4 Hyperparameter Priors and Registration Parameter Priors

We define the prior distributions for the hyperparameters of two of the HR image priors:

$$\begin{aligned} p(\lambda,\rho,\kappa,\beta) \equiv & \operatorname{Gamma}(\lambda|a_\lambda^{(0)},b_\lambda^{(0)})\operatorname{Gamma}(\rho|a_\rho^{(0)},b_\rho^{(0)}) \\ & \times \operatorname{Gamma}(\kappa|a_\kappa^{(0)},b_\kappa^{(0)})\operatorname{Gamma}(\beta|a_\beta^{(0)},b_\beta^{(0)}), \end{aligned} \tag{4.18}$$

where the form of prior distributions of "causal" Gaussian MRF prior for Algorithm 1 is same to one of "compound" Gaussian MRF prior for Algorithm 2, whereas their parameters are different from each other and we define them as non-informative as possible in the following subsection for each algorithm. Superscript (0) is added because we use these parameters as the initial values of VB later.

For the registration parameters including the blurring parameter, we also define the corresponding prior as

$$p(\boldsymbol{\Phi}) \equiv \prod_{l=1}^{L} \mathcal{N}(\boldsymbol{\phi}_l|\boldsymbol{\mu}_{\boldsymbol{\phi}_l}^{(0)},\boldsymbol{\Sigma}_{\boldsymbol{\phi}_l}^{(0)}), \quad \text{where} \tag{4.19}$$

$$\boldsymbol{\mu}_{\boldsymbol{\phi}_l}^{(0)} \equiv [0,0,0,12/\alpha^2],\ \boldsymbol{\Sigma}_{\boldsymbol{\phi}_l}^{(0)} \equiv \operatorname{diag}[10^{-3},10^{0},10^{0},10^{-3}]. \tag{4.20}$$

For the rotational motion parameter θ_l, the prior assumes 0 ± 1.81 degree ($\frac{180}{\pi}\sqrt{10^{-3}} \approx 1.81$). This assumption is considered suitable for this SR task. Similarly, an assumption of 0 ± 1 pixels for translational motion parameters $[\vec{o}_l]_h$ and $[\vec{o}_l]_v$ is considered suitable. For blurring parameter γ_l, $\mu_{\gamma_l}^{(0)}$ is taken to be the value equivalent to the anti-aliasing of the scale factor α.

4.5.5 Joint Distribution

The joint distribution for all of the random variables $\boldsymbol{z} \equiv [\boldsymbol{x},\boldsymbol{\eta},\lambda,\rho,\kappa,\beta,\boldsymbol{\Phi}]$, as well as $\boldsymbol{Y}$ can now be explicitly given as

$$p(\boldsymbol{Y},\boldsymbol{z}) = p(\boldsymbol{Y}|\boldsymbol{x},\beta,\boldsymbol{\Phi})p(\boldsymbol{x},\boldsymbol{\eta}|\lambda,\rho,\kappa)p(\lambda,\rho,\kappa,\beta)p(\boldsymbol{\Phi}), \tag{4.21}$$

Once the joint distribution is obtained, we can derive all the marginal and conditional distributions; e.g., the posterior distribution $p(\boldsymbol{z}|\boldsymbol{Y})$ and joint distribution of the HR

and LR images $p(\boldsymbol{Y}, \boldsymbol{x})$.

4.6 Variational Bayes Algorithm for Posterior Mean Estimation

Though we could derive the optimal estimation function and proposed the probabilistic models according to that, we cannot obtain the analytical solutions of the posterior distribution $p(\boldsymbol{z}|\boldsymbol{Y})$ and marginalized posterior distribution $p(\boldsymbol{x}|\boldsymbol{Y})$. Consequently, we have to rely on approximations, that is VB and Taylor approximations. Here, we derive two algorithms: Algorithm 1 based on "causal" Gaussian MRF prior and Algorithm 2 based on "compound" Gaussian MRF prior.

4.6.1 Variational Bayes

VB [52] provides a trial distribution $q(\boldsymbol{z})$ that approximates the true posterior. According to the formulation in Chapter 2, we assume a trial distribution $q(\boldsymbol{z})$ that approximates the true posterior in a factorized form:

$$q(\boldsymbol{z}) \equiv q(\boldsymbol{x})q(\boldsymbol{\eta})q(\lambda, \rho, \kappa, \beta)q(\boldsymbol{\Phi}). \tag{4.22}$$

We identify the optimal trial distribution that minimizes the KL divergence from the trial distribution to the true posterior distribution, $D_{\mathrm{KL}}(q(\boldsymbol{z})\|p(\boldsymbol{z}|\boldsymbol{Y}))$, as the best approximation of the true distribution.

Under the factorization assumption of the trial distribution and the extremal condition of the KL divergence, each optimal trial distribution should satisfy the self-consistent equations,

$$q^{(0)}(\boldsymbol{z}_i) \equiv p(\boldsymbol{z}_i), \tag{4.23}$$

$$q^{(t+1)}(\boldsymbol{z}_i) \propto \exp\langle \ln p(\boldsymbol{z}|\boldsymbol{Y})\rangle_{\prod_{j\neq i} q^{(t)}(\boldsymbol{z}_j)}, \tag{4.24}$$

4.6.2 Taylor Approximations

Although VB is a widely used general framework, its application is difficult in practice because it requires a conjugate prior. The prior distributions we have introduced are not conjugate priors. However, we have found that simple Taylor approximations make them conjugate and enable the analytical exact expectations in (4.24).

Here, to simplify the notation, we define the mean values of the latent variables $\boldsymbol{\eta}$, the hyper parameters $\lambda, \rho, \kappa, \beta$, and the registration parameters $\boldsymbol{\phi}_l$ over the trial distributions in the step number t of the updates of VB as $\boldsymbol{\mu}_{\boldsymbol{\eta}}^{(t)}$, $\mu_{\lambda}^{(t)}$, $\mu_{\rho}^{(t)}$, $\mu_{\kappa}^{(t)}$, $\mu_{\beta}^{(t)}$, $\boldsymbol{\mu}_{\boldsymbol{\phi}_l}^{(t)}$.

Specifically, we use first-order Taylor approximations for non-linear terms, which are introduced in Chapter 2.

For the observation model in Eq. (4.6), $\boldsymbol{W}(\boldsymbol{\phi}_l)$ is approximated around $\boldsymbol{\phi}_l = \boldsymbol{\mu}_{\boldsymbol{\phi}_l}^{(t)}$,

$$\boldsymbol{W}(\boldsymbol{\phi}_l) \approx \boldsymbol{W}_l^{(t)} + \sum_{k=1}^{4} [\boldsymbol{\phi}_l - \boldsymbol{\mu}_{\boldsymbol{\phi}_l}^{(t)}]_k \boldsymbol{W}_{l,k}^{\prime(t)}, \tag{4.25}$$

where

$$\boldsymbol{W}_l^{(t)} \equiv \boldsymbol{W}(\boldsymbol{\mu}_{\boldsymbol{\phi}_l}^{(t)}), \tag{4.26}$$

$$\boldsymbol{W}_{l,k}^{\prime(t)} \equiv \left. \frac{\partial \boldsymbol{W}(\boldsymbol{\phi}_l)}{\partial \phi_{l,k}} \right|_{\boldsymbol{\phi}_l = \boldsymbol{\mu}_{\boldsymbol{\phi}_l}^{(t)}}. \tag{4.27}$$

For the "causal" Gaussian MRF prior in Eq. (4.12), $\ln |\boldsymbol{A}(\boldsymbol{\eta}, \rho, \kappa)|$ is approximated around $[\boldsymbol{\eta}, \ln \rho, \ln \kappa] = [\boldsymbol{\mu}_{\boldsymbol{\eta}}^{(t)}, \ln \mu_\rho^{(t)}, \ln \mu_\kappa^{(t)}]$,

$$\begin{aligned}
\ln |\boldsymbol{A}(\boldsymbol{\eta}, \rho, \kappa)| &\approx \ln \left| \boldsymbol{A}(\boldsymbol{\mu}_{\boldsymbol{\eta}}^{(t)}, \mu_\rho^{(t)}, \mu_\kappa^{(t)}) \right| \\
&+ \mathrm{tr} \Bigg(\boldsymbol{A}(\boldsymbol{\mu}_{\boldsymbol{\eta}}^{(t)}, \mu_\rho^{(t)}, \mu_\kappa^{(t)})^{-1} \Big[\mu_\rho^{(t)} \boldsymbol{A}(\boldsymbol{\eta} - \boldsymbol{\mu}_{\boldsymbol{\eta}}^{(t)}, 1, 0) \\
&+ (\ln \rho - \ln \mu_\rho^{(t)}) \mu_\rho^{(t)} \boldsymbol{A}(\boldsymbol{\mu}_{\boldsymbol{\eta}}^{(t)}, 1, 0) + (\ln \kappa - \ln \mu_\kappa^{(t)}) \mu_\kappa^{(t)} \boldsymbol{I} \Big] \Bigg).
\end{aligned} \tag{4.28}$$

We also use a similar approximation around $[\boldsymbol{\eta}, \ln \rho, \ln \kappa] = [\boldsymbol{\mu}_{\boldsymbol{\eta}}^{(t+1)}, \ln \mu_\rho^{(t)}, \ln \mu_\kappa^{(t)}]$. In addition, $\ln \mathrm{Sigmoid}(\lambda)$ in Eq. (4.12) is approximated around $\ln \lambda = \ln \mu_\lambda^{(t)}$,

$$\begin{aligned}
\ln\ \mathrm{Sigmoid}(\lambda) &\approx \ln\ \mathrm{Sigmoid}(\mu_\lambda^{(t)}) \\
&+ (\ln \lambda - \ln \mu_\lambda^{(t)}) \mu_\lambda^{(t)}\ \mathrm{Sigmoid}(-\mu_\lambda^{(t)}).
\end{aligned} \tag{4.29}$$

For the "compound" Gaussian MRF prior in Eq. (4.17), $\ln \sum_{\boldsymbol{\eta}} \exp \Big\{ -\lambda \sum_{i \sim j} (1 - \eta_{i,j}) - \frac{1}{2} \ln \Big| \frac{1}{2\pi} \boldsymbol{A}(\boldsymbol{\eta}, \rho, \kappa) \Big| \Big\}$ is approximated around $\ln \lambda = \ln \mu_\lambda^{(t)}$,

$$\begin{aligned}
&\ln \sum_{\boldsymbol{\eta}} \exp \left\{ -\lambda \sum_{i \sim j} (1 - \eta_{i,j}) - \frac{1}{2} \ln \left| \frac{1}{2\pi} \boldsymbol{A}(\boldsymbol{\eta}, \rho, \kappa) \right| \right\} \\
&\approx \ln \sum_{\boldsymbol{\eta}} \exp \left\{ -\mu_\lambda^{(t)} \sum_{i \sim j} (1 - \eta_{i,j}) - \frac{1}{2} \ln \left| \frac{1}{2\pi} \boldsymbol{A}(\boldsymbol{\eta}, \rho, \kappa) \right| \right\} \\
&- (\ln \lambda - \ln \mu_\lambda^{(t)}) \mu_\lambda^{(t)} \frac{\sum_{\boldsymbol{\eta}} \left\{ \sum_{i \sim j} (1 - \eta_{i,j}) \right\} \exp \left\{ -\mu_\lambda^{(t)} \sum_{i \sim j} (1 - \eta_{i,j}) - \frac{1}{2} \ln \left| \frac{1}{2\pi} \boldsymbol{A}(\boldsymbol{\eta}, \rho, \kappa) \right| \right\}}{\sum_{\boldsymbol{\eta}} \exp \left\{ -\mu_\lambda^{(t)} \sum_{i \sim j} (1 - \eta_{i,j}) - \frac{1}{2} \ln \left| \frac{1}{2\pi} \boldsymbol{A}(\boldsymbol{\eta}, \rho, \kappa) \right| \right\}}.
\end{aligned} \tag{4.30}$$

Also, $\ln|\boldsymbol{A}(\boldsymbol{\eta},\rho,\kappa)|$ in Eq. (4.17) is approximated around $[\boldsymbol{\eta},\ln\rho,\ln\kappa] = [\boldsymbol{\mu}_{\boldsymbol{\eta}}^{(t)},\ln\mu_{\rho}^{(t)},\ln\mu_{\kappa}^{(t)}]$

$$\begin{aligned}&\ln|\boldsymbol{A}(\boldsymbol{\eta},\rho,\kappa)| \qquad (4.31)\\ &\approx \ln\left|\boldsymbol{A}(\boldsymbol{\mu}_{\boldsymbol{\eta}}^{(t)},\mu_{\rho}^{(t)},\mu_{\kappa}^{(t)})\right| + \mathrm{tr}\left(\boldsymbol{A}(\boldsymbol{\mu}_{\boldsymbol{\eta}}^{(t)},\mu_{\rho}^{(t)},\mu_{\kappa}^{(t)})^{-1}\Big[\mu_{\rho}^{(t)}\boldsymbol{A}(\boldsymbol{\eta}-\boldsymbol{\mu}_{\boldsymbol{\eta}}^{(t)},1,0)\right.\\ &\left.+(\ln\rho-\ln\mu_{\rho}^{(t)})\mu_{\rho}^{(t)}\boldsymbol{A}(\boldsymbol{\mu}_{\boldsymbol{\eta}}^{(t)},1,0)+(\ln\kappa-\ln\mu_{\kappa}^{(t)})\mu_{\kappa}^{(t)}\boldsymbol{I}\Big]\right).\end{aligned}$$

This approximation enables us to solve the exponential-order calculation cost problem of the HR image priors. It makes it possible to calculate the normalization term of them.

4.6.3 Update Equations for Algorithm 1

We derive Algorithm 1 as the iterative updating equations based on VB using "causal" Gaussian MRF prior.

First, we define the parameters for the prior distributions of the hyperparameters to be as non-informative as possible since generally, prior distributions should be non-informative unless we have explicit reasons because an informative prior leads to heuristics:

$$\begin{aligned}&a_{\lambda}^{(0)}\equiv 10^{-2}, b_{\lambda}^{(0)}\equiv 10^{-2},\ a_{\rho}^{(0)}\equiv 10^{-2}, b_{\rho}^{(0)}\equiv 10^{-2},\\ &a_{\kappa}^{(0)}\equiv 10^{-2}, b_{\kappa}^{(0)}\equiv 10^{-2},\ a_{\beta}^{(0)}\equiv 10^{-2}, b_{\beta}^{(0)}\equiv 10^{-2}. \qquad (4.32)\end{aligned}$$

For a gamma distribution, the number of effective prior observations in the Bayesian framework is equal to two times parameter a. The number of observations for the hyperparameter λ is $N_{\boldsymbol{\eta}}$ in this SR. Also, that for ρ and κ is $N_{\boldsymbol{x}}$, and that for β is $LN_{\boldsymbol{y}}$. Therefore, the above settings – e.g., $2a_{\lambda}^{(0)} \ll N_{\boldsymbol{\eta}}$ – are considered sufficiently non-informative.

The trial distributions are obtained from Eqs. (4.6), (4.12), (4.18), (4.19), and (4.23)-(4.29) as follows:

$$q^{(t)}(\boldsymbol{\eta}) = \prod_{i\sim j}\mathrm{Bernoulli}(\eta_{i,j}|\mu_{\eta_{i,j}}^{(t)}), \qquad (4.33)$$

$$q^{(t)}(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_{\boldsymbol{x}}^{(t)},\boldsymbol{\Sigma}_{\boldsymbol{x}}^{(t)}), \qquad (4.34)$$

$$\begin{aligned}q^{(t)}(\lambda,\rho,\kappa,\beta) = &\,\mathrm{Gamma}(\lambda|a_{\lambda}^{(t)},b_{\lambda}^{(t)})\,\mathrm{Gamma}(\rho|a_{\rho}^{(t)},b_{\rho}^{(t)})\\ &\times\mathrm{Gamma}(\kappa|a_{\kappa}^{(t)},b_{\kappa}^{(t)})\,\mathrm{Gamma}(\beta|a_{\beta}^{(t)},b_{\beta}^{(t)}), \qquad (4.35)\end{aligned}$$

$$q^{(t)}(\boldsymbol{\Phi}) = \prod_{l=1}^{L}\mathcal{N}(\boldsymbol{\phi}_l|\boldsymbol{\mu}_{\boldsymbol{\phi}_l}^{(t)},\boldsymbol{\Sigma}_{\boldsymbol{\phi}_l}^{(t)}). \qquad (4.36)$$

Using the mean values of the hyperparameters $\lambda, \rho, \kappa, \beta$ over the trial distributions $q^{(t)}(\lambda, \rho, \kappa, \beta)$, $\mu_\lambda^{(t)} = \frac{a_\lambda^{(t)}}{b_\lambda^{(t)}}$, $\mu_\rho^{(t)} = \frac{a_\rho^{(t)}}{b_\rho^{(t)}}$, $\mu_\kappa^{(t)} = \frac{a_\kappa^{(t)}}{b_\kappa^{(t)}}$, $\mu_\beta^{(t)} = \frac{a_\beta^{(t)}}{b_\beta^{(t)}}$, we can analytically compute the parameters at step $t+1$ in Eqs. (4.33) - (4.36):

$$\mu_{\eta_{i,j}}^{(t+1)} = \text{Sigmoid}\left(\mu_\lambda^{(t)} + \frac{1}{2}\mu_\rho^{(t)} C_{\eta_{i,j}}^{(t)}\right), \tag{4.37}$$

where

$$C_{\eta_{i,j}}^{(t)} \equiv \text{tr}\left(\left(\boldsymbol{A}(\boldsymbol{\mu}_{\boldsymbol{\eta}}^{(t)}, \mu_\rho^{(t)}, \mu_\kappa^{(t)})^{-1} - \boldsymbol{C}_{\boldsymbol{x}}^{(t)}\right)\boldsymbol{M}_{i,j}\right), \tag{4.38}$$

$$\boldsymbol{C}_{\boldsymbol{x}}^{(t)} \equiv \boldsymbol{\mu}_{\boldsymbol{x}}^{(t)}[\boldsymbol{\mu}_{\boldsymbol{x}}^{(t)}]^\top + \boldsymbol{\Sigma}_{\boldsymbol{x}}^{(t)}, \tag{4.39}$$

$$[\boldsymbol{M}_{i,j}]_{k,l} \equiv \begin{cases} +1, & (k,l) = (i,i) \text{ or } (j,j), \\ -1, & (k,l) = (i,j) \text{ or } (j,i), \\ 0, & \text{otherwise.} \end{cases} \tag{4.40}$$

$$\boldsymbol{\mu}_{\boldsymbol{x}}^{(t+1)} = \boldsymbol{\Sigma}_{\boldsymbol{x}}^{(t+1)}\left[\mu_\beta^{(t)} \sum_{l=1}^{L} \boldsymbol{y}_l^\top \boldsymbol{W}_l^{(t)}\right]^\top, \tag{4.41}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{x}}^{(t+1)} = \left[\boldsymbol{A}(\boldsymbol{\mu}_{\boldsymbol{\eta}}^{(t+1)}, \mu_\rho^{(t)}, \mu_\kappa^{(t)}) + \mu_\beta^{(t)} \sum_{l=1}^{L} \boldsymbol{C}_{\boldsymbol{W}_l}^{\prime(t)}\right]^{-1}, \tag{4.42}$$

where

$$\boldsymbol{C}_{\boldsymbol{W}_l}^{\prime(t)} \equiv [\boldsymbol{W}_l^{(t)}]^\top \boldsymbol{W}_l^{(t)} + \sum_{k,k'} [\boldsymbol{\Sigma}_{\boldsymbol{\phi}_l}^{(t)}]_{k,k'} [\boldsymbol{W}_{l,k}^{\prime(t)}]^\top \boldsymbol{W}_{l,k'}^{\prime(t)}. \tag{4.43}$$

$$a_\lambda^{(t+1)} = a_\lambda^{(0)} + N_{\boldsymbol{\eta}} \mu_\lambda^{(t)} \text{Sigmoid}(-\mu_\lambda^{(t)}), \tag{4.44}$$

$$b_\lambda^{(t+1)} = b_\lambda^{(0)} + \sum_{i \sim j} (1 - \mu_{\eta_{i,j}}^{(t+1)}), \tag{4.45}$$

$$a_\rho^{(t+1)} = a_\rho^{(0)} + \frac{\mu_\rho^{(t)}}{2} \text{tr}\left(\boldsymbol{A}(\boldsymbol{\mu}_{\boldsymbol{\eta}}^{(t+1)}, \mu_\rho^{(t)}, \mu_\kappa^{(t)})^{-1} \boldsymbol{A}(\boldsymbol{\mu}_{\boldsymbol{\eta}}^{(t+1)}, 1, 0)\right) \tag{4.46}$$

$$b_\rho^{(t+1)} = b_\rho^{(0)} + \frac{1}{2} \text{tr}\left(\boldsymbol{C}_{\boldsymbol{x}}^{(t+1)} \boldsymbol{A}(\boldsymbol{\mu}_{\boldsymbol{\eta}}^{(t+1)}, 1, 0)\right), \tag{4.47}$$

$$a_\kappa^{(t+1)} = a_\kappa^{(0)} + \frac{\mu_\kappa^{(t)}}{2} \text{tr}\left(\boldsymbol{A}(\boldsymbol{\mu}_{\boldsymbol{\eta}}^{(t+1)}, \mu_\rho^{(t)}, \mu_\kappa^{(t)})^{-1}\right) \tag{4.48}$$

$$b_\kappa^{(t+1)} = b_\kappa^{(0)} + \frac{1}{2} \text{tr}\left(\boldsymbol{C}_{\boldsymbol{x}}^{(t+1)}\right), \tag{4.49}$$

$$a_\beta^{(t+1)} = a_\beta^{(0)} + \frac{1}{2} L N_{\boldsymbol{y}}, \tag{4.50}$$

$$b_\beta^{(t+1)} = b_\beta^{(0)} + \frac{1}{2} \sum_{l=1}^{L} \left(\text{tr}\left(\boldsymbol{C}_{\boldsymbol{x}}^{(t+1)} \boldsymbol{C}_{\boldsymbol{W}_l}^{\prime(t)}\right) - 2\boldsymbol{y}_l^\top \boldsymbol{W}_l^{(t)} \boldsymbol{\mu}_{\boldsymbol{x}}^{(t+1)} + \boldsymbol{y}_l^\top \boldsymbol{y}_l\right). \tag{4.51}$$

$$\boldsymbol{\mu}_{\boldsymbol{\phi}_l}^{(t+1)} = \boldsymbol{\Sigma}_{\boldsymbol{\phi}_l}^{(t+1)}\left[[\boldsymbol{\Sigma}_{\boldsymbol{\phi}_l}^{(0)}]^{-1} \boldsymbol{\mu}_{\boldsymbol{\phi}_l}^{(0)} + \mu_\beta^{(t)} [\boldsymbol{C}_{\boldsymbol{\phi}_l}^{\prime\prime(t+1)} \boldsymbol{\mu}_{\boldsymbol{\phi}_l}^{(t)} - \boldsymbol{C}_{\boldsymbol{\phi}_l}^{\prime(t+1)}]\right], \tag{4.52}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{\phi}_l}^{(t+1)} = \left[[\boldsymbol{\Sigma}_{\boldsymbol{\phi}_l}^{(0)}]^{-1} + \mu_\beta^{(t)} \boldsymbol{C}_{\boldsymbol{\phi}_l}^{\prime\prime(t+1)}\right]^{-1}, \tag{4.53}$$

where

$$[\boldsymbol{C}'^{(t+1)}_{\boldsymbol{\phi}_l}]_k \equiv \frac{1}{2}\,\mathrm{tr}\left(\boldsymbol{C}^{(t+1)}_{\boldsymbol{x}}\left[[\boldsymbol{W}^{(t)}_l]^\top \boldsymbol{W}'^{(t)}_{l,k} + [\boldsymbol{W}'^{(t)}_{l,k}]^\top \boldsymbol{W}^{(t)}_l\right]\right) - \boldsymbol{y}_l^\top \boldsymbol{W}'^{(t)}_{l,k}\boldsymbol{\mu}^{(t+1)}_{\boldsymbol{x}}, \tag{4.54}$$

$$[\boldsymbol{C}''^{(t+1)}_{\boldsymbol{\phi}_l}]_{k,k'} \equiv \mathrm{tr}\left(\boldsymbol{C}^{(t+1)}_{\boldsymbol{x}}[\boldsymbol{W}'^{(t)}_{l,k}]^\top \boldsymbol{W}'^{(t)}_{l,k'}\right). \tag{4.55}$$

For (4.23) and (4.24), we update those distributions as follows. First, we compute $q^{(t+1)}(\boldsymbol{\eta})$ using $q^{(t)}(\boldsymbol{x}, \lambda, \rho, \kappa, \beta, \boldsymbol{\Phi})$. Second, we compute $q^{(t+1)}(\boldsymbol{x})$ using $q^{(t+1)}(\boldsymbol{\eta})q^{(t)}(\lambda, \rho, \kappa, \beta, \boldsymbol{\Phi})$. Finally, we compute $q^{(t+1)}(\lambda, \rho, \kappa, \beta)$ using $q^{(t+1)}(\boldsymbol{x}, \boldsymbol{\eta})q^{(t)}(\boldsymbol{\Phi})$ and $q^{(t+1)}(\boldsymbol{\Phi})$ using $q^{(t+1)}(\boldsymbol{x}, \boldsymbol{\eta})q^{(t)}(\lambda, \rho, \kappa, \beta)$. Here, we simply compute only the parameters of those distributions because we can compute the expectations in Eq. (4.24) analytically by using Taylor approximations in Eqs. (4.25) - (4.29).

For the initial parameters of the trial distributions of $\boldsymbol{\eta}$ and $\boldsymbol{x}$, we use non-informative values,

$$\boldsymbol{\mu}^{(0)}_{\boldsymbol{\eta}} \equiv \mathbf{0},\ \boldsymbol{\mu}^{(0)}_{\boldsymbol{x}} \equiv \mathbf{0},\ \boldsymbol{\Sigma}^{(0)}_{\boldsymbol{x}} \equiv \mathbf{0}. \tag{4.56}$$

For the initial parameters for λ, ρ, κ, β and $\boldsymbol{\Phi}$, we use the same values as their prior's values.

We obtain the well-approximated PM of $\hat{\boldsymbol{x}}^*$ as $\boldsymbol{\mu}^{(\infty)}_{\boldsymbol{x}}$. Realistically, instead of $\boldsymbol{\mu}^{(\infty)}_{\boldsymbol{x}}$, we use $\boldsymbol{\mu}^{(t+1)}_{\boldsymbol{x}}$ when the following convergence conditions hold for $\boldsymbol{\mu}^{(t+1)}_{\boldsymbol{x}}$ and each $\mu^{(t+1)}_{\phi_{l,k}}$,

$$\begin{aligned} &\frac{1}{N_{\boldsymbol{x}}}\|\boldsymbol{\mu}^{(t+1)}_{\boldsymbol{x}} - \boldsymbol{\mu}^{(t)}_{\boldsymbol{x}}\|_2^2 < 10^{-4}, \\ &\frac{1}{L}\sum_{l=1}^{L}\frac{(\mu^{(t+1)}_{\phi_{l,k}} - \mu^{(t)}_{\phi_{l,k}})^2}{[\boldsymbol{\sigma}^2_{\boldsymbol{\phi}}]_k} < 10^{-4} \quad (k = 1, 2, 3, 4), \end{aligned} \tag{4.57}$$

where we defined $\boldsymbol{\sigma}^2_{\boldsymbol{\phi}} \equiv [10^{-3}, 10^0, 10^0, 10^{-3}]$ as the scaling constant.

4.6.4 Update Equations for Algorithm 2

We derive Algorithm 2 as the iterative updating equations based on VB using "compound" Gaussian MRF prior.

First, we define the parameters for the prior distributions of the hyperparameters to be as non-informative as possible:

$$a^{(0)}_{\lambda} \equiv 3 \times 10^{-2}, \tag{4.58}$$

$$b^{(0)}_{\lambda}, a^{(0)}_{\rho}, b^{(0)}_{\rho}, a^{(0)}_{\kappa}, b^{(0)}_{\kappa}, a^{(0)}_{\beta}, b^{(0)}_{\beta} \equiv 10^{-2} \tag{4.59}$$

From Eqs. (4.6), (4.17), (4.18), (4.19), (4.23)-(4.25), (4.30) and (4.31), the trial

distributions are obtained as the following distributions:

$$q^{(t)}(\boldsymbol{\eta}) = \prod_{i\sim j} \text{Bernoulli}(\eta_{i,j}|\mu_{\eta_{i,j}}^{(t)}), \tag{4.60}$$

$$q^{(t)}(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_{\boldsymbol{x}}^{(t)}, \boldsymbol{\Sigma}_{\boldsymbol{x}}^{(t)}), \tag{4.61}$$

$$\begin{aligned} q^{(t)}(\lambda,\rho,\kappa,\beta) = &\text{Gamma}(\lambda|a_\lambda^{(t)}, b_\lambda^{(t)})\,\text{Gamma}(\rho|a_\rho^{(t)}, b_\rho^{(t)}) \\ &\times \text{Gamma}(\kappa|a_\kappa^{(t)}, b_\kappa^{(t)})\,\text{Gamma}(\beta|a_\beta^{(t)}, b_\beta^{(t)}), \end{aligned} \tag{4.62}$$

$$q^{(t)}(\boldsymbol{\Phi}) = \prod_{l=1}^{L} \mathcal{N}(\boldsymbol{\phi}_l|\boldsymbol{\mu}_{\boldsymbol{\phi}_l}^{(t)}, \boldsymbol{\Sigma}_{\boldsymbol{\phi}_l}^{(t)}). \tag{4.63}$$

Using the mean values of the hyperparameters $\lambda, \rho, \kappa, \beta$ over the trial distributions $q^{(t)}(\lambda,\rho,\kappa,\beta)$, $\mu_\lambda^{(t)} = \frac{a_\lambda^{(t)}}{b_\lambda^{(t)}}$, $\mu_\rho^{(t)} = \frac{a_\rho^{(t)}}{b_\rho^{(t)}}$, $\mu_\kappa^{(t)} = \frac{a_\kappa^{(t)}}{b_\kappa^{(t)}}$, $\mu_\beta^{(t)} = \frac{a_\beta^{(t)}}{b_\beta^{(t)}}$, we can analytically compute the parameters at step $t+1$ in Eqs. (4.60) - (4.63)

$$\mu_{\eta_{i,j}}^{(t+1)} = \text{Sigmoid}\left(\mu_\lambda^{(t)} - \frac{1}{2}\mu_\rho^{(t)}\,\text{tr}\left(\boldsymbol{C}_{\boldsymbol{x}}^{(t)}\boldsymbol{M}_{i,j}\right)\right), \tag{4.64}$$

where

$$\boldsymbol{C}_{\boldsymbol{x}}^{(t)} \equiv \boldsymbol{\mu}_{\boldsymbol{x}}^{(t)}[\boldsymbol{\mu}_{\boldsymbol{x}}^{(t)}]^\top + \boldsymbol{\Sigma}_{\boldsymbol{x}}^{(t)}, \tag{4.65}$$

$$[\boldsymbol{M}_{i,j}]_{k,l} \equiv \begin{cases} +1, & (k,l) = (i,i) \text{ or } (j,j), \\ -1, & (k,l) = (i,j) \text{ or } (j,i), \\ 0, & \text{otherwise.} \end{cases} \tag{4.66}$$

$$\boldsymbol{\mu}_{\boldsymbol{x}}^{(t+1)} = \boldsymbol{\Sigma}_{\boldsymbol{x}}^{(t+1)}\left[\mu_\beta^{(t)}\sum_{l=1}^{L}\boldsymbol{y}_l^\top\boldsymbol{W}_l^{(t)}\right]^\top, \tag{4.67}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{x}}^{(t+1)} = \left[\boldsymbol{A}(\boldsymbol{\mu}_{\boldsymbol{\eta}}^{(t+1)}, \mu_\rho^{(t)}, \mu_\kappa^{(t)}) + \mu_\beta^{(t)}\sum_{l=1}^{L}\boldsymbol{C}_{\boldsymbol{W}_l}^{\prime(t)}\right]^{-1}, \tag{4.68}$$

where

$$\boldsymbol{C}_{\boldsymbol{W}_l}^{\prime(t)} \equiv [\boldsymbol{W}_l^{(t)}]^\top\boldsymbol{W}_l^{(t)} + \sum_{k,k'}[\boldsymbol{\Sigma}_{\boldsymbol{\phi}_l}^{(t)}]_{k,k'}[\boldsymbol{W}_{l,k}^{\prime(t)}]^\top\boldsymbol{W}_{l,k'}^{\prime(t)}. \tag{4.69}$$

$$a_\lambda^{(t+1)} = a_\lambda^{(0)} + \mu_\lambda^{(t)} \sum_{i \sim j} (1 - \hat{\eta}_{i,j}^{(t+1)}), \tag{4.70}$$

$$b_\lambda^{(t+1)} = b_\lambda^{(0)} + \sum_{i \sim j} (1 - \mu_{\eta_{i,j}}^{(t+1)}), \tag{4.71}$$

$$a_\rho^{(t+1)} = a_\rho^{(0)} + \frac{\mu_\rho^{(t)}}{2} \mathrm{tr}\left(\boldsymbol{A}(\boldsymbol{\mu}_{\boldsymbol{\eta}}^{(t+1)}, \mu_\rho^{(t)}, \mu_\kappa^{(t)})^{-1} \boldsymbol{A}(\boldsymbol{\mu}_{\boldsymbol{\eta}}^{(t+1)}, 1, 0) \right) \tag{4.72}$$

$$b_\rho^{(t+1)} = b_\rho^{(0)} + \frac{1}{2} \mathrm{tr}\left(\boldsymbol{C}_{\boldsymbol{x}}^{(t+1)} \boldsymbol{A}(\boldsymbol{\mu}_{\boldsymbol{\eta}}^{(t+1)}, 1, 0) \right), \tag{4.73}$$

$$a_\kappa^{(t+1)} = a_\kappa^{(0)} + \frac{\mu_\kappa^{(t)}}{2} \mathrm{tr}\left(\boldsymbol{A}(\boldsymbol{\mu}_{\boldsymbol{\eta}}^{(t+1)}, \mu_\rho^{(t)}, \mu_\kappa^{(t)})^{-1} \right) \tag{4.74}$$

$$b_\kappa^{(t+1)} = b_\kappa^{(0)} + \frac{1}{2} \mathrm{tr}\left(\boldsymbol{C}_{\boldsymbol{x}}^{(t+1)} \right), \tag{4.75}$$

$$a_\beta^{(t+1)} = a_\beta^{(0)} + \frac{1}{2} L N_{\boldsymbol{y}}, \tag{4.76}$$

$$b_\beta^{(t+1)} = b_\beta^{(0)} + \frac{1}{2} \sum_{l=1}^{L} \left(\mathrm{tr}\left(\boldsymbol{C}_{\boldsymbol{x}}^{(t+1)} \boldsymbol{C}_{\boldsymbol{W}_l}^{\prime(t)} \right) - 2 \boldsymbol{y}_l^\top \boldsymbol{W}_l^{(t)} \boldsymbol{\mu}_{\boldsymbol{x}}^{(t+1)} + \boldsymbol{y}_l^\top \boldsymbol{y}_l \right), \tag{4.77}$$

where

$$\hat{\eta}_{i,j}^{(t+1)} \equiv \mathrm{Sigmoid}\left(\mu_\lambda^{(t)} - \left(\left[\boldsymbol{A}(\boldsymbol{\mu}_{\boldsymbol{\eta}}^{(t)}, \mu_\rho^{(t)}, \mu_\kappa^{(t)})^{-1} \right]_{i,j} - \left[\boldsymbol{A}(\boldsymbol{\mu}_{\boldsymbol{\eta}}^{(t)}, \mu_\rho^{(t)}, \mu_\kappa^{(t)})^{-1} \right]_{i,i} \right) \right), \tag{4.78}$$

$$\boldsymbol{\mu}_{\boldsymbol{\phi}_l}^{(t+1)} = \boldsymbol{\Sigma}_{\boldsymbol{\phi}_l}^{(t+1)} \left[[\boldsymbol{\Sigma}_{\boldsymbol{\phi}_l}^{(0)}]^{-1} \boldsymbol{\mu}_{\boldsymbol{\phi}_l}^{(0)} + \mu_\beta^{(t)} [\boldsymbol{C}_{\boldsymbol{\phi}_l}^{\prime\prime(t+1)} \boldsymbol{\mu}_{\boldsymbol{\phi}_l}^{(t)} - \boldsymbol{C}_{\boldsymbol{\phi}_l}^{\prime(t+1)}] \right], \tag{4.79}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{\phi}_l}^{(t+1)} = \left[[\boldsymbol{\Sigma}_{\boldsymbol{\phi}_l}^{(0)}]^{-1} + \mu_\beta^{(t)} \boldsymbol{C}_{\boldsymbol{\phi}_l}^{\prime\prime(t+1)} \right]^{-1}, \tag{4.80}$$

where

$$[\boldsymbol{C}_{\boldsymbol{\phi}_l}^{\prime(t+1)}]_k \equiv \frac{1}{2} \mathrm{tr}\left(\boldsymbol{C}_{\boldsymbol{x}}^{(t+1)} \left[[\boldsymbol{W}_l^{(t)}]^\top \boldsymbol{W}_{l,k}^{\prime(t)} + [\boldsymbol{W}_{l,k}^{\prime(t)}]^\top \boldsymbol{W}_l^{(t)} \right] \right) - \boldsymbol{y}_l^\top \boldsymbol{W}_{l,k}^{\prime(t)} \boldsymbol{\mu}_{\boldsymbol{x}}^{(t+1)}, \tag{4.81}$$

$$[\boldsymbol{C}_{\boldsymbol{\phi}_l}^{\prime\prime(t+1)}]_{k,k'} \equiv \mathrm{tr}\left(\boldsymbol{C}_{\boldsymbol{x}}^{(t+1)} [\boldsymbol{W}_{l,k}^{\prime(t)}]^\top \boldsymbol{W}_{l,k'}^{\prime(t)} \right). \tag{4.82}$$

For (4.23) and (4.24), we update those distributions as follows. First, we compute $q^{(t+1)}(\boldsymbol{\eta})$ using $q^{(t)}(\boldsymbol{x}, \lambda, \rho, \kappa, \beta, \boldsymbol{\Phi})$. Second, we compute $q^{(t+1)}(\boldsymbol{x})$ using $q^{(t+1)}(\boldsymbol{\eta}) q^{(t)}(\lambda, \rho, \kappa, \beta, \boldsymbol{\Phi})$. Finally, we compute $q^{(t+1)}(\lambda, \rho, \kappa, \beta)$ using $q^{(t+1)}(\boldsymbol{x}, \boldsymbol{\eta}) q^{(t)}(\boldsymbol{\Phi})$ and $q^{(t+1)}(\boldsymbol{\Phi})$ using $q^{(t+1)}(\boldsymbol{x}, \boldsymbol{\eta}) q^{(t)}(\lambda, \rho, \kappa, \beta)$. For the initial parameters of the trial distributions of $\boldsymbol{\eta}$ and $\boldsymbol{x}$, we use non-informative values, $\boldsymbol{\mu}_{\boldsymbol{\eta}}^{(0)} \equiv \boldsymbol{0}$, $\boldsymbol{\mu}_{\boldsymbol{x}}^{(0)} \equiv \boldsymbol{0}$, $\boldsymbol{\Sigma}_{\boldsymbol{x}}^{(0)} \equiv \boldsymbol{0}$. As the initial parameters for λ, ρ, β, κ, and $\boldsymbol{\Phi}$ we use the same values as their prior's values in Eq. (4.18), (4.19). Here, we simply compute only the parameters of those distributions because we can compute the expectations in Eq. (4.24) analytically by using Taylor approximations in Eqs. (4.25), (4.30) and (4.31).

Table 4.1 PSNR of the proposed method (a higher value is better) and ISNRs against three previous methods (a higher value is better) for different images and SNR levels

Image	SNR	PSNR	ISNR			
	[dB]	(Algorithm 2)	(vs Bilinear)	(vs Kanemura)	(vs Babacan)	(vs Algorithm 1)
Lena	20	29.31 ± 0.30	5.45 ± 0.33	0.67 ± 0.34	0.02 ± 0.11	0.05 ± 0.01
	30	32.15 ± 0.36	8.20 ± 0.37	1.74 ± 0.34	0.52 ± 0.18	0.10 ± 0.20
	40	34.19 ± 0.60	10.24 ± 0.60	3.21 ± 0.53	0.95 ± 0.60	1.49 ± 0.77
Cameraman	20	21.76 ± 0.20	4.13 ± 0.21	0.95 ± 0.32	-0.04 ± 0.08	-0.01 ± 0.01
	30	23.59 ± 0.28	5.92 ± 0.28	1.56 ± 0.32	-0.01 ± 0.11	-0.06 ± 0.02
	40	25.04 ± 0.41	7.37 ± 0.42	2.70 ± 0.30	0.32 ± 0.27	-0.01 ± 0.14
Pepper	20	29.73 ± 0.24	3.68 ± 0.26	0.09 ± 0.41	0.23 ± 0.10	0.11 ± 0.87
	30	31.65 ± 0.33	5.51 ± 0.33	0.76 ± 0.48	0.11 ± 0.22	0.35 ± 0.33
	40	32.23 ± 0.51	6.09 ± 0.51	1.11 ± 0.45	-0.17 ± 0.48	0.93 ± 0.56
Clock	20	23.29 ± 0.28	5.38 ± 0.29	1.40 ± 0.23	0.10 ± 0.09	0.01 ± 0.01
	30	25.42 ± 0.29	7.46 ± 0.29	2.59 ± 0.30	0.29 ± 0.13	-0.03 ± 0.01
	40	27.08 ± 0.38	9.13 ± 0.38	4.00 ± 0.31	0.74 ± 0.32	-0.07 ± 0.12
Text	20	24.68 ± 0.32	5.83 ± 0.33	1.65 ± 0.29	-0.06 ± 0.09	0.02 ± 0.02
	30	27.27 ± 0.43	8.38 ± 0.44	3.09 ± 0.41	0.19 ± 0.18	-0.03 ± 0.04
	40	29.28 ± 0.62	10.39 ± 0.62	4.85 ± 0.51	0.78 ± 0.51	1.98 ± 0.69

We obtain the well approximated PM of $\boldsymbol{x}$ as $\boldsymbol{\mu}_{\boldsymbol{x}}^{(t+1)}$, for which the following convergence conditions hold for $\boldsymbol{\mu}_{\boldsymbol{x}}^{(t+1)}$ and each $\mu_{\phi_{l,k}}^{(t+1)}$,

$$\begin{aligned} &\frac{1}{N_{\boldsymbol{x}}}\|\boldsymbol{\mu}_{\boldsymbol{x}}^{(t+1)} - \boldsymbol{\mu}_{\boldsymbol{x}}^{(t)}\|_2^2 < 10^{-5}, \\ &\frac{1}{L}\sum_{l=1}^{L}\frac{(\mu_{\phi_{l,k}}^{(t+1)} - \mu_{\phi_{l,k}}^{(t)})^2}{[\boldsymbol{\sigma}_{\boldsymbol{\phi}}^2]_k} < 10^{-5} \quad (k = 1, 2, 3, 4), \end{aligned} \tag{4.83}$$

where we defined $\boldsymbol{\sigma}_{\boldsymbol{\phi}}^2 \equiv [10^{-3}, 10^0, 10^0, 10^{-3}]$ as the scaling constant.

4.7 Experimental Results

The proposed method was evaluated using five gray-scale images with a size of 40×40 pixels, as shown in Fig. 4.2. From each image, $L = 10$ images with a size of 10×10 pixels were created by using (4.5), (4.6) with the settings of the parameters α, $\boldsymbol{\Phi}$, and β as the following. The resolution enhancement factor α was 4. The transformation parameter $\boldsymbol{\Phi}$ was randomly created according to the prior distribution in (4.19), where it is similar to that in previous work [141–145, 147]. The noise level parameter β was set for signal-to-noise ratios (SNR) of 20, 30, and 40 dB for each image. Samples of the created images are shown in Fig. 4.3.

Figure 4.4 shows the images estimated under SNR= 30dB by the proposed method. The resolution of each image appeared to be better than the corresponding observed image in Fig. 4.3.

Table 4.1 lists the quantitative results compared to those from the methods of bilinear interpolation, Kanemura *et al.* [143] which is the variational EM approach with a causal Gaussian MRF prior using the MAP estimation function, and Babacan *et al.* [142] which is the VB approach with a TV prior using the MAP estimation function. Note that we added a slight modification to these methods because they employ slightly different models. For example, the original method [142] assumes the blurring parameter γ is known, so we set γ as the mean value of the true distribution for this method. Also, we introduced a strong prior for λ in the Kanemura method [143] in contrast to the original method, because this parameter sometimes becomes negative. We evaluated the results with regard to the expectation and the standard deviation of the improvement in signal-to-noise ratio (ISNR) over 10 experiments on each image and for each SNR. ISNR is the relative PSNR defined as

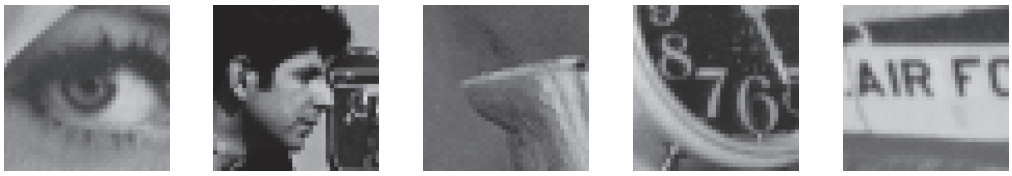
$$\mathrm{ISNR} \equiv \mathrm{PSNR}(\hat{\boldsymbol{x}}^*(\boldsymbol{Y})\|\boldsymbol{x}) - \mathrm{PSNR}(\tilde{\boldsymbol{x}}\|\boldsymbol{x}), \quad \text{where} \tag{4.84}$$

$$\mathrm{PSNR}(\hat{\boldsymbol{x}}^*(\boldsymbol{Y})\|\boldsymbol{x}) \equiv 10\log_{10}\frac{2^2}{\frac{1}{N_{\boldsymbol{x}}}\|\hat{\boldsymbol{x}}^*(\boldsymbol{Y}) - \boldsymbol{x}\|_2^2}, \tag{4.85}$$

where $\boldsymbol{x}$ is the true HR image, $\hat{\boldsymbol{x}}^*(\boldsymbol{Y})$ is the image estimated by the proposed method, and $\tilde{\boldsymbol{x}}$ is the image estimated by the compared method. A higher ISNR value means better improvement of the estimate against the estimate of the compared method. We see that the ISNRs of the images estimated by the proposed method were mostly better than those by the other methods. In the subjective visual comparison in Fig. 4.5, we also see that the edges are not overemphasized in the images estimated by the proposed method compared to those in the images estimated by the other methods. Regarding the estimation function, we used the optimal estimation function, the PM. From the experimental results, we see that the SR methods with the PM estimation function (i.e., the proposed methods) were more accurate than the SR methods with other estimation functions (i.e., MAP by Kanemura *et al.* and Babacan *et al.*). This indicates that PM is an optimal estimation function for PSNR based on mean square error.

Table 4.2 lists the root mean square errors (RMSE) of registration parameters estimated by our method and the other methods. To evaluate the estimated registration parameters, we took the RMSEs over 50 experiments (10 experiments $\times$ 5 images) for each noise level. Of course, a lower RMSE value means a better estimate. We see that the RMSEs of the proposed method were mostly higher than those of the other methods.

The calculation times with the proposed method and with the other methods for each estimate, using an Intel Core i7 2600 processor, were almost the same, about 10 minutes.



(a) Lena (b) Cameraman (c) Pepper (d) Clock (e) Text

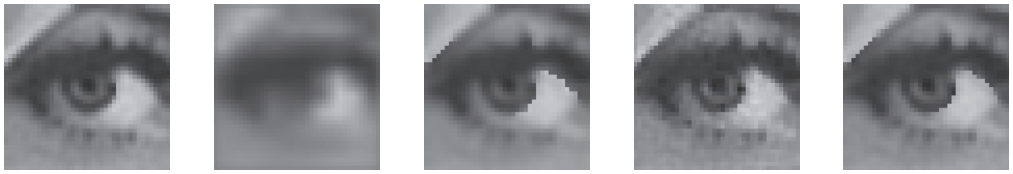
Fig. 4.2 Five original images used in the experiments

(a) Lena (b) Cameraman (c) Pepper (d) Clock (e) Text

Fig. 4.3 Observed images when warped, blurred, downsampled by an enhancement factor of 4, and noised with SNR= 30dB AWGN

(a) Lena (b) Cameraman (c) Pepper (d) Clock (e) Text

Fig. 4.4 Images estimated from observed images in Fig. 4.3 under SNR= 30dB with Algorithm 2



(a) Algorithm 2 (b) Bilinear (c) Kanemura (d) Babacan (e) Algorithm 1

Fig. 4.5 Comparison of estimated images with Algorithm 2 and methods by Bilinear interpolation, Kanemura *et al.*, Babacan *et al.*, and Algorithm 1 under SNR= 30dB

4.8 Discussion

4.8.1 Super-resolution Model

With regard to the observation model, we used a linear transformation and AWGN. The use of the linear transformation model is advantageous since an arbitrary transformation matrix $\boldsymbol{W}(\boldsymbol{\phi}_l)$ can be employed because of the Taylor approximation. The transformation matrix can be constructed by multiplying three matrices: the warping,

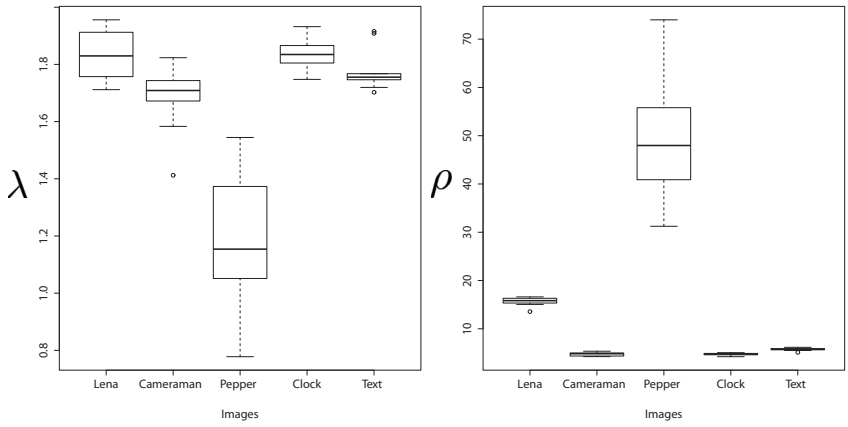
Table 4.2 RMSEs of registration parameters (a lower value is better) for different SNR levels

parameter	SNR	RMSE		
	[dB]	(Algorithm 1)	(Kanemura)	(Babacan)
θ	20	0.006	0.006	0.006
	25	0.004	0.004	0.004
	30	0.002	0.003	0.003
$[\vec{o}]_h$	20	0.094	0.095	0.094
	25	0.054	0.059	0.056
	30	0.041	0.060	0.046
$[\vec{o}]_v$	20	0.074	0.073	0.076
	25	0.044	0.052	0.047
	30	0.037	0.044	0.036
γ	20	0.031	0.033	—
	25	0.025	0.030	—
	30	0.028	0.028	—

blurring, and downsampling matrices [142]. A disadvantage of this is that sub-pixel errors might accumulate. We prefer matrix construction via a continuous function [141]. We improved the construction by introducing an elliptic theta function for the normalizing constant in (4.10). This normalizing constant provides fair pixel weights for both marginal and central areas of the HR image and faithfully represents the Gaussian PSF.

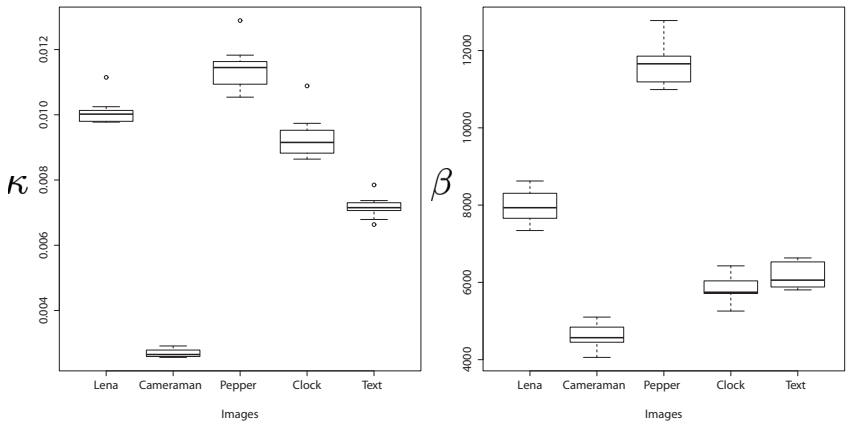
With regard to the HR image prior, we used a two types of prior, that is "causal" Gaussian MRF prior and "compound" Gaussian MRF prior. They usually have an exponential calculation cost, $\mathcal{O}(2^{N_{\boldsymbol{\eta}}})$, for the normalizing constant or, equivalently, the partition function, and this is an obstacle to direct calculation of the PM solution. The MAP solution has been used in work elsewhere because it is not affected by the normalizing constant. We have shown how we can adopt these priors for PM SR and that our algorithms have only a polynomial calculation cost $\mathcal{O}(N_{\boldsymbol{x}}^3)$. From our experiments, we think "compound" Gaussian MRF prior is considered preferable to a "causal" Gaussian MRF prior as a natural image prior.

With regard to the hyperparameter priors, we also improved the existing method. As the edge penalty parameter λ, Kanemura *et al.* [143] implicitly assumed $\lambda \in \mathbb{R}$, which leads to a negative λ and consequently results in an edge-strewn image. We assumed $\lambda > 0$ by setting its prior according to a gamma distribution, resulting in an appropriate inference. As the smoothness parameter ρ, they practically fixed the



(a) Lambda

(b) Rho



(c) Kappa

(d) Beta

Fig. 4.6 Box and whisker plot of the PM for each hyperparameter, λ, ρ, κ, and β, and image under SNR= 30dB noise

value of ρ with a strongly informative prior. We chose a non-informative prior for ρ. We show the box and whisker plot of the PM for each hyperparameter over 10 experiments on each image under SNR= 30dB noise in Fig. 4.6. As can be seen, the inferred value of the PM of ρ showed wide variation, with an approximately 10-fold maximum-to-minimum ratio, depending on the original image. This result can be interpreted as meaning it is worth inferring ρ in each HR image. Furthermore, λ and κ respectively showed approximately 2-fold and 4-fold ranges of variation. Regarding the contrast parameter κ, they assumed $\kappa \equiv 0$, which leads to $|\boldsymbol{A}| = 0$, and this results in an improper normalizing constant. While we assume $\kappa > 0$, which leads to a proper normalizing constant, we can consequently take the term of $\ln|\boldsymbol{A}|$ into account in the update equations of the VB.

With regard to the prior distribution for the blurring parameter γ, we used a Gaussian distribution even though γ is a positive real number. This is because we selected a simpler expression. We tried using the prior of the gamma distribution as γ, but the improvement was small. One disadvantage of this model is that a non-informative setting for this prior may lead to a nonsense result where the inferred γ is negative. Moreover, we employed a somewhat informative prior for γ. This is because the blurring parameter γ and smoothness hyperparameter ρ are somewhat complementary. This means that simultaneous estimation of γ and ρ is difficult. Tipping *et al.* [141] and Kanemura *et al.* [143] fixed ρ, and Babacan *et al.* [147] fixed γ.

4.8.2 Computational Algorithm based on Variational Bayes and Taylor Approximations

With regard to the Taylor approximation for the transformation matrix $\boldsymbol{W}(\boldsymbol{\phi}_l)$, we used the first-order approximation in (4.25) because it is more stable than the second-order approximation. This first-order approximation was proposed by Villena *et al.* [147]. The second-order approximation was proposed by Pickup *et al.* [145], and they obtained good results. We also tried the second-order approximation, but it sometimes made the algorithm unstable because it sometimes failed to produce a positive definite matrix for the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{x}}$.

With regard to the Taylor approximation for $\ln|\boldsymbol{A}(\boldsymbol{\eta},\rho,\kappa)|$, $\ln$ Sigmoid(λ), and $\ln\sum_{\boldsymbol{\eta}}\exp\left\{-\lambda\sum_{i\sim j}(1-\eta_{i,j})-\frac{1}{2}\ln\left|\frac{1}{2\pi}\boldsymbol{A}(\boldsymbol{\eta},\rho,\kappa)\right|\right\}$, we introduced the first-order approximation in (4.28) - (4.30). Note that the Taylor expansion not with respect to ρ,κ,λ, but with respect to $\ln\rho,\ln\kappa,\ln\lambda$ is our key idea to solve the conjugate prior problem. Indeed, we could successfully derive the terms originating from $\ln|\boldsymbol{A}|$ in update equations ((4.38), (4.65), (4.46), (4.72), (4.48), and (4.74)). Kanemura *et al.* [143, 144] ignored the term of $\ln|\boldsymbol{A}|$ because of the high calculation cost, and this would result in less accurate inference. As for $\boldsymbol{\eta}$, we implicitly assumed that $\boldsymbol{\eta}$ is not a binary vector but a continuous vector and did the differentiation. This assumption is based on (4.15). If we make another assumption – i.e., replacement of $\eta_{i,j}$ with $\eta_{i,j}^2$ in Eq. (4.15) – Eq. (4.15) has the same meaning, but the result of the Taylor approximation will differ from the current form.

4.8.3 Discussion on Experimental Results

With regard to the experimental results, the proposed method outperforms the other methods in terms of the ISNR for most images and noise levels. Moreover, its

estimation of the registration parameters was more accurate than ones of the other methods for most conditions. Therefore, we conclude the proposed method is on the whole superior to the other methods. Compared with bilinear interpolation and Kanemura's method, the superiority of the proposed method was clear. Compared with the Babacan's method, the superiority of the proposed method was relatively slight. We think that the reason is our numerical optimization method falls slightly short of optimization because the proposed method uses more approximations than other methods. Especially, in the case of the Pepper image in 40 dB noise, the porposed method was worse than the Babacan's method. This inferiority is considered to be caused by unstable estimation of γ and ρ, where Babacan's method fixed the value of γ to the true expected value in our implementation. Intuitively, the Pepper image is smoother than the other images and has fewer edges. Therefore, this feature is considered to be less preferable for complementary parameters of γ and ρ.

With regard to the calculation cost, the proposed algorithm requires $\mathcal{O}(N_{\boldsymbol{x}}^3)$. This calculation cost order is given by two matrix inversions: $\boldsymbol{\Sigma}_{\boldsymbol{x}}^{(t+1)}$ in (4.42) and $\boldsymbol{A}$ in (4.38), (4.65), (4.48), and (4.74). We found that a simple approximation such as considering all the off-diagonal elements to be zero reduces the calculation time but obviously degrades accuracy. We hope to solve this problem in our future work.

4.9 Summary

In this chapter, we proposed a Bayesian image super-resolution (SR) method with "causal" and "compound" Gaussian MRF priors. We improved current models with respect to three points: 1) the combined transformation model through a preferable normalization term using the elliptic theta function, 2) the "causal" and "compound" Gaussian MRF models through introduction of a contrast parameter κ, which provides an effective normalizing constant including $\ln|\boldsymbol{A}|$, and 3) the hyperparameter prior model through application of a gamma distribution for the edge penalty parameter λ, which prevents an unfavorable edge-strewn image. We then logically derived the optimal estimation function, that is, not the joint MAP or marginalized ML but the PM. The estimation function is computed by using VB. We solved the conjugate prior problem in VB by introducing Taylor approximations. Other than these approximations, we did not use any approximations such as ignoring the term $\ln|\boldsymbol{A}|$. Experimental results showed that the proposed method is mostly superior to current methods in accuracy.

Chapter.5

Bayesian Input Selective Regression

5.1 Introduction

Supervised learning is a fundamental task in machine learning and artificial intelligence [13, 162]. A focus of such a task is in learning a model representing the relationship between observed data and a corresponding label, wherein the learned model can be used for assigning label to new observed data without label. The task is performed using labeled training data that consists of pairs of data and labels.

The quantity and quality of labeled training data has a huge influence on the quality of the learned model. Recently, the cost of preparing a huge amount of labeled training data has decreased thanks to growth in crowdsourcing services, social networking services, and sensor networks [163]. We can learn a model by using the huge amount of the data labeled in these ways. However, the training data acquired by these means may often contain wrong labels and be likely a mixed bag. When the quality of the training data is expected to be low, we would traditionally use a robust method such as one based on a heavy-tailed distribution [123, 164–166]. Also, when we use crowdsourcing for labeling, we should use one of many approaches that can handle the label quality in such situations [167–169]. Most current methods comprising these approaches explicitly or implicitly rely on the assumption that the proportion of correct labels in the training data is higher than that of wrong labels. Using weighting techniques for assessing the noise strength, ability of crowd workers, and instance difficulty, they learn a model by majority rule of the labels.

In particular, in the case of supervised learning on time-sequential data, it is possible that the proportion of the correct labels is *lower than* that of the wrong labels. A label is not attached to a point of the sequence but rather attached to a time window of the sequence. In this case, only a small part of the data in this window likely reflects the label, and the remaining part does not reflect it, as shown in Figure 5.1. In this case, since the proportion of correct labels may be lower than that of the wrong ones,

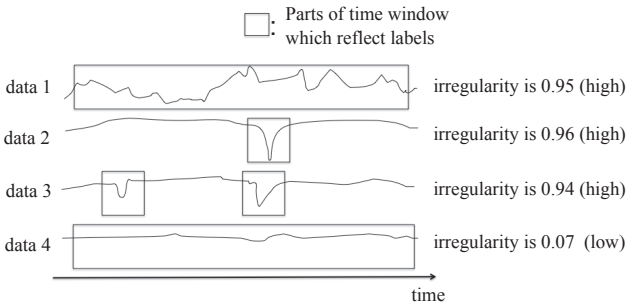


Fig. 5.1 Data 1, 2, and 3 are quite different time sequences but are attached similar high irregularity label because of the existence of parts having high irregularity. On the other hand, data 4 is similar to data 2 and 3 in most parts of the window, but is attached with low irregularity label. Note that the irregularity is defined in the interval $[0, 1]$.

we cannot use the majority rule in the robust methods. Also, since the feature vector from the whole window may not reflect the label, other methods cannot use such data to learn the model. They require the part of the sequence reflecting the label to be selected from the sequence.

In this chapter, we focus on a regression problem using such mixed bag data [40]. We formulate a problem in which we learn the regression model from sets of training data. Each of the sets has an only single label and only one of the training data sample in the set reflects the label. We propose a model to select valuable data from each of the sets for learning the desired regression model. Our model has hidden variables representing which of the training data sample in the set corresponds to the label. Based on the framework of Bayesian optimal estimation, we can simultaneously estimate the hidden variables and parameters of the regression model stably. We experimentally evaluated our method using artificial and real-world datasets in experiments.

5.2 Related Work

There have been studies on handling the uncertainty of labels. The majority of these studies has been on robust estimations, such as an estimation based on heavy-tailed distributions [123, 164–166]. The t-regression, which is based on the student's t-distribution, is one of the most common robust regression methods [165, 170–173]. The L1-based estimation, which is related to median-based methods, is also commonly used [123, 174]. Most of these methods weight each of the training data sample based on its noise level and prune the data to which a large amount of noise was added during the learning of the regression model. The literature on crowdsourcing has studied on explicitly handling the uncertainty of manual labeling [167–169]. These approaches

learn the regression and classification models robustly by improving the quality of labels, where they obtain multiple labels for each training data from multiple labelers and weight them based on the ability of labelers and difficulties of labeling examples.

A Multiple Instance Learning (MIL) problem [175–178] has handled such a mixed bag or the multiple instance in machine learning. However, our problem setting is different from the problem setting of the MIL, which requires to handle the mixed bag even in prediction. It would be interesting future work to apply our fully Bayesian approach to a MIL problem by modifying our formulation.

5.3 Problem Setting

Suppose we are given N sets of training samples, $\{\boldsymbol{X}^{(n)}\}_{n=1}^{N}$, and the n-th set $\boldsymbol{X}^{(n)}$ has K training samples as $\boldsymbol{X}^{(n)} \equiv \{\boldsymbol{x}_1^{(n)}, \boldsymbol{x}_2^{(n)}, \ldots, \boldsymbol{x}_K^{(n)}\}$, where $\boldsymbol{x}_k^{(n)} \in \mathbb{R}^D$ is a D-dimensional feature vector for the k-th sample in the n-th set. A single label $y^{(n)} \in \mathbb{R}$ is attached to the n-th set. Then the N sets of the labels can be represented as $\boldsymbol{y} \equiv \{y^{(1)}, y^{(2)}, \ldots, y^{(N)}\}$. One of the K training samples in the n-th set corresponds to the label $y^{(n)}$, but we do not know which of them it is.

Our goal is to learn the relationship between the feature vector $\boldsymbol{x}$ and the label y by using the given data $\{\boldsymbol{X}^{(n)}\}_{n=1}^{N}$ and $\boldsymbol{y}$, and we use the relationship for making the prediction.

5.4 Posterior Mean Estimation for Input Selective Regression

We formalize this prediction problem as the estimates of y by the estimation function, $y^*(\boldsymbol{x}, \{\boldsymbol{X}^{(n)}\}_{n=1}^{N}, \boldsymbol{y})$, to which the observed variables $\boldsymbol{x}$, $\{\boldsymbol{X}^{(n)}\}_{n=1}^{N}$ and $\boldsymbol{y}$ have been input.

We first consider the evaluation criterion based on the Bayesian perspective for this task. Since the label y is a real number, we use the L2-norm error (mean square error) as the evaluation criterion. We define the error function for the task as the squared difference between y and the estimate by the estimation function $y^*(\boldsymbol{x}, \{\boldsymbol{X}^{(n)}\}_{n=1}^{N}, \boldsymbol{y})$:

$$\begin{aligned} &\mathrm{Error}(y, y^*(\boldsymbol{x}, \{\boldsymbol{X}^{(n)}\}_{n=1}^{N}, \boldsymbol{y})) \\ &\qquad \equiv \|y - y^*(\boldsymbol{x}, \{\boldsymbol{X}^{(n)}\}_{n=1}^{N}, \boldsymbol{y})\|_2^2, \end{aligned} \tag{5.1}$$

Using the model parameters $\boldsymbol{\theta}$ which are explicitly defined later, we define the evaluation criterion as the minimization of the population mean of the error function Eq. (5.1):

$$\underset{y^*(\boldsymbol{x}, \{\boldsymbol{X}^{(n)}\}_{n=1}^{N}, \boldsymbol{y})}{\operatorname{argmin}} \left\langle \|y - y^*(\boldsymbol{x}, \{\boldsymbol{X}^{(n)}\}_{n=1}^{N}, \boldsymbol{y})\|_2^2 \right\rangle_{p(y, \boldsymbol{x}, \{\boldsymbol{X}^{(n)}\}_{n=1}^{N}, \boldsymbol{y}, \boldsymbol{\theta})} \tag{5.2}$$

Then, we can derive the optimal estimation function using the result in Eq. (2.14) as the PM,

$$\begin{aligned}
&\hat{y}^*(\boldsymbol{x}, \{\boldsymbol{X}^{(n)}\}_{n=1}^N, \boldsymbol{y}) \qquad (5.3)\\
&= \underset{y^*(\boldsymbol{x}, \{\boldsymbol{X}^{(n)}\}_{n=1}^N, \boldsymbol{y})}{\operatorname{argmin}} \left\langle \|y - y^*(\boldsymbol{x}, \{\boldsymbol{X}^{(n)}\}_{n=1}^N, \boldsymbol{y})\|_2^2 \right\rangle_{p(y, \boldsymbol{x}, \{\boldsymbol{X}^{(n)}\}_{n=1}^N, \boldsymbol{y}, \boldsymbol{\theta})}\\
&= \int y \int p(y|\boldsymbol{x}, \boldsymbol{\theta}) \left(\prod_{n=1}^N p(y^{(n)}|\boldsymbol{X}^{(n)}, \boldsymbol{\theta}) \right) p(\boldsymbol{\theta}) \mathrm{d}\theta \mathrm{d}y,\\
&= \int y \; p(y|\boldsymbol{x}, \{\boldsymbol{X}^{(n)}\}_{n=1}^N, \boldsymbol{y}) \mathrm{d}y,
\end{aligned}$$

where the posterior distribution $p(y|\boldsymbol{x}, \{\boldsymbol{X}^{(n)}\}_{n=1}^N, \boldsymbol{y})$ represents the probability distribution of the label y given $\boldsymbol{x}$, $\{\boldsymbol{X}^{(n)}\}_{n=1}^N$, and $\boldsymbol{y}$. As shown in Eq. (5.3), the posterior distribution consists of the regression model and the prior model for the model parameters. In the following section, we propose them.

5.5 Probabilistic Hidden Structure Modeling for Input Selective Regression

5.5.1 Bayesian Regression Model for Selecting a Valuable Subset

We design a regression model by introducing hidden variables $\boldsymbol{h}^{(n)} \in \{0, 1\}^K$, $\sum_{k=1}^K h_k^{(n)} = 1$ that represent which of the K training samples in the n-th set corresponds to the n-th label $y^{(n)}$ in the 1-of-K notation, as shown in Figure 5.2. For example, if $\boldsymbol{h}^{(n)} = [1, 0, 0, 0, ...]$, the 1-st training sample $\boldsymbol{x}_1^{(n)}$ in the n-th set corresponds to the n-th label $y^{(n)}$. If $\boldsymbol{h}^{(n+1)} = [0, 0, 1, 0, ...]$, the 3-rd training sample $\boldsymbol{x}_3^{(n+1)}$ in the $n+1$-st set corresponds to the $n+1$-st label $y^{(n+1)}$. The N set of hidden variables is represented as $\boldsymbol{H} \equiv \{\boldsymbol{h}^{(n)}\}_{n=1}^N$. Although we use the 1-of-K notation for $\boldsymbol{h}^{(n)}$, our learning procedure estimates $\boldsymbol{h}^{(n)}$ probabilistically. Thus, we can represent a situation in which multiple samples in the n-th set correspond to the n-th label $y^{(n)}$ with specific weights, such as $[0.3, 0.1, 0.6, 0, ...]$.

Next, we define the regression model for $\boldsymbol{X}^{(n)}$ and $y^{(n)}$ when $h_k = 1$ as

$$p(y^{(n)}|\boldsymbol{X}^{(n)}, h_k^{(n)} = 1, \boldsymbol{w}, \beta) \equiv \mathcal{N}(y^{(n)}|\boldsymbol{w}^\top \boldsymbol{x}_k^{(n)}, \beta^{-1}), \qquad (5.4)$$

where the parameters $\boldsymbol{w} \equiv [w_1, w_2, \ldots, w_D]^\top \in \mathbb{R}^D$ and $\beta > 0$ are model parameters to be learned. In particular, $\boldsymbol{w}$ represents the regression coefficients, and the d-th element in $\boldsymbol{w}$ is that for the d-th feature in $\boldsymbol{x}_k^{(n)}$.

Since we do not know which of the K training samples in $\boldsymbol{X}^{(n)}$ corresponds to the label $y^{(n)}$, an arbitrary element in $\boldsymbol{h}$ can become one. Thus, our model has K mixture

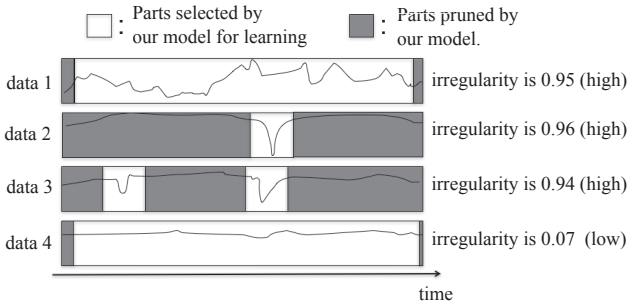


Fig. 5.2 Our model selects useful parts for learning the model.

components such that

$$p(y^{(n)}|\boldsymbol{X}^{(n)}, \boldsymbol{h}^{(n)}, \boldsymbol{w}, \beta) \equiv \prod_{k=1}^{K} \mathcal{N}\left(y^{(n)}\middle|\boldsymbol{w}^\top \boldsymbol{x}_k^{(n)}, \beta^{-1}\right)^{h_k^{(n)}} \tag{5.5}$$
$$= \frac{\exp\left(-\frac{\beta}{2}\sum_{k=1}^{K} h_k^{(n)}(y^{(n)} - \boldsymbol{w}^\top \boldsymbol{x}_k^{(n)})^2\right)}{(2\pi\beta^{-1})^{\frac{1}{2}}}.$$

Through the estimation of $\boldsymbol{h}^{(n)}$ in this model for the n-th set, we can select valuable training samples from the n-th set for learning the regression model.

We define the regression model for the prediction as

$$p(y|\boldsymbol{x}, \boldsymbol{w}, \beta) \equiv \mathcal{N}(y|\boldsymbol{w}^\top \boldsymbol{x}, \beta^{-1}), \tag{5.6}$$

where the parameters $\boldsymbol{w}$ and β are same as them in Eq. (5.5).

5.5.2 Conjugate Priors for Model Parameters

For the prior distributions of $\boldsymbol{H}$ and β, we simply use the conjugate priors:

$$p(\boldsymbol{H}) \equiv \prod_{n=1}^{N} \text{Categorical}\left(\boldsymbol{h}^{(n)}\middle|\boldsymbol{\xi}_{\boldsymbol{h}}^{(0)}\right), \tag{5.7}$$
$$p(\beta) \equiv \text{Gamma}\left(\beta\middle|a_\beta^{(0)}, b_\beta^{(0)}\right), \tag{5.8}$$

where the parameters $\boldsymbol{\xi}_{\boldsymbol{h}}^{(0)}$, $a_\beta^{(0)}$, and $b_\beta^{(0)}$ are treated as input parameters given as part of the model. We chose the hyperparameter values in Eqs. (5.7) - (5.8) to be as non-informative as possible and to have a quite flat distribution: $\boldsymbol{\xi}_{\boldsymbol{h}}^{(0)} = 10^{-10} \times \boldsymbol{i}$, $a_\beta^{(0)}/N = b_\beta^{(0)}/N = 10^{-10}$ and $\boldsymbol{\Sigma}_{\boldsymbol{w}}^{(0)} = 10^{10} \times \boldsymbol{I}$.

For pruning irrelevant features in the feature vector $\boldsymbol{x}$, we use the automatic relevance determination (ARD) prior [179–181] as the prior of the coefficients $\boldsymbol{w}$:

$$p(\boldsymbol{w}|\boldsymbol{\alpha}) \equiv \prod_{d=1}^{D} \mathcal{N}(w_d|0, \alpha_d). \tag{5.9}$$

Similarly to [182], this is the conjugate prior distribution for $\boldsymbol{w}$. The parameter $\boldsymbol{\alpha}$ can be also estimated in Bayesian framework. Using the ARD prior, we can get a sparse solution for $\boldsymbol{w}$: many of their estimated coefficients are zero.

We define hyperprior distributions for $\boldsymbol{\alpha}$ using the conjugate priors:

$$p(\boldsymbol{\alpha}) \equiv \prod_{d=1}^{D} \text{Gamma}\left(\alpha_d \middle| a_\alpha^{(0)}, b_\alpha^{(0)}\right), \tag{5.10}$$

where the hyperparameter values in Eq. (5.10) are also non-informative: $a_\alpha^{(0)} = b_\alpha^{(0)} = 10^{-10}$.

5.5.3 Joint Distribution

We can now write down the joint distribution for the random variables $\boldsymbol{y}$, $\boldsymbol{H}$, $\boldsymbol{w}$, β, $\boldsymbol{\alpha}$ as,

$$\begin{aligned} &p(\boldsymbol{y}, \boldsymbol{H}, \boldsymbol{w}, \beta, \boldsymbol{\alpha} | \{\boldsymbol{X}^{(n)}\}_{n=1}^N) \\ &\quad \equiv \left(\prod_{n=1}^{N} p(y^{(n)} | \boldsymbol{X}^{(n)}, \boldsymbol{h}^{(n)}, \boldsymbol{w}, \beta) p(\boldsymbol{h}^{(n)}) \right) p(\boldsymbol{w}|\boldsymbol{\alpha}) p(\beta) p(\boldsymbol{\alpha}). \end{aligned} \tag{5.11}$$

All marginal and conditional distributions including the posteriors $p(\boldsymbol{H}|\{\boldsymbol{X}^{(n)}\}_{n=1}^N, \boldsymbol{y})$, $p(\boldsymbol{w}|\{\boldsymbol{X}^{(n)}\}_{n=1}^N, \boldsymbol{y})$, $p(\beta|\{\boldsymbol{X}^{(n)}\}_{n=1}^N, \boldsymbol{y})$, and $p(\boldsymbol{\alpha}|\{\boldsymbol{X}^{(n)}\}_{n=1}^N, \boldsymbol{y})$ can be derived in terms of this joint distribution.

5.6 Variational Bayes Algorithm for Posterior Mean Estimation

Here, we design a learning algorithm for simultaneously estimating the hidden variables $\boldsymbol{H}$ for the N sets and the parameters $\boldsymbol{w}$ and β of the proposed model from the training data, $\{\boldsymbol{X}^{(n)}\}_{n=1}^N$ and $\boldsymbol{y}$. In the probabilistic formulation, the goal is to find the posterior distributions $p(\boldsymbol{H}|\{\boldsymbol{X}^{(n)}\}_{n=1}^N, \boldsymbol{y})$, $p(\boldsymbol{w}|\{\boldsymbol{X}^{(n)}\}_{n=1}^N, \boldsymbol{y})$, $p(\beta|\{\boldsymbol{X}^{(n)}\}_{n=1}^N, \boldsymbol{y})$, and $p(\boldsymbol{\alpha}|\{\boldsymbol{X}^{(n)}\}_{n=1}^N, \boldsymbol{y})$ which represent the probability distributions for $\boldsymbol{H}$, $\boldsymbol{w}$, β and $\boldsymbol{\alpha}$ given the training data.

Here, it is not possible to obtain an exact analytical solution for the posteriors. Instead, we will derive an approximate solution by using the VB method [52].

According to the formulation in Chapter 2, the VB approach approximately finds the posterior distribution over the set of unobserved variables, $p(\boldsymbol{H}, \boldsymbol{w}, \beta, \boldsymbol{\alpha}|\{\boldsymbol{X}^{(n)}\}_{n=1}^N, \boldsymbol{y})$, in a factorized form:

$$\begin{aligned} p(\boldsymbol{H}, \boldsymbol{w}, \beta, \boldsymbol{\alpha}|\{\boldsymbol{X}^{(n)}\}_{n=1}^N, \boldsymbol{y}) &\approx q(\boldsymbol{H}, \boldsymbol{w}, \beta, \boldsymbol{\alpha}) \\ &\equiv q(\boldsymbol{H}) q(\boldsymbol{w}) q(\beta, \boldsymbol{\alpha}). \end{aligned} \tag{5.12}$$

Table 5.1 Estimated parameters for each case of $K \in \{2, 5, 10\}$ with normal noise (noise precision $\beta = 10$).

	value of each element in $\boldsymbol{w}$						β
true	1.5	−2	0.5	0	0	0	10
estimated ($K = 2$)	1.5	−2.0	0.50	0.0	0.0	0.0	10
estimated ($K = 5$)	1.5	−2.0	0.50	0.0	0.0	0.0	10
estimated ($K = 10$)	1.5	−2.0	0.51	0.0	0.0	0.0	10

We identify the optimal approximate distribution that minimizes the KL divergence [183] from the approximate distribution $q(\boldsymbol{H}, \boldsymbol{w}, \beta, \boldsymbol{\alpha})$ to the true posterior distribution $p(\boldsymbol{H}, \boldsymbol{w}, \beta, \boldsymbol{\alpha} | \{\boldsymbol{X}^{(n)}\}_{n=1}^N, \boldsymbol{y})$ as the best approximation of the true distribution.

From Eqs. (2.26), (2.27), (5.5) and (5.7) - (5.10), we solve the following iterative updating equations:

$$q(\boldsymbol{H}) = \prod_{n=1}^{N} \text{Categorical}\left(\boldsymbol{h}^{(n)} \middle| \boldsymbol{\xi}_{\boldsymbol{h}^{(n)}}\right), \tag{5.13}$$

$$q(\boldsymbol{w}) = \mathcal{N}\left(\boldsymbol{w} \middle| \boldsymbol{\mu}_{\boldsymbol{w}}, \boldsymbol{\Sigma}_{\boldsymbol{w}}\right), \quad \text{and} \tag{5.14}$$

$$q(\beta, \boldsymbol{\alpha}) = \text{Gamma}\left(\beta \middle| a_\beta, b_\beta\right) \prod_{d=1}^{D} \text{Gamma}\left(\alpha_d \middle| a_{\alpha_d}, b_{\alpha_d}\right), \tag{5.15}$$

where the mean values of the hyperparameters β and α_d over the trial distributions $q^{(t)}(\beta, \boldsymbol{\alpha})$ at step t on VB algorithm are given by

$$\mu_\beta^{(t)} = \frac{a_\beta^{(t)}}{b_\beta^{(t)}}, \quad \mu_{\alpha_d}^{(t)} = \frac{a_{\alpha_d}^{(t)}}{b_{\alpha_d}^{(t)}}. \tag{5.16}$$

Here are the specific update equations at step $t+1$:

$$\xi_{h_k^{(n)}}^{(t+1)} = \frac{\exp\left[\xi_{h_k}^{(0)} - \frac{1}{2}\mu_\beta^{(t)} c_{n,k}\right]}{\sum_{j=1}^{K} \exp\left[\xi_{h_j}^{(0)} - \frac{1}{2}\mu_\beta^{(t)} c_{n,j}\right]}, \text{ where} \tag{5.17}$$

$$c_{n,k} \equiv \left(y^{(n)} - [\boldsymbol{\mu}_{\boldsymbol{w}}^{(t)}]^\top \boldsymbol{x}_k^{(n)}\right)^2 + \text{tr}\left(\boldsymbol{x}_k^{(n)} \left[\boldsymbol{x}_k^{(n)}\right]^\top \boldsymbol{\Sigma}_{\boldsymbol{w}}^{(t)}\right). \tag{5.18}$$

$$\boldsymbol{\mu}_{\boldsymbol{w}}^{(t+1)} = \boldsymbol{\Sigma}_{\boldsymbol{w}}^{(t+1)} \left[\mu_{\beta}^{(t)} \sum_{n=1}^{N} \sum_{k=1}^{K} \xi_{h_k^{(n)}}^{(t)} y^{(n)} \boldsymbol{x}_k^{(n)} \right], \tag{5.19}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{w}}^{(t+1)} = \left[\mu_{\beta}^{(t)} \sum_{n=1}^{N} \sum_{k=1}^{K} \xi_{h_k^{(n)}}^{(t)} \boldsymbol{x}_k^{(n)} \left[\boldsymbol{x}_k^{(n)} \right]^{\top} + \boldsymbol{\mu}_{\boldsymbol{\alpha}}^{(t)} \boldsymbol{I} \right]^{-1},$$

$$a_{\beta}^{(t+1)} = a_{\beta}^{(0)} + \frac{1}{2} N, \tag{5.20}$$

$$b_{\beta}^{(t+1)} = b_{\beta}^{(0)} + \frac{1}{2} \sum_{n=1}^{N} \sum_{k=1}^{K} \xi_{h_k^{(n)}}^{(t)} c_{n,k}, \tag{5.21}$$

$$a_{\alpha_d}^{(t+1)} = a_{\alpha}^{(0)} + \frac{1}{2}, \quad \text{and} \tag{5.22}$$

$$b_{\alpha_d}^{(t+1)} = b_{\alpha}^{(0)} + \frac{1}{2} \Big([\mu_{w_d}^{(t)}]^2 + [\boldsymbol{\Sigma}_{\boldsymbol{w}}^{(t)}]_{d,d} \Big). \tag{5.23}$$

We can iteratively update q by simply computing only the parameters of these distributions in Eqs. (5.13) - (5.15). For the initial values of the parameters, we can use the same values as those of the corresponding priors. In practice, we stop the VB iterations when the relative differences between the current values of the variables, $\boldsymbol{z}_c$, and the previous values of the variables, $\boldsymbol{z}_p$, are sufficiently low:

$$\frac{\|\boldsymbol{z}_c - \boldsymbol{z}_p\|_2^2}{\|\boldsymbol{z}_p\|_2^2} < 10^{-5}. \tag{5.24}$$

After the above stopping condition is satisfied, we obtain the final outcome $q(\boldsymbol{H})$, $q(\boldsymbol{w})$, $q(\beta)$, and $q(\boldsymbol{\alpha})$ directly, which corresponds to an approximation of the learned posteriors, $p(\boldsymbol{H}|\{\boldsymbol{X}^{(n)}\}_{n=1}^N, \boldsymbol{y})$, $p(\boldsymbol{w}|\{\boldsymbol{X}^{(n)}\}_{n=1}^N, \boldsymbol{y})$, $p(\beta|\{\boldsymbol{X}^{(n)}\}_{n=1}^N, \boldsymbol{y})$, and $p(\boldsymbol{\alpha}|\{\boldsymbol{X}^{(n)}\}_{n=1}^N, \boldsymbol{y})$ since the distribution of $\boldsymbol{H}$, $\boldsymbol{w}$, β, and $\boldsymbol{\alpha}$ has been factorized as shown in Eq. (5.12).

Using the learned $\boldsymbol{\mu}_{\boldsymbol{w}}$, we can predict the label for the new data as follows:

$$y \equiv \boldsymbol{\mu}_{\boldsymbol{w}}^{\top} \boldsymbol{x}. \tag{5.25}$$

Note that we can directly estimate the our objective, that is the predictive posterior distribution $p(y|\boldsymbol{X}, \{\boldsymbol{X}^{(n)}\}_{n=1}^N, \boldsymbol{y})$, by using the VB method. However, since the predictive posterior distribution requires VB iterations for each new data, it is quite costly to compute. Instead, we use Eq. (5.25) as an approximation of the predictive PM of $\hat{y}^*$ in Eq. (5.3) that can be computed with a much lower computational cost. We will discuss this approximation in the Discussion section.

5.7 Experimental Results

We assessed the effectiveness of our approach in numerical experiments. First, we artificially generated datasets to study the performance of our algorithm (Sec-

Table 5.2 Estimated parameters for each case of $K \in \{2, 5, 10\}$ with high noise (noise precision $\beta = 1$).

	value of each element in $\boldsymbol{w}$						β
true	1.5	−2	0.5	0	0	0	1
estimated ($K = 2$)	1.6	−2.1	0.53	0.0	0.0	0.0	1.0
estimated ($K = 5$)	1.7	−2.3	0.60	0.0	0.0	0.0	0.94
estimated ($K = 10$)	2.7	−3.6	1.2	0.0	0.0	0.0	0.33

tion 5.7.1). We then applied it to real-world time-sequential data from the UCI machine learning repository [184] (Section 5.7.2).

5.7.1 Experiment on Artificial Dataset

We studied the validity of our algorithm by simultaneously estimating $\boldsymbol{H}$, $\boldsymbol{w}$, and β from the artificial validation dataset. In preparing the artificial validation dataset, we randomly generated $N \times K$ training samples, $\{\boldsymbol{X}^{(n)}\}_{n=1}^{N}$, from the standard Gaussian distribution $\mathcal{N}(\boldsymbol{x}|0, \boldsymbol{I})$, where the number of dimensions of $\boldsymbol{x}$ was 6. Then, using $\{\boldsymbol{X}^{(n)}\}_{n=1}^{N}$, we generated the corresponding N sets of labels $\boldsymbol{y}$ from the distribution in Eq. (5.6), where we randomly selected one of the K training samples in each n-th set from a uniform distribution, and a limited number of the coefficients, $\boldsymbol{w}$, had non-zero values, *i.e.*, $\boldsymbol{w} = \{1.5, -2.0, 0.5, 0, 0, 0\}$. We repeatedly evaluated the proposed method for each of the following settings: the noise precision $\beta \in \{10, 1\}$, which correspond normal and high noise settings, and number of training samples in each training set $K \in \{2, 5, 10\}$. In the case of $K = 10$, only 10 percent of the data correctly corresponds to labels. In general, it is quite hard to learn regression models using such data. In this experiment, the number of training sets was $N = 10000$.

Tables 5.1 and 5.2 compares the estimated $\boldsymbol{w}$ and β with the true ones. Also, Table 5.3 shows the estimation accuracy of $\boldsymbol{h}$, which is defined as the proportion of indexes in which the maximum value in the estimated $\boldsymbol{h}$ is exactly the same as the true one selected in generating the data, where 1 is the best and 0 is the worst. The result confirms that our method can simultaneously estimate all of the parameters and hidden variables well except for the most difficult setting in which $K = 10$ and $\beta = 1$. Note that we can get a sparse solution for the coefficient thanks to the ARD prior.

Finally, Figure 5.3 compares our approach with common regression methods, which are t-regression [165, 170–173] with L1-regularization [185], relevance vector machine (RVM) with a linear kernel [181, 182], and random forest [186]. Since these baseline

Table 5.3 Estimation accuracy of hidden variables in each case of $K \in \{2, 5, 10\}$ and noise precision $\beta \in \{10, 1\}$. Chance level is the accuracy that would be expected by random choices.

	$K = 2$	$K = 5$	$K = 10$
chance level	0.5	0.2	0.1
proposed (noise precision $\beta = 10$)	0.95	0.81	0.65
peoposed (noise precision $\beta = 1$)	0.84	0.55	0.22

methods are not able to select valuable samples, in the training of these models, they select one of the K training samples in the n-th set from the same uniform distribution used to generate the data. We evaluated the results with regard to the mean absolute error (MAE) over M test samples, which were generated from the same distribution as the training samples, and the number of test samples M was $M = 10000$. MAE is defined as

$$\mathrm{MAE} \equiv \frac{1}{M} \sum_{m=1}^{M} \left| y_{\mathrm{true}}^{(m)} - y_{\mathrm{estimate}}^{(m)} \right|, \tag{5.26}$$

where $y_{\mathrm{true}}^{(m)}$ is the true y in the m-th test sample, and $y_{\mathrm{estimate}}^{(m)}$ is the estimated y for the m-th test sample. We computed the standard error of the absolute error (the error bars in Fig. 5.3). From Fig. 5.3, we can see that the overall performance of our method is significantly better than those of the alternatives. The t-regression with L1-regularization, which is the well-known robust regression method, achieved a good result in the case of $K = 2$, but it did not work in the case of $K = 5$ or $K = 10$. Our method can select the valuable samples from each n-th set in $\{\boldsymbol{X}^{(n)}\}_{n=1}^{N}$ and achieved the best performance in all of the settings. We chose MAE for evaluating the results since it is widely used for evaluating the accuracy in such uncertain and high noise settings having many outliers. MAE is more robust to outliers than the mean square error since it does not square the error. The mean square error penalizes large error more. If we square a big number, it becomes much larger relative to the others. The evaluation results with regard to the mean square error were almost same to the ones with regard to MAE, although the vertical scale of the graph with the mean square error became too large.

5.7.2 Experiment using UCI Dataset

We evaluated the proposed method in the prediction task for indoor temperatures from temporal sequences of sensor outputs in a house [187]. The dataset for this task

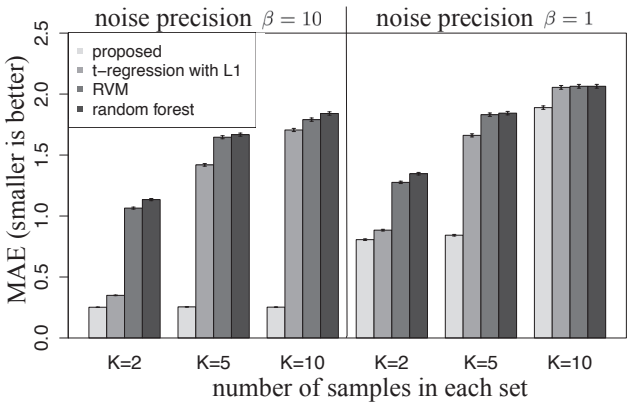


Fig. 5.3 Comparison of the proposed method and several regression methods in terms of MAE (smaller is better) on an artificial dataset. Error bars represent the standard error.

was a real-world dataset collected from the publicly available UCI machine learning repository [184].

The dataset consisted of 4137 samples and each sample had 24 number of attributes. Regarding the feature vector, we used all the attributes except for non numeric attributes and attributes always taking 0. In this problem setting, y is the indoor temperature at a future timestamp, which is standardized by subtracting its mean and dividing by its standard deviation, and $\boldsymbol{X}$ is the set of $K = 4$ number of training samples $\boldsymbol{x}$ which are computed from four different time windows in an hour before the timestamp; in particular, we use the features in the first fifteen minutes as $\boldsymbol{x}_1$, the features in the second fifteen minutes as $\boldsymbol{x}_2$, the features in the third fifteen minutes as $\boldsymbol{x}_3$, and the features in the last fifteen minutes as $\boldsymbol{x}_4$. Our model prunes ones corrupted by noise and outliers and selects valuable ones in the time windows for training the prediction model.

Table 5.4 compares our approach with RVM [181,182] with a linear kernel. Since the baseline method does not have the ability to select valuable samples, in the training of the model, it selects one of the K training samples in the n-th set from the uniform distribution. In prediction, we always use the features in the last fifteen minutes, $\boldsymbol{x}_4$, for both of our method and the baseline method. We evaluated the results with regard to the mean absolute error (MAE) in 5-fold cross validation using the dataset and also computed the standard error of the absolute error. From Table 5.4, we can see that the MAE of our method is 10% better than that of the baseline method.

Finally, Table 5.5 shows a typical examples of the estimation results of $\boldsymbol{h}$ that represent which of the K training samples in the n-th set corresponds to the n-th label. We can see that the estimation results of $\boldsymbol{h}$ are significantly different from each other. It suggests that the ability to select valuable training samples in each n-th set

Table 5.4 Comparison of the proposed method and baseline method in terms of MAE (smaller is better) on the UCI dataset.

method	MAE (smaller is better)
RVM	0.36 ± 0.0058
peoposed	$\mathbf{0.32} \pm 0.0055$

Table 5.5 Examples of estimation results of hidden variables $\boldsymbol{h}$.

data index	$h_1^{(n)}$	$h_2^{(n)}$	$h_3^{(n)}$	$h_4^{(n)}$
$n = 118$	0.0025	0.12	0.51	0.37
$n = 119$	0.65	0.26	0.07	0.02
$n = 120$	0.25	0.28	0.23	0.24
$n = 121$	0.12	0.25	0.22	0.41

is important for the prediction accuracy even in real-world case.

5.8 Discussion

5.8.1 Stability of Bayesian Inference

In the VB updates, we use the nature of Bayesian inference that we can evaluate the posterior of the estimation result [13, 188, 189]. By evaluating the confidence of the estimation result, which is computed using the posterior, of each of the variables at each step of VB, we can tune the update width properly on the estimations of the variables at each step and can obtain a stable final estimation result in a situation in which there are many variables to be learned. This property is quite useful in the case of that the confidence of the estimation result is important, such as in Bayesian optimization [190,191], Bayesian active learning [192,193], and Bayesian reinforcement learning [194–197].

5.8.2 Approximation of Predictive Posterior Mean

The approximation for the predictive posterior mean, as shown in Eq. (5.25), corresponds to the predictive posterior mean of y when we assume that $q(\boldsymbol{w})$ is the true

posterior of $\boldsymbol{w}$ and β is fixed by its mean value over the learned $q(\beta)$, $\frac{a_\beta}{b_\beta}$:

$$\begin{aligned}\hat{y}^*(\boldsymbol{x}, \{\boldsymbol{X}^{(n)}\}_{n=1}^N, \boldsymbol{y}) &= \int y\ p(y|\boldsymbol{x}, \{\boldsymbol{X}^{(n)}\}_{n=1}^N, \boldsymbol{y})\mathrm{d}y \\ &\approx \int y\ p\left(y\left|\boldsymbol{x}, \boldsymbol{w}, \beta = \frac{a_\beta}{b_\beta}\right.\right) q(\boldsymbol{w})\mathrm{d}\boldsymbol{w}\mathrm{d}y \\ &= \boldsymbol{\mu}_{\boldsymbol{w}}^\top \boldsymbol{x}.\end{aligned} \tag{5.27}$$

While we can skip the VB iterations for each new data thanks to this approximation, this type of the approximation sometimes underestimates the uncertainty of the data and causes an overfitting problem because β is not marginalized out from $p(y|\boldsymbol{x}, \boldsymbol{w}, \beta)$ and $\frac{a_\beta}{b_\beta}$ is just inputted into $p(y|\boldsymbol{x}, \boldsymbol{w}, \beta)$ as the point estimate of β. We can see this drawback in the experimental results using the artificial dataset with $K = 10$ and $\beta = 1$ in Table 5.2. The absolute values of the estimated $\boldsymbol{w}$ were higher than the true one. This indicates the overfitting of $\boldsymbol{w}$ to the "noisy" training data.

5.8.3 Other Applications of Input Selective Regression

We can straightforwardly extend the model in Eq. (5.5) so that it can handle a non-linear relationship between $\boldsymbol{x}$ and y by using the basis function or kernel function $\boldsymbol{\phi}(\boldsymbol{x}_k)$ as follows:

$$p(y|\boldsymbol{X}, \boldsymbol{h}, \boldsymbol{w}, \beta) \equiv \prod_{k=1}^K \mathcal{N}\left(y|\boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}_k), \beta^{-1}\right)^{h_k}. \tag{5.28}$$

We can also extend our model so that it can be applied to classification tasks with a specific link function and distribution. Similarly, we may use other noise models, such as the t-distribution. Such investigations will be for future work.

As stated in the Related Work section, it would be interesting to apply our fully Bayesian approach to an MIL problem [175–178] in future work. Since our problem setting is different from the problem setting of the MIL, which requires to handle the mixed bag or multiple instance even in prediction, we need to modify Eq. (5.25) for the MIL problem setting so that Eq. (5.25) can select the valuable sample in the newly observed bag.

Although we use the 1-of-K notation for $\boldsymbol{h}^{(n)}$, we can explicitly represent a situation in which multiple samples in the n-th set correspond to the n-th label $y^{(n)}$ by defining the hidden variables $\boldsymbol{h}^{(n)}$ as $\boldsymbol{h}^{(n)} \in \{0, 1\}^K, \quad \sum_{k=1}^K h_k^{(n)} \geq 1$. For example, if $\boldsymbol{h}^{(n)} = [1, 0, 1, 0, 0, ..., 0, ...]$, both of the 1-st and 3-rd training samples in the n-th set, $\boldsymbol{x}_1^{(n)}$ and $\boldsymbol{x}_3^{(n)}$, correspond to the n-th label, $y^{(n)}$. In this case, we can use the Bernoulli distribution as the prior for each $h_k^{(n)}$. Since such modeling has excessive flexibility

and is easily trapped into local optima, we need to constrain the value of $\sum_{k=1}^{K} h_k^{(n)}$ in practical use by, for example, more hierarchical modeling.

5.9 Summary

We formulated a regression problem selecting a valuable subset from each set of the mixed bag training data using Bayesian modeling with hidden variables. For the proposed model, we designed an efficient learning algorithm by using VB. Our method does not have any parameters that require careful tuning, thanks to its fully Bayesian modeling. Experimental results show that our approach performed better than baseline methods on an artificial dataset and on a real-world dataset. Our method can achieve robust regression even in the case in which only 10% of the data correctly corresponds to labels.

Chapter.6

Concluding Remarks

We investigated Bayesian optimal estimation with probabilistic hidden structure modeling, which allows us to obtain a stable solution for each problem in situations with limited amounts of and/or poor-quality data. For each problem setting, while maintaining the computational feasibility of the Bayesian optimal estimation, we design models having enough complexity for sufficiently representing data variations and appropriate constraints for regularizing the limitations of the data. In addition, thanks to the fully Bayesian treatment, all the methods proposed in this thesis do not require any parameter tuning, which is a favorable property in practice.

In Chapter 2, we derived Bayesian optimal estimation functions for some evaluation criteria. These functions are based on the posterior distribution of the unobservable target variables given the observed variables. We also derived efficient computational algorithms for them as analytic tools that were used throughout this thesis.

In Chapter 3, we tackled the novel problem of estimating traffic flows from poor-quality web-camera images without any labeled training data. We devised algorithms for estimating the traffic volume and traffic velocity in this problem setting. In the traffic-volume-estimation problem, we proposed a Gaussian mixture model (GMM) whose mixture index is equated with the number of vehicles in the observation and showed that the stick-breaking process (SBP) elegantly resolves the technical challenge, that is, how to associate the mixture index with the count without any label information as to the count unlike current approaches in image analysis, which typically involve explicit object detection using labeled training images. For the traffic velocity estimation problem, we proposed a method in which the traffic velocity is estimated from observed temporal sequences of the numbers of vehicles. The proposed method does not require tracking of vehicles or any labeled data. It is based on the fact that the some proportion of vehicles in two or more consecutive observations will be the same vehicles. The proportion will increase as the traffic velocity decreases, and it directly represents the correlation between the numbers of vehicles in consecu-

tive observations. The proposed method is useful for measuring traffic velocities with low-quality, inexpensive sensors such as webcams. The experiments confirmed that our methods work with inexpensive sensors having low sampling rates, such as one observation every several seconds. Improving the feature extraction step would be worthwhile for applying our approach to other applications, such as counting crowds and cells in images, counting words in text, and counting patterns in time-series data. This is because that the features required in each application are not often trivial, and we need to use multiple features which may be effective to the estimation. Although we used a single feature in the traffic volume estimation, we can include many other features in the variational Bayes (VB) formulation and feature selection may be also possible in this case. Introducing a more hierarchical model for the hyperparameters may also improve accuracy and robustness within the Bayesian framework.

In Chapter 4, we proposed a novel super-resolution (SR) model and derived a Bayesian optimal estimation algorithm with causal and compound Gaussian Markov random field (MRF) priors for images and the VB method. We showed that we can solve the conjugate prior problem on the VB method and the exponential-order calculation cost problem of the causal and compound Gaussian MRF prior with simple Taylor approximations. Our experimental results show that our method was more accurate than other current methods. The proposed method is an SR method with a favorable model and an optimal estimation function. We believe our approach to the problem regarding the conjugate prior and the exponential-order calculation cost can be applied to many other problems, and we will attempt to do so in the future.

In Chapter 5, we addressed the problem in which a regression model is learned from sets of training data. Each set only has a single label, and only one of the training data samples in the set reflects the label. We designed an algorithm for estimating which of the training data sample in each of the sets corresponds to the label, as well as for training the regression model on the basis of hidden variable modeling and posterior inference with the VB method. Experimental results show that our approach performs better than baseline methods on an artificial dataset and on a real-world dataset. In the future, we plan to apply our approach to other learning tasks, such as classification and multiple instance learning (MIL) problems.

List of Academic Achievements

Journal (refereed)

◯ 1. Takayuki Katsuki, Tetsuro Morimura, Masato Inoue, "Traffic Velocity Estimation From Vehicle Count Sequences," IEEE Transactions on Intelligence Transportation Systems, to appear, 2016.

◯ 2. Takayuki Katsuki, Akira Torii, Masato Inoue, "Posterior Mean Super-resolution with a Causal Gaussian Markov Random Field Prior," IEEE Transactions on Image Processing 21(7), pp. 3182–3193, IEEE, 2012.

3. Tsuyoshi Idé, Takayuki Katsuki, Tetsuro Morimura, Robert Morris, "City-Wide Traffic Flow Estimation From a Limited Number of Low-Quality Cameras," IEEE Transactions on Intelligence Transportation Systems, to appear, 2016

Conference proceedings (refereed)

◯ 1. Takayuki Katsuki, Masato Inoue, "Bayesian Regression Selecting Valuable Subset from Mixed Bag Training Data," In Proceedings of the 23rd International Conference on Pattern Recognition (ICPR2016), to appear, 2016.

◯ 2. Takayuki Katsuki, Tesuro Morimura, Tsuyoshi Idé, "Unsupervised Object Counting without Object Recognition," In Proceedings of the 23rd International Conference on Pattern Recognition (ICPR2016), to appear, 2016.

◯ 3. Takayuki Katsuki, Masato Inoue, "Posterior Mean Super-Resolution with a Compound Gaussian Markov Random Field Prior," In Proceedings of the 2012 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP2012), pp. 841–844, 2012.

4. Satoshi Hara, Takayuki Katsuki, Hiroki Yanagisawa, Takafumi Ono, Ryo Okamoto, Shigeki Takeuchi, "Consistent and Efficient Nonparametric Different-Feature Selection", In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS2017), to appear, 2017.

5. Daisuke Sato, Tetsuro Morimura, Takayuki Katsuki, Yosuke Toyota, Tsuneo Kato, Hironobu Takagi, “Automated Help System for Novice Older Users from Touchscreen Gestures,” In Proceedings of the 23rd International Conference on Pattern Recognition (ICPR2016), to appear, 2016.
6. Kumiko Maeda, Tetsuro Morimura, Takayuki Katsuki, Masayoshi Teraguchi, “Frugal signal control using low resolution web-camera and traffic flow estimation,” In Proceedings of the 2014 Winter Simulation Conference, pp. 2082–2091, 2014.
7. Takayuki Osogami, Takayuki Katsuki, “A Hierarchical Bayesian Choice Model with Visibility,” In Proceedings of the 22nd International Conference on Pattern Recognition (ICPR2014), pp. 3618–3623, 2014.
8. Vikas Joshi, Nithya Rajamani, Takayuki Katsuki, Naveen Prathapaneni, LV Subramaniam, “Information fusion based learning for frugal traffic state sensing,” In Proceedings of the 23rd international joint conference on Artificial Intelligence (IJCAI2013), pp. 2826–2832, 2013.
9. Tsuyoshi Idé, Takayuki Katsuki, Tetsuro Morimura, Robert Morris, “Monitoring Entire-City Traffic using Low-Resolution Web Cameras,” In Proceedings of the 20th ITS World Congress, #3143, 2013.

Domestic workshops

1. 勝木 孝行, 森村 哲郎, “低フレームレート時系列画像からの Bayes 交通速度推定,” 第 16 回情報論的学習理論ワークショップ (IBIS2013), 2013.
2. 飯田紘士，勝木孝行，恐神貴行，中川 裕志, “ベイズ推定を用いた指数忘却型自己回帰モデルによるトレンド, 季節性を含むデータの予測,” 第 92 回数理モデル化と問題解決 (MPS) 研究発表会, 2013.
3. 勝木孝行, 森村哲郎, 井手 剛, “低画質な定点画像からの教師なし車両台数推定,” 第 15 回情報論的学習理論ワークショップ (IBIS2012), 2012.
4. 勝木 孝行, 井上 真郷, “混合モデルとしての複層 Gauss-Markov 確率場による画像の修復と領域分割,” 電子情報通信学会技術研究報告, 111(275), IBISML2011-75, pp. 223–230, 2011.
5. 勝木 孝行, 鳥居 英, 井上 真郷, “複層 Markov 確率場と線形劣化変換に対する Bayes 超解像,” 電子情報通信学会技術研究報告, 110(83), NC2010-10, pp. 63–68, 2010.

Acknowledgments

本研究を進め、まとめるにあたり、指導教員である井上真郷教授に心から感謝致します。先生には学部 3 年生から本日までの長きに渡り、また、研究から広い意味での学業、私生活に至るまで、多大なるご指導とご支援を頂きました。私が今技術に携わることを楽しいと感じ、生涯の仕事とすべく活力を持って取り組めるのは先生のご指導あってのことです。本当にありがとうございます。

本研究を進める上で村田昇教授に深く感謝しております。卒業論文、修士論文の中間発表等において、幾度となく的確なご指導とご意見を賜りました。副査もお受け頂き、本当にありがとうございます。

副査をお受け頂き、また、学部時代から授業等においてたくさんのご指導を賜った内田健康教授、渡邊亮教授に大変感謝致します。

IBM Thomas J. Watson Research Center の井手剛博士に心から感謝致します。私が右も左もわからず社会に出た折から今日に至るまで、仕事への取り組み方、研究、論文の書き方等のあらゆる面で多大なるご指導とご支援を頂きました。学位取得を目指すにあたっても背中を押して頂き、本当に感謝しております。ありがとうございます。

IBM 東京基礎研究所の森村哲郎博士に感謝致します。日々の業務や研究において常にご指導、ご支援を賜り、また、技術的なご相談を頻繁にさせて頂き大変感謝しております。ありがとうございます。

井上真郷研究室の皆様に感謝致します。直接関わりの深い先輩、同期、後輩の皆様には言葉では表しきれない感謝を感じています。皆様のおかげで楽しく研究を続けることができました。先輩である鳥居英氏には卒業後もご意見を頂き、また研究を引き継がせて頂く等、本研究に関して大変丁寧なご指導を頂き非常に感謝しております。また、社会人として博士課程で戻ってきた際も、若い世代の方々に暖かく迎えて頂けたことに本当に感謝しています。

IBM 東京基礎研究所の皆様に感謝致します。日々快適に研究を行う環境を頂き、議論やご指導、ご支援をありがとうございます。学位取得を目指すにあたっても、ご理解とサポートをありがとうございました。

早稲田大学先進理工学研究科電気・情報生命専攻の先生方に感謝致します。基礎的な内容から産業応用に至るまで幅広くカバーされたカリキュラムの中で教育を受けることができ大変幸せでした。

IBM 東京基礎研究所の渡辺日出雄博士に深く感謝致します。昨今のデータ解析への需要の高まりにより業務が非常に忙しい中、学位取得に向けご理解とご支援、ご指導を頂き本当にありがとうございます。

最後に、常に私を支えてくれた家族に感謝します。日々研究や仕事に打ち込む中で家族と過ごすふとしたひとときが心の拠り所でした。いつも本当に、ありがとうございます。

References

[1] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers, "Big data: The next frontier for innovation, competition, and productivity," 2011.

[2] H. Chen, R. H. Chiang, and V. C. Storey, "Business intelligence and analytics: From big data to big impact." *MIS quarterly*, vol. 36, no. 4, pp. 1165–1188, 2012.

[3] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[4] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[5] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: real-time event detection by social sensors," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 851–860.

[6] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *IEEE Communications magazine*, vol. 48, no. 9, pp. 140–150, 2010.

[7] W. Edwards, L. D. Phillips, W. L. Hays, and B. C. Goodman, "Probabilistic information processing systems: Design and evaluation," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 3, pp. 248–265, 1968.

[8] T. Winograd, "Understanding natural language," *Cognitive psychology*, vol. 3, no. 1, pp. 1–191, 1972.

[9] C. D. Manning and H. Schütze, *Foundations of statistical natural language processing*. MIT Press, 1999, vol. 999.

[10] D. A. Forsyth and J. Ponce, *Computer vision: a modern approach*. Prentice Hall Professional Technical Reference, 2002.

[11] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on Computer vision*, vol. 2. Ieee, 1999, pp. 1150–1157.

[12] D. J. MacKay, *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[13] C. M. Bishop, "Pattern recognition and machine learning," *Machine Learning*, vol. 128, 2006.

[14] N. A. Gershenfeld, *The nature of mathematical modeling*. Cambridge university press, 1999.

[15] R. Aris, *Mathematical modelling techniques*. Courier Corporation, 2012.

[16] P. F. Brown, P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, "Class-based n-gram models of natural language," *Computational linguistics*, vol. 18, no. 4, pp. 467–479, 1992.

[17] C. Lin and E. Hovy, "Automatic evaluation of summaries using n-gram co-occurrence statistics," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for Computational Linguistics, 2003, pp. 71–78.

[18] W. B. Cavnar, J. M. Trenkle *et al.*, "N-gram-based text categorization," *Ann Arbor MI*, vol. 48113, no. 2, pp. 161–175, 1994.

[19] G. Doddington, "Automatic evaluation of machine translation quality using n-gram co-occurrence statistics," in *Proceedings of the second international conference on Human Language Technology Research*. Morgan Kaufmann Publishers Inc., 2002, pp. 138–145.

[20] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[21] T. Tuytelaars and K. Mikolajczyk, "Local invariant feature detectors: a survey," *Foundations and trends® in computer graphics and vision*, vol. 3, no. 3, pp. 177–280, 2008.

[22] A. Björck, *Numerical methods for least squares problems*. SIAM, 1996.

[23] C. L. Lawson and R. J. Hanson, *Solving least squares problems*. SIAM, 1995, vol. 15.

[24] V. Vapnik, *The nature of statistical learning theory*. Springer Science & Business Media, 2000.

[25] T. M. Mitchell, "Machine learning," *New York*, 1997.

[26] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics Springer, Berlin, 2001, vol. 1.

[27] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural computation*, vol. 4, no. 1, pp. 1–58, 1992.

[28] A. G. Ivakhnenko and V. G. Lapa, *Cybernetic predicting devices*. CCM Information Corporation, 1965.

[29] D. H. Ballard, "Modular learning in neural networks," in *Proceedings of the sixth National conference on Artificial intelligence–Volume 1.* AAAI Press, 1987, pp. 279–284.

[30] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[31] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[32] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[33] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning.* MIT Press, 2016, http://www.deeplearningbook.org.

[34] K. P. Murphy, *Machine learning: a probabilistic perspective.* MIT press, 2012.

[35] T. Katasuki, T. Morimura, and T. Idé, "Bayesian unsupervised vehicle counting," IBM Research Report, Armonk, NY, USA, RT0951, Tech. Rep., 2013.

[36] T. Katsuki, T. Morimura, and T. Idé, "Unsupervised object counting without object recognition," in *Proceedings of the 23rd International Conference on Pattern Recognition (ICPR 2016)*, 2016, pp. — (to appear).

[37] T. Katsuki, T. Morimura, and M. Inoue, "Traffic velocity estimation from vehicle count sequences," *Transactions on Intelligent Transportation Systems*, pp. — (to appear), 2016.

[38] T. Katsuki, A. Torii, and M. Inoue, "Posterior-mean super-resolution with a causal gaussian markov random field prior," *IEEE Transactions on Image Processing*, vol. 21, no. 7, pp. 3182–3193, 2012.

[39] T. Katsuki and M. Inoue, "Posterior mean super-resolution with a compound gaussian markov random field prior," in *Proceedings of the 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2012).* IEEE, 2012, pp. 841–844.

[40] ——, "Bayesian regression selecting valuable subset from mixed bag training data," in *Proceedings of the 23rd International Conference on Pattern Recognition (ICPR 2016)*, 2016, pp. — (to appear).

[41] H. Raiffa, "Applied statistical decision theory," 1974.

[42] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The journal of chemical physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[43] W. K. Hastings, "Monte carlo sampling methods using markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[44] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on pattern analysis and*

machine intelligence, no. 6, pp. 721–741, 1984.

[45] M. A. Tanner and W. H. Wong, "The calculation of posterior distributions by data augmentation," *Journal of the American statistical Association*, vol. 82, no. 398, pp. 528–540, 1987.

[46] A. E. Gelfand and A. F. Smith, "Sampling-based approaches to calculating marginal densities," *Journal of the American statistical association*, vol. 85, no. 410, pp. 398–409, 1990.

[47] W. R. Gilks, *Markov chain monte carlo*. Wiley Online Library, 2005.

[48] C. Robert and G. Casella, "A short history of markov chain monte carlo: subjective recollections from incomplete data," *Statistical Science*, pp. 102–115, 2011.

[49] ——, *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

[50] R. M. Neal, "Slice sampling," *Annals of statistics*, pp. 705–741, 2003.

[51] C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan, "An introduction to mcmc for machine learning," *Machine learning*, vol. 50, no. 1-2, pp. 5–43, 2003.

[52] H. Attias and L. W. Ar, "Inferring parameters and structure of latent variable models by variational Bayes," in *Proc. of the Fifteenth Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers, 1999, pp. 21–30.

[53] D. G. Tzikas, A. C. Likas, and N. P. Galatsanos, "The variational approximation for bayesian inference," *IEEE Signal Processing Magazine*, vol. 25, no. 6, pp. 131–146, 2008.

[54] T. Hosino, K. Watanabe, and S. Watanabe, "Stochastic complexity of variational bayesian hidden markov models," in *Proceedings of the 2005 IEEE International Joint Conference on Neural Networks*, vol. 2. IEEE, 2005, pp. 1114–1119.

[55] K. Watanabe and S. Watanabe, "Stochastic complexities of gaussian mixtures in variational bayesian approximation," *Journal of Machine Learning Research*, vol. 7, no. Apr, pp. 625–644, 2006.

[56] S. Nakajima and S. Watanabe, "Variational bayes solution of linear neural networks and its generalization performance," *Neural Computation*, vol. 19, no. 4, pp. 1112–1153, 2007.

[57] K. Watanabe and S. Watanabe, "Stochastic complexities of general mixture models in variational bayesian learning," *Neural Networks*, vol. 20, no. 2, pp. 210–219, 2007.

[58] K. Watanabe, M. Shiga, and S. Watanabe, "Upper bound for variational free energy of bayesian networks," *Machine Learning*, vol. 75, no. 2, pp. 199–215, 2009.

[59] S. Nakajima, M. Sugiyama, S. D. Babacan, and R. Tomioka, "Global analytic solution of fully-observed variational bayesian matrix factorization," *Journal of Machine Learning Research*, vol. 14, no. Jan, pp. 1–37, 2013.

[60] C. Bishop, D. Spiegelhalter, and J. Winn, "VIBES: A variational inference engine for Bayesian networks," in *Advances in Neural Information Processing Systems 15*. MIT Press, 2003, pp. 777–784.

[61] M. H. Dunham, Y. Meng, and J. Huang, "Extensible markov model," in *Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM'04)*. IEEE, 2004, pp. 371–374.

[62] M. Wojnarski, P. Gora, M. Szczuka, H. S. Nguyen, J. Swietlicka, and D. Zeinalipour, "IEEE ICDM 2010 contest: Tomtom traffic prediction for intelligent gps navigation," in *2010 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2010, pp. 1372–1376.

[63] C. Urmson, C. Baker, J. Dolan, P. Rybski, B. Salesky, W. Whittaker, D. Ferguson, and M. Darms, "Autonomous driving in traffic: Boss and the Urban Challenge," *AI Magazine*, vol. 30, no. 2, pp. 17–28, 2009.

[64] Y. Sekine, *AI in Intelligent Vehicle Highway Systems: Papers from the 1993 Workshop*. AAAI Press, 1994.

[65] T. Morimura, T. Osogami, and T. Idé, "Solving inverse problem of markov chain with partial observations," in *Advances in Neural Information Processing Systems*, 2013, pp. 1655–1663.

[66] B. Pan, U. Demiryurek, and C. Shahabi, "Utilizing real-world transportation data for accurate traffic prediction," in *Proceedings of the IEEE 12th International Conference on Data Mining (ICDM2012)*. IEEE, 2012, pp. 595–604.

[67] H. Kriegel, M. Renz, M. Schubert, and A. Zuefle, "Statistical density prediction in traffic networks," in *Proc. of the 8th SIAM international conference on Data Mining*. SIAM, 2008, pp. 692–703.

[68] V. Coric, N. Djuric, and S. Vucetic, "Frugal traffic monitoring with autonomous participatory sensing," in *Proc. of the 14th SIAM International Conference on Data Mining*. SIAM, 2014.

[69] T. Idé and S. Kato, "Travel-time prediction using gaussian process regression: A trajectory-based approach." in *Proc. of the 9th SIAM international conference on Data Mining*, 2009, pp. 1183–1194.

[70] T. Nakata and J. Takeuchi, "Mining traffic data from probe-car system for travel time prediction," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 817–822.

[71] Y. Jin, J. Dai, and C. Lu, "Spatial-temporal data mining in traffic incident detection," in *Proc. SIAM DM 2006 Workshop on Spatial Data Mining*, vol. 5.

SIAM, 2006.

[72] S. Chawla, Y. Zheng, and J. Hu, "Inferring the root cause in road traffic anomalies," in *Proceedings of the IEEE 12th International Conference on Data Mining (ICDM2012)*. IEEE, 2012, pp. 141–150.

[73] J. Lan, C. Long, R. C. Wong, Y. Chen, Y. Fu, D. Guo, S. Liu, Y. Ge, Y. Zhou, and J. Li, "A new framework for traffic anomaly detection," in *Proc. of the 14th SIAM International Conference on Data Mining*. SIAM, 2014.

[74] N. Buch, S. Velastin, and J. Orwell, "A review of computer vision techniques for the analysis of urban traffic," *IEEE Trans. on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 920–939, 2011.

[75] AccessKenya.com, "http://traffic.accesskenya.com/."

[76] S. Santini, "Analysis of traffic flow in urban areas using web cameras," in *Fifth IEEE Workshop on Applications of Computer Vision*, 2000, pp. 140–145.

[77] T. Idé, T. Katsuki, T. Morimura, and R. Morris, "Monitoring entire-city traffic using low-resolution web cameras," in *Proceedings of the 20th ITS World Congress, Tokyo*, 2013.

[78] ——, "City-wide traffic flow estimation from a limited number of low-quality cameras," *IEEE Transactions on Intelligent Transportation Systems*, pp. — (to appear), 2016.

[79] V. Joshi, N. Rajamani, K. Takayuki, N. Prathapaneni, and L. V. Subramaniam, "Information fusion based learning for frugal traffic state sensing," in *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*. AAAI Press, 2013, pp. 2826–2832.

[80] J. Choi, K. Sung, and Y. Yang, "Multiple vehicles detection and tracking based on scale-invariant feature transform," in *Proc. IEEE Intl Conf. Intelligent Transportation Systems*, 2007, pp. 528–533.

[81] A. Kembhavi, D. Harwood, and L. Davis, "Vehicle detection using partial least squares," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 33, no. 6, pp. 1250–1265, 2011.

[82] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik, "A real-time computer vision system for measuring traffic parameters," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 97)*, 1997, pp. 495–501.

[83] Z. Kim and J. Malik, "Fast vehicle detection with probabilistic feature grouping and its application to vehicle tracking," in *Proc. IEEE Intl. Conf. on Computer Vision*, vol. 1, 2003, pp. 524–531.

[84] K. Robert, "Video-based traffic monitoring at day and night vehicle features detection tracking," in *Proc. IEEE Intl. Conf. on Intelligent Transportation*

Systems, 2009, pp. 1–6.

[85] Y. Chen, B. Wu, H. Huang, and C. Fan, "A real-time vision system for nighttime vehicle detection and traffic surveillance," *IEEE Trans. on Industrial Electronics*, vol. 58, no. 5, pp. 2030–2044, 2011.

[86] S. Hu, J. Wu, and L. Xu, "Real-time traffic congestion detection based on video analysis," *Journal of Information and Computational Science*, vol. 9, no. 10, pp. 2907–2914, 2012.

[87] X. Yu, L. Duan, and Q. Tian, "Highway traffic information extraction from skycam mpeg video," in *Proc. IEEE Intl. Conf. on Intelligent Transportation Systems*, 2002, pp. 37–42.

[88] M. Haag and H.-H. Nagel, "Combination of edge element and optical flow estimates for 3d-model-based vehicle tracking in traffic image sequences," *International Journal of Computer Vision*, vol. 35, no. 3, pp. 295–319, 1999.

[89] S. Indu, M. Gupta, and A. Bhattacharyya, "Vehicle tracking and speed estimation using optical flow method," *International Journal of Engineering Science and Technology (IJEST)*, vol. 3, no. 1, 2011.

[90] J. Lan, J. Li, G. Hu, B. Ran, and L. Wang, "Vehicle speed measurement based on gray constraint optical flow algorithm," *Optik-International Journal for Light and Electron Optics*, vol. 125, no. 1, pp. 289–295, 2014.

[91] J. C. Herrera, D. B. Work, R. Herring, X. J. Ban, Q. Jacobson, and A. M. Bayen, "Evaluation of traffic data obtained via gps-enabled mobile phones: The mobile century field experiment," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 4, pp. 568–583, 2010.

[92] R. Zito, G. d'Este, and M. A. Taylor, "Global positioning systems in the time domain: How useful a tool for intelligent vehicle-highway systems?" *Transportation Research Part C: Emerging Technologies*, vol. 3, no. 4, pp. 193–209, 1995.

[93] T. Schoepflin and D. Dailey, "Correlation technique for estimating traffic speed from cameras," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1855, pp. 66–73, 2003.

[94] ——, "Cross-correlation tracking technique for extracting speed from cameras under adverse conditions," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1867, pp. 36–45, 2004.

[95] Y. Cho and J. Rice, "Estimating velocity fields on a freeway from low-resolution videos," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 463–469, 2006.

[96] S. Pumrin and D. Dailey, "Roadside camera motion detection for automated speed measurement," in *Proceedings of The IEEE 5th International Conference*

on Intelligent Transportation Systems. IEEE, 2002, pp. 147–151.

[97] D. J. Dailey and L. Li, "An algorithm to estimate vehicle speed using uncalibrated cameras," *Transportation Research Record 1719*, pp. 27–32, 2000.

[98] D. J. Dailey, F. W. Cathey, and S. Pumrin, "An algorithm to estimate mean traffic speed using uncalibrated cameras," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 2, pp. 98–107, 2000.

[99] T. N. Schoepflin and D. J. Dailey, "Dynamic camera calibration of roadside traffic management cameras for vehicle speed estimation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 4, no. 2, pp. 90–98, 2003.

[100] Y. Malinovskiy, Y. Wu, and Y. Wang, "Video-based vehicle detection and tracking using spatiotemporal maps," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2121, pp. 81–89, 2009.

[101] B. D. Greenshields, J. Thompson, H. Dickinson, and R. Swinton, "The photographic method of studying traffic behavior," in *Highway Research Board Proceedings*, vol. 13, 1934.

[102] B. Greenshields, W. Channing, H. Miller *et al.*, "A study of traffic capacity," in *Highway research board proceedings*. National Research Council (USA), Highway Research Board, 1935.

[103] H. Greenberg, "An analysis of traffic flow," *Operations research*, vol. 7, no. 1, pp. 79–85, 1959.

[104] R. Underwood, "Speed, volume, and density relationship: quality and theory of traffic flow," *Yale Bureau of Highway Traffic*, pp. 141–188, 1961.

[105] J. Del Castillo and F. Benitez, "On the functional form of the speed-density relationship–I: general theory," *Transportation Research Part B: Methodological*, vol. 29, no. 5, pp. 373–389, 1995.

[106] ——, "On the functional form of the speed-density relationship–II: empirical investigation," *Transportation Research Part B: Methodological*, vol. 29, no. 5, pp. 391–406, 1995.

[107] J. S. Drake, J. L. Schofer, and A. D. May Jr, "A statistical analysis of speed-density hypotheses. in vehicular traffic science," *Highway Research Record*, no. 154, 1967.

[108] H. Wang, D. Ni, Q. Chen, and J. Li, "Stochastic modeling of the equilibrium speed–density relationship," *Journal of Advanced Transportation*, vol. 47, no. 1, pp. 126–150, 2013.

[109] H. Wang, J. Li, Q. Chen, and D. Ni, "Speed-density relationship: From deterministic to stochastic," in *Transportation Research Board 88th Annual Meeting*, vol. 10, 2009.

[110] N. Radjou, J. Prabhu, and S. Ahuja, *Jugaad innovation: Think frugal, be flex-*

ible, generate breakthrough growth. John Wiley & Sons, 2012.

[111] N. Radjou, "Creative problem-solving in the face of extreme limits," https://www.ted.com/talks/navi_radjou_creative_problem_solving_in_the_face_of_extreme_limits?language=en, 2014.

[112] J. Sethuraman, "A constructive definition of Dirichlet priors," *Statistica Sinica*, vol. 4, pp. 639–650, 1994.

[113] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62–66, jan. 1979.

[114] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010, pp. 2963–2970.

[115] I. Saleemi and M. Shah, "Multiframe many–many point correspondence for vehicle tracking in high density wide area aerial videos," *International Journal of Computer Vision*, pp. 1–22, 2013.

[116] D. M. Blei and M. I. Jordan, "Variational inference for Dirichlet process mixtures," *Bayesian Analysis*, vol. 1, pp. 121–144, 2005.

[117] M. West, "Hyperparameter estimation in Dirichlet process mixture models," *Duke University Technical Report*, vol. 92-A03, 1992.

[118] C. E. Rasmussen, "The infinite gaussian mixture model," in *Advances in Neural Information Processing Systems*, vol. 12, 1999, pp. 554–560.

[119] T. S. Ferguson, "A Bayesian Analysis of Some Nonparametric Problems," *The Annals of Statistics*, vol. 1, no. 2, pp. 209–230, 1973. [Online]. Available: http://dx.doi.org/10.2307/2958008

[120] S. Z. Li, Z. Zhang, H. Shum, and H. Zhang, "Floatboost learning for classification," in *Advances in Neural Information Processing Systems*, vol. 15, 2002.

[121] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR2001)*, vol. 1. IEEE, 2001, pp. 511–518.

[122] P. Negri, X. Clady, S. M. Hanif, and L. Prevost, "A cascade of boosted generative and discriminative classifiers for vehicle detection," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, p. 136, 2008.

[123] R. R. Wilcox, *Introduction to robust estimation and hypothesis testing.* Academic Press, 2012.

[124] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, jun 2010.

[125] ——, "The PASCAL VOC2012 Results," http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012.

[126] C. Papageorgiou and T. Poggio, "A trainable system for object detection," *International Journal of Computer Vision*, vol. 38, no. 1, pp. 15–33, 2000.

[127] J. W. Tukey, "Exploratory data analysis," 1977.

[128] United States Department of Transportation, "Ngsim – next generation simulation," http://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm.

[129] M. Nixon and A. S. Aguado, *Feature extraction & image processing*. Access Online via Elsevier, 2008.

[130] K. Fukunaga, *Introduction to statistical pattern recognition*. Academic press, 1990.

[131] I. Porteous, A. T. Ihler, P. Smyth, and M. Welling, "Gibbs sampling for (coupled) infinite mixture models in the stick breaking representation," in *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2006.

[132] K. Kurihara, M. Welling, and Y. W. Teh, "Collapsed variational dirichlet process mixture models." in *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, vol. 7, 2007, pp. 2796–2801.

[133] W. Yih, J. Goodman, L. Vanderwende, and H. Suzuki, "Multi-document summarization by maximizing informative content-words," in *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, 2007, pp. 1776–1782.

[134] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth, "The author-topic model for authors and documents," in *Proceedings of the 20th conference on Uncertainty in artificial intelligence*. AUAI Press, 2004, pp. 487–494.

[135] J. D. Mcauliffe and D. M. Blei, "Supervised topic models," in *Advances in Neural Information Processing Systems*, 2008, pp. 121–128.

[136] T. Ma, I. Sato, and H. Nakagawa, "The hybrid nested/hierarchical dirichlet process and its application to topic modeling with word differentiation," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[137] I. Batal, H. Valizadegan, G. F. Cooper, and M. Hauskrecht, "A temporal pattern mining approach for classifying electronic health record data," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 4, no. 4, p. 63, 2013.

[138] Z. Liu and M. Hauskrecht, "A regularized linear dynamical system framework for multivariate time series analysis," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 2015. NIH Public Access, 2015, p. 1798.

[139] G. Heyrani Nobari and T. Chua, "User intent identification from online discussions using a joint aspect-action topic model," in *Twenty-Eighth AAAI Confer-*

ence on Artificial Intelligence, 2014.

[140] R. Tsai and T. Huang, "Multiframe image restoration and registration," in *Advances in computer vision and image processing*, vol. 1, no. 2, 1984, pp. 317–339.

[141] M. E. Tipping and C. M. Bishop, "Bayesian image super resolution," in *Advances in Neural Information Processing Systems*, 2003, pp. 1279–1286.

[142] S. D. Babacan, R. Molina, and A. K. Katsaggelos, "Variational bayesian super resolution," *IEEE Transactions on Image Processing*, vol. 20, no. 4, pp. 984–999, 2011.

[143] A. Kanemura, S. Maeda, and S. Ishii, "Hyperparameter estimation in bayesian image superresolution with a compound markov random field prior," in *2007 IEEE Workshop on Machine Learning for Signal Processing*. IEEE, 2007, pp. 181–186.

[144] ——, "Superresolution with compound markov random fields via the variational em algorithm," *Neural Networks*, vol. 22, no. 7, pp. 1025–1034, 2009.

[145] L. C. Pickup, D. P. Capel, S. J. Roberts, and A. Zisserman, "Bayesian image super-resolution, continued," in *Advances in Neural Information Processing Systems*, 2006, pp. 1089–1096.

[146] R. C. Hardie, K. J. Barnard, and E. E. Armstrong, "Joint map registration and high-resolution image estimation using a sequence of undersampled images," *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1621–1633, 1997.

[147] S. Villena, M. Vega, S. D. Babacan, R. Molina, and A. K. Katsaggelos, "Image prior combination in super-resolution image registration & reconstruction," in *2010 IEEE International Workshop on Machine Learning for Signal Processing*. IEEE, 2010, pp. 355–360.

[148] R. Molina, J. Mateos, A. K. Katsaggelos, and M. Vega, "Bayesian multichannel image restoration using compound gauss-markov random fields," *IEEE Transactions on Image Processing*, vol. 12, no. 12, pp. 1642–1654, 2003.

[149] P. Vandewalle, L. Sbaiz, J. Vandewalle, and M. Vetterli, "Super-resolution from unregistered and totally aliased signals using subspace methods," *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3687–3703, 2007.

[150] S. Borman and R. Stevenson, "Spatial resolution enhancement of low-resolution image sequences-a comprehensive review with directions for future research," *Lab. Image and Signal Analysis, University of Notre Dame, Tech. Rep*, 1998.

[151] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, "Advances and challenges in super-resolution," *International Journal of Imaging Systems and Technology*, vol. 14, no. 2, pp. 47–57, 2004.

[152] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruc-

tion: a technical overview," *IEEE signal processing magazine*, vol. 20, no. 3, pp. 21–36, 2003.

[153] M. Ng, T. Chan, M. G. Kang, and P. Milanfar, "Special issue on super-resolution imaging: Analysis, algorithms, and applications," *EURASIP Journal on Advances in Signal Processing*, vol. 2006, no. 1, pp. 1–2, 2006.

[154] A. K. Katsaggelos, R. Molina, and J. Mateos, "Super resolution of images and video," *Synthesis Lectures on Image, Video, and Multimedia Processing*, vol. 1, no. 1, pp. 1–134, 2007.

[155] P. Milanfar, *Super-resolution imaging*. CRC press, 2010.

[156] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 721–741, 1984.

[157] R. Chellappa and E. A.K. Jain, *Markov random fields: Theory and application*. CA: Academic Press, 1991.

[158] F. Jeng and J. W. Woods, "Compound gauss-markov random fields for image estimation," *IEEE Transactions on Signal Processing*, vol. 39, no. 3, pp. 683–697, 1991.

[159] V. Mnih, G. E. Hinton *et al.*, "Generating more realistic images using gated mrf's," in *Advances in Neural Information Processing Systems*, 2010, pp. 2002–2010.

[160] R. Molina, M. Vega, J. Abad, and A. K. Katsaggelos, "Parameter estimation in bayesian high-resolution image reconstruction with multisensors," *IEEE Transactions on Image Processing*, vol. 12, no. 12, pp. 1655–1667, 2003.

[161] F. Jeng and J. W. Woods, "Simulated annealing in compound gaussian random fields [image processing]," *IEEE Transactions on Information Theory*, vol. 36, no. 1, pp. 94–107, 1990.

[162] A. K. Jain, R. P. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4–37, 2000.

[163] H. Jagadish, J. Gehrke, A. Labrinidis, Y. Papakonstantinou, J. M. Patel, R. Ramakrishnan, and C. Shahabi, "Big data and its technical challenges," *Communications of the ACM*, vol. 57, no. 7, pp. 86–94, 2014.

[164] A. O'Hagan, "On outlier rejection phenomena in bayes inference," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 358–367, 1979.

[165] M. West, "Outlier models and prior distributions in bayesian linear regression," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 431–439, 1984.

[166] M. Feng, L. Y. Loy, K. Sim, C. Phua, F. Zhang, and C. Guan, "Artifact correc-

tion with robust statistics for non-stationary intracranial pressure signal monitoring," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR)*. IEEE, 2012, pp. 557–560.

[167] V. S. Sheng, F. Provost, and P. G. Ipeirotis, "Get another label? improving data quality and data mining using multiple, noisy labelers," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 614–622.

[168] J. Whitehill, T. Wu, J. Bergsma, J. R. Movellan, and P. L. Ruvolo, "Whose vote should count more: Optimal integration of labels from labelers of unknown expertise," in *Advances in Neural Information Processing Systems*, 2009, pp. 2035–2043.

[169] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy, "Learning from crowds," *The Journal of Machine Learning Research*, vol. 11, pp. 1297–1322, 2010.

[170] K. L. Lange, R. J. Little, and J. M. Taylor, "Robust statistical modeling using the t distribution," *Journal of the American Statistical Association*, vol. 84, no. 408, pp. 881–896, 1989.

[171] J. Geweke, "Bayesian treatment of the independent student-t linear model," *Journal of Applied Econometrics*, vol. 8, no. S1, pp. S19–S40, 1993.

[172] C. Liu and D. B. Rubin, "Ml estimation of the t distribution using em and its extensions, ecm and ecme," *Statistica Sinica*, pp. 19–39, 1995.

[173] G. McLachlan and T. Krishnan, *The EM algorithm and extensions*. John Wiley & Sons, 2007, vol. 382.

[174] S. C. Narula and J. F. Wellington, "The minimum sum of absolute errors regression: A state of the art survey," *International Statistical Review/Revue Internationale de Statistique*, pp. 317–326, 1982.

[175] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial intelligence*, vol. 89, no. 1, pp. 31–71, 1997.

[176] S. Ray and D. Page, "Multiple instance regression," in *Proceedings of the 18th International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc., 2001, pp. 425–432.

[177] D. R. Dooly, Q. Zhang, S. A. Goldman, and R. A. Amar, "Multiple-instance learning of real-valued data," *Journal of Machine Learning Research*, vol. 3, pp. 651–678, 2002.

[178] Z. Wang, L. Lan, and S. Vucetic, "Mixture model for multiple instance regression and applications in remote sensing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 6, pp. 2226–2237, 2012.

[179] D. J. C. Mackay, "Bayesian non-linear modelling for the prediction competition," *ASHRAE Transactions*, vol. 100, pp. 1053–1062, 1994.

[180] R. M. Neal, *Bayesian Learning for Neural Networks*. Springer, 1996.

[181] M. E. Tipping, "Sparse bayesian learning and the relevance vector machine," *The journal of machine learning research*, vol. 1, pp. 211–244, 2001.

[182] C. M. Bishop and M. E. Tipping, "Variational relevance vector machines," in *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 2000, pp. 46–53.

[183] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[184] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[185] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[186] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[187] F. Zamora-Martínez, P. Romeu, P. Botella-Rocamora, and J. Pardo, "On-line learning of indoor temperature forecasting models towards energy efficiency," *Energy and Buildings*, vol. 83, pp. 162–172, 2014.

[188] R. M. Neal, *Bayesian learning for neural networks*. Springer Science & Business Media, 2012, vol. 118.

[189] C. Williams and C. Rasmussen, "Gaussian processes for regression," in *Advances in Neural Information Processing Systems*. MIT Press, 1996.

[190] D. R. Jones, "A taxonomy of global optimization methods based on response surfaces," *Journal of global optimization*, vol. 21, no. 4, pp. 345–383, 2001.

[191] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Advances in Neural Information Processing Systems*, 2012, pp. 2951–2959.

[192] K. Chaloner and I. Verdinelli, "Bayesian experimental design: A review," *Statistical Science*, pp. 273–304, 1995.

[193] G. Kumar and V. Govindaraju, "Bayesian active learning for keyword spotting in handwritten documents," in *Proceedings of the 22nd International Conference on Pattern Recognition (ICPR2014)*. IEEE, 2014, pp. 2041–2046.

[194] W. R. Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," *Biometrika*, vol. 25, no. 3/4, pp. 285–294, 1933.

[195] M. J. Strens, "A bayesian framework for reinforcement learning," in *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*. Morgan Kaufmann Publishers Inc., 2000, pp. 943–950.

[196] A. Wilson, A. Fern, S. Ray, and P. Tadepalli, "Multi-task reinforcement learning: a hierarchical bayesian approach," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 1015–1022.

[197] J. Asmuth, L. Li, M. L. Littman, A. Nouri, and D. Wingate, "A bayesian sampling approach to exploration in reinforcement learning," in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2009, pp. 19–26.