Traffic Prediction During Large-scale Events Based on Pattern-aware Regression

TAKAFUMI OKUKUBO^{1,2,a)} YOSHIKI BANDO^{2,b)} MASAKI ONISHI^{2,c)}

Received: April 13, 2021, Accepted: October 8, 2021

Abstract: This paper presents a statistical method combined with a neural network for efficient traffic prediction from a limited amount of training data. The traffic prediction during a large-scale event is essential to maintain the safety of event participants. The conventional methods for predicting traffic time series, however, cannot be utilized because the rare nature of the large-scale events prevents us from preparing a sufficient amount of training data. To efficiently train traffic prediction from a limited amount of training data, we propose a pattern-aware regression method that reduces the number of model parameters by interpreting traffic data as a weighted sum of latent behavior patterns. The proposed method trains a neural regression model to predict the weights of these patterns from the event information instead of directly predicting the traffic time series. The behavior patterns are jointly estimated during the training in a Bayesian manner to avoid overfitting. We performed experiments with foot traffic data recorded at a real soccer stadium and show that the proposed method outperforms the conventional direct regression methods. We also demonstrate an application of our method for predicting travel time from the stadium to the nearest highway interchange, which outperforms a popular commercial service.

Keywords: traffic prediction, large-scale event, non-negative matrix factorization, deep Bayesian learning

1. Introduction

Intelligent transportation systems have been developed to keep people safe and comfortable in the traffic of a city [1], [2], [3]. Predicting traffic congestion has been considered as an essential function for the intelligent transportation system [3]. One of the causes of traffic congestion are large-scale events such as sports games and entertainment events. Such a large-scale event often creates sudden and excessive traffic congestion of people and automobiles after the event is over because a large number of visitors leave the event site at the same time. This kind of congestion could result in serious crowd accidents in the worst-case scenario, such as the human crush at Akashi, Japan in July 2001. The traffic congestion could also decrease a visitor's satisfaction with the event and make people living around the event site uncomfortable. To avoid these traffic troubles, various methods have been proposed for forecasting and controlling excessive traffic congestion at large-scale events [4], [5].

A problem with conventional traffic prediction methods is that their prediction fails with a limited amount of training data because they rely on long-term past traffic histories as the training data [6], [7], [8]. Since large-scale events where a large number of people visit are not often held, it is difficult to obtain a large amount of traffic data enough to train the conventional methods.

For example, there are only a few major games at a soccer stadium in a year, when a large number of spectators visit. Several methods predict sudden congestion from not only the traffic histories but also social media posts or the traveling history of the visitors [4], [9]. Such external data are not easy to obtain in terms of cost and access rights for event organizers.

A promising approach to predict traffic congestion from a limited amount of training data is to assume that observed traffic is composed of a few behavior patterns of people. Some people, for example, leave an event site before the event is over, while others stay longer at the site. The proportion of these behavior patterns could be considered to depend on the content of the event. The observed traffic, thus, could be considered as a sum of such latent behavior patterns weighted by the number of people corresponding to each group. By predicting the weights of the behavior patterns instead of directly predicting the raw time series, we can significantly reduce the number of predicted variables, which prevents the regression model from overfitting.

In this paper, we propose a pattern-aware regression method that can make great use of a limited amount of traffic data as shown in **Fig. 1**. Our method predicts a time series of foot traffic at a specific location of a large-scale event by taking as input the event information (e.g., starting time, weather, the number of spectators). We interpret the traffic time series as a sum of latent behavior patterns weighted by their activations based on non-negative matrix factorization (NMF) [10]. The proposed method trains a non-linear regression model to predict the activations while jointly finding the latent behavior patterns from the training data. We formulate this method as a Bayesian genera-

¹ Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba, Ibaraki 305–8577, Japan

² National Institute of Advanced Industrial Science and Technology (AIST), Koto, Tokyo 135–0064, Japan

a) taka.okukubo@gmail.com

b) y.bando@aist.go.jp

c) onishi@ni.aist.go.jp

Fig. 1 Overview of the proposed traffic prediction method.

tive model to prevent the model parameters from overfitting. To demonstrate an application of the foot traffic prediction, we further predict a travel time from the event site to another specific location by using predicted foot traffic. The prediction performance can be improved by using the training data of not only travel times but also the foot traffic.

We previously published a workshop paper [11] that applies the low-rank approximation by NMF into traffic prediction with a small amount of training data. We proposed a probabilistic generative model suitable for the traffic prediction task. Because in our workshop paper only a neural regression model was evaluated, in this paper, we perform an experimental comparison of several regression models for predicting the activations of behavior patterns. We also demonstrate the application of our method for efficiently predicting a travel time from the event site to a specific highway interchange. The experimental evaluation with foot traffic data recorded at a stadium shows that the proposed method predicts the traffic better than conventional direct methods. The travel time prediction was, on the other hand, evaluated with moving trajectories of automobiles passing a specific road around the stadium. We show that our method outperformed naïve regression methods that directly predicted the travel time from the event information. We also report that our method worked more precisely than a popular commercial service Google Maps. In summary, we list up the contributions in this paper as follows:

- performing an experimental comparison of regression models for our pattern-aware foot-traffic prediction including neural networks, random forests, and generalized additive models,
- experimentally showing that our Bayesian method was robust against the hyperparameter selection compared with the maximum likelihood method,
- extending our foot-traffic prediction to a method to predict travel times,
- evaluating our method with real data recorded at a soccer stadium, and
- experimentally showing that our travel time prediction outperformed the performance of Google Maps.

2. Related Work

This section reviews existing studies of traffic prediction and non-negative matrix factorization.

2.1 Traffic Prediction

Most approaches for traffic prediction have utilized time-series regression that predicts the traffic of vehicles or people at a next time step based on its past traffic series. Existing studies have used statistical regression models such as an autoregressive model [6] and deep neural networks such as a recurrent neural network [8]. They have also utilized a non time-series regression method such as support vector machines [12] and random forests [13]. Additional information, such as weather [14] and the traffic data observed at other places [15], [16], has been known to improve the traffic prediction performance. It is also reported that such additional information improved the traffic prediction during large-scale events, which creates congestion difficult to be predicted only from a recent traffic history. Existing methods have used data describing contents of an event and movements of people during an event, such as moving histories observed with Global Positioning System (GPS) [4], search queries of a web mapping service [17], and event-related information from social media [9]. These methods have been designed to make good use of large datasets for improving the prediction performance. Conversely, they could often fail to predict a traffic series with a small amount of training data.

2.2 Non-negative Matrix Factorization

NMF can efficiently reduce the number of parameters in an unsupervised manner by approximating observed non-negative vectors with a weighted sum of a small number of non-negative latent vectors [10]. NMF has been applied for audio source separation [18], face recognition [19], document recognition [20], and social media analysis [21]. NMF has been reported to be improved by selecting a specific likelihood function suitable for target data. In audio source separation, Poisson and exponential likelihoods are known to outperform the conventional Gaussian likelihood [22]. It has been also reported that a Bayesian extension of NMF can improve the performance of NMF [23].

Some studies have combined NMF and regression models [24], [25], [26], [27]. Several traffic prediction methods utilized NMF for clustering the observations of distributed road sensors [24], [25]. Several studies integrated NMF with a regression model that predicts basis vectors or their weight (activation) vectors for clustering images [26] or predicting electricity consumption [27]. These methods are based on the minimization of mean square errors, which corresponds to the maximization of a Gaussian likelihood. The Gaussian model is inconsistent for the non-negative observations in terms of a probabilistic generative model. As in the audio applications, the suitable likelihood function for traffic prediction should be investigated.

3. Foot Traffic Prediction

In this section, we describe the proposed traffic prediction method that introduces the low-rank approximation by NMF.

3.1 Problem Specification

The problem setting of a foot traffic prediction during a large-scale event is specified as follows:

Inputs: a feature vector $\mathbf{x}^* \in \mathbb{R}^D$ of the target event

Outputs: the expected value of foot traffic series $\hat{\mathbf{y}}^* \in \mathbb{R}_+^T$,

Training data: N pairs of feature vectors $\mathbf{x}^{(n)} \in \mathbb{R}^D$, and traffic series $\mathbf{y}^{(n)} \in \mathbb{N}^T$ that are collected in advance,

where T and D represent the numbers of time frames and dimensions of the feature vector, respectively, and $\mathbb{N}$, $\mathbb{R}$, and $\mathbb{R}_+$, are sets of natural, real, and nonnegative numbers, respectively. We use the superscript (n) for training samples $n = 1, \dots, N$, $*$ for a target to be predicted, $\hat{\cdot}$ for an expected value of a random variable. In this paper, we assume that each time frame is one minute long, and we predict the number of people (foot traffic) observed during the one minute period. The feature vector consists of information related to the target event, such as the number of spectators and weather at the event site.

3.2 Model Formulation

Since the foot traffics are (non-negative) count data, we formulate foot traffic $y_t \in \mathbb{N}$ at time t with the following Poisson regression model:

$$y_t | f, \mathbf{x} \sim \text{Poisson}(f_t(\mathbf{x})), \quad (1)$$

where $f_t : \mathbb{R}^D \rightarrow \mathbb{R}_+$ is a nonlinear regression model that predicts the expected value of y_t . Because this direct regression has a large number of parameters for all the nonlinear functions f_t over time frames $t = 1, \dots, T$, it often fails to train the regression model from a limited amount of training data. To tackle this problem, we formulate the foot traffic by K basis vectors $\mathbf{w}_k \in \mathbb{R}_+^T$ each of which represents a template traffic pattern of visitors as follows:

$$y_t | f, \mathbf{x}, \mathbf{W} \sim \text{Poisson}\left(\sum_{k=1}^K w_{kt} f_k(\mathbf{x})\right), \quad (2)$$

where $f_k : \mathbb{R}^D \rightarrow \mathbb{R}_+$ is a nonlinear regression model which predicts the weight (activation) for the k -th traffic pattern. This formulation can significantly reduce the number of parameters for the nonlinear regression model by setting $K \ll T$. We put a prior distribution on $\mathbf{W}$ to prevent overfitting. A gamma distribution is introduced for the prior on $\mathbf{W}$ as follows:

$$w_{kt} \sim \text{Gamma}(a_0^{(w)}, b_0^{(w)}), \quad (3)$$

where $a_0^{(w)}$ and $b_0^{(w)}$ are hyperparameters that we set in advance. We utilize the Gamma prior because it is a conjugate prior of the Poisson likelihood.

3.3 Training

We train the regression model f_k to maximize the log-marginal likelihood $\log p(\mathbf{Y} | \mathbf{X}, f)$ for N training pairs of $\mathbf{x}^{(n)}$ and $\mathbf{y}^{(n)}$. Since it is difficult to optimize this log-marginal likelihood analytically, we introduce the following variational posterior distribution q to optimize the likelihood approximately:

$$p(\mathbf{W} | \mathbf{X}, f) \approx q(\mathbf{W}) \triangleq \prod_k q(\mathbf{w}_k). \quad (4)$$

We train the regression model f_k to maximize a lower-bound of the log-marginal likelihood, namely the evidence lower bound (ELBO) defined as follows:

$$\mathcal{L}^{(w)} = \mathbb{E}_q[\log p(\mathbf{Y} | \mathbf{X}, f, \mathbf{W})] - \mathcal{D}_{\text{KL}}[q(\mathbf{W}) | p(\mathbf{W})]. \quad (5)$$

The whole training is formulated to interactively update the regression model f_k and the variational posterior q . This procedure is described as a variational expectation-maximization (EM) algorithm [28]. In the E-step, we update the variational distribution $q(w_{kt})$ while the regression model f_k is fixed. The variational distribution is updated to maximize the ELBO of Eq. (5) in the similar way as the Bayesian NMF [23]:

$$q(w_{kt}) \leftarrow \text{Gamma}(a_{kt}, b_{kt}), \quad (6)$$

$$a_{kt}^{(w)} = a_0^{(w)} + \sum_{n=1}^N y_t^{(n)} \psi_{kt}^{(n)}, \quad b_{kt}^{(w)} = b_0^{(w)} + \sum_{n=1}^N f_k(\mathbf{x}^{(n)}), \quad (7)$$

where $\psi_{kt}^{(n)} \propto \exp(\mathbb{E}_q[\log w_{kt}]) f_k(\mathbf{x}^{(n)})$ is an auxiliary variable satisfying $\sum_{k=1}^K \psi_{kt}^{(n)} = 1$, which is obtained by calculating a Lagrange multiplier. In the M-step, on the other hand, we update the regression model f_k while the variational distribution $q(w_{kt})$ is fixed. The regression model is updated to maximize the ELBO (Eq. (5)) by using a stochastic gradient descent method.

3.4 Prediction

The predicted foot traffic $\hat{y}_t^*$ is an expected value of the predictive distribution $p(y_t^* | \mathbf{x}^*, f, \mathbf{Y}, \mathbf{X})$. The expected value $\hat{y}_t^*$ is calculated as follows:

$$\hat{y}_t^* \triangleq \mathbb{E}_p[y_t^* | \mathbf{x}^*, f, \mathbf{Y}, \mathbf{X}] = \int y_t^* p(y_t^* | \mathbf{x}^*, f, \mathbf{Y}, \mathbf{X}) dy_t^* \quad (8)$$

$$= \int y_t^* \int p(y_t^* | \mathbf{x}^*, f, \mathbf{w}_t) p(\mathbf{w}_t | \mathbf{Y}, \mathbf{X}) d\mathbf{w}_t dy_t^* \quad (9)$$

$$\approx \int y_t^* \int p(y_t^* | \mathbf{x}^*, f, \mathbf{w}_t) q(\mathbf{w}_t) d\mathbf{w}_t dy_t^* \quad (10)$$

$$= \int \mathbb{E}_p[y_t^* | \mathbf{x}^*, f, \mathbf{w}_t] q(\mathbf{w}_t) d\mathbf{w}_t. \quad (11)$$

$$= \sum_{k=1}^K \mathbb{E}_q[w_{kt}] f_k(\mathbf{x}^*) = \sum_{k=1}^K \frac{a_{kt}^{(w)}}{b_{kt}^{(w)}} f_k(\mathbf{x}^*). \quad (12)$$

4. Experimental Evaluation of Foot Traffic Prediction

To evaluate the proposed method of the foot traffic prediction, we conducted an experiment with foot traffic recorded during real large-scale events.

4.1 Dataset

In this experiment, we used foot traffic data of 15 games recorded in the Kashima soccer stadium, which is a home stadium of a Japanese professional soccer team. Many of the spectators were the supporters of the home team. We recorded exiting spectators at a gate of the stadium for 15 games between December 2018 and December 2019. **Figure 2** shows how our recording was conducted at the gate of the stadium. For recording the foot traffic, we used a human tracking system based on a light detection and ranging (LiDAR) sensor [29], which can detect and track

Fig. 3 Comparison of foot traffic prediction performance among the NN, RF, and GAM.

Fig. 2 Traffic data collection.

Table 1 Features used for prediction in our experiment.

Feature	Value range	Description
First half win	{0, 1}	Whether the home team wins the 1st half
Kickoff time	[14, 19]	The start time of the game
Spectator count	[13502, 34312]	Sum of Spectators

human movement in a range between 0.5 and 22 m from the sensor at 10 fps. We recorded the foot traffic for 120 minutes from 90 minutes after the kickoffs (beginnings of each game), which almost corresponds to 30 minutes before the end of the games. We observed the number of people leaving the stadium every minute. Since we had only one human tracking system, our system was set up at a gate that most of the spectators left through. This gate was also closest to the stadium's parking area and the nearest train station.

Table 1 describes event information used to predict the foot traffic time series. These features were selected through our observation experience at the stadium. We conducted standardization for the features by their means and standard deviations. We also used squared of the continuous feature variables (i.e., the kickoff times and spectator counts).

4.2 Experimental Conditions

Since we had only traffic data for 15 games, the prediction performance was evaluated by a leave-one-out cross validation where one game was used for evaluation and the other $N = 14$ games were used for training in each validation. We calculated root mean square errors (RMSEs) and mean absolute percentage errors (MAPEs) for the evaluation of prediction performance. Since the traffic data can contain zeros, we calculated MAPEs as the average of absolute errors divided by $\max(10, y_t^*)$, where y_t^* is a ground-truth foot traffic at time t .

The hyperparameters for the proposed method (BNMF-based regression) were summarized as follows. A neural network with three fully-connected layers was used as the regression model f_k . The first two layers of the network consisted of 64 units with a rectified linear unit (ReLU) function as an activation function. Each layer has dropout with dropping rate of 0.2 to avoid overfitting. The output layer consisted of K units with a softplus activation function. We used an Adam optimizer with a learning rate of 1.0×10^{-3} and a weight decay rate of 1.0×10^{-5} for training the neural regression model f_k . The hyperparameters $a_0^{(w)}$ and $b_0^{(w)}$ were set to 1.0, $1.0 \times K$, respectively. The number of the iteration for the EM algorithm was set to 300. For every iteration of the EM algorithm, the weights of the neural network was updated 30 times. The number of basis vectors K , which depends on the complexity of the target data, was varied between 2 and 10 to evaluate the robustness against the hyperparameter setting. These hyperparameters were determined experimentally by using the cross validation.

We evaluated the proposed method with two baseline methods, a direct regression method and an NMF-based regression method. The direct regression method uses a regression model to predict all an time-series directly from the feature vector. The NMF-based regression method is similar to the proposed method

Fig. 4 Prediction results of foot traffic with the neural network as the regression model f_k .

except that the basis vectors were updated by a maximum likelihood manner instead of our Bayesian manner. For both of the baseline methods, we evaluated the Gaussian and Poisson likelihoods as a loss function. We used the same network configuration as our method.

To evaluate the effectiveness of the neural network (NN) regression, we evaluated a generalized additive model (GAM) [30] and a random forest (RF) [31] for f_k , which have been popularly used for training the regression from a small amount of data. The GAM was implemented by using a Python library called pyGAM [32], and the RF was implemented by using scikit-learn [33]. We set the regularization parameter of GAM (lam in pyGAM) to 6.0×10^{-2} , and the number of trees in RF (n_estimators in scikit-learn) to 20. The other hyperparam-

eters of GAM and RF followed their default values.

4.3 Experimental Results

Figure 3 shows the prediction performance of the foot traffic for each regression model. For all the regression models of NN, RF, and GAM, the NMF-based regressions outperformed the direct regressions in both RMSE and MAPE when the K giving the best performance was selected. These results demonstrated that our approach of introducing a low-rank approximation by NMF was effective for traffic prediction with the small amount of training data. The NMF-based regressions with maximum likelihood estimation were degraded by underfitting or overfitting if K was not appropriately selected. Our Bayesian method, in contrast, mitigated this problem and achieved stable performance over dif-

ferent values of K between 2 and 10. These characteristics will free us from carefully tuning the hyperparameter K .

The comparison of NN, RF, and GAM showed that the neural network improved the prediction performances of the (B)NMF-based regression methods with the best K conditions. The comparison of the Gaussian and Poisson likelihood functions showed that the NMF-based regressions with the Poisson likelihood had better RMSEs and MAPEs than those with the Gaussian likelihood in most values of K when the neural network was used for the regression model. This result indicates that the formulation with a suitable function for count data was helpful to predict the foot traffic from the small amount of the training data. The proposed BNMF-based neural regression had the best value of the RMSE when K was 6, and the NMF-based neural regression with Gaussian and Poisson likelihoods had those when K was 2. The prediction performance of the NMF-based regressions deteriorated without the tuning of K . In contrast, the proposed BNMF-based regression, which estimates the basis vectors in a Bayesian manner, maintained good prediction performance in both the RMSE and MAPE regardless of the number of bases K . These results demonstrated that our Bayesian approach was effective for cutting down the cost of tuning K .

Figure 4 shows the prediction results of the foot traffic for all the 15 games with NN-based regression. For the NMF-based and BNMF-based methods, we showed the results with K having the best RMSEs. We can see that the direct methods often degraded the prediction performance, especially as seen in Game IDs: 4, 8, and 13. The proposed BNMF-based regression method seemed to predict the ground-truth foot traffic better than other baseline methods. The foot traffic with three congestion peaks, however, was not successfully predicted as shown in Game IDs: 1 and 14. This could be caused by an extremely limited number (i.e., one) of training samples. Since we performed the leave-one-out cross validation, the training data had only one sample having three peaks when Game ID: 1 or 14 was predicted. Our future work includes extending our method to deal with these unexpected or difficult-to-predict situations.

5. Travel Time Prediction

We then describe the travel time prediction method utilizing the estimated foot traffic $\hat{\mathbf{y}}$.

5.1 Problem Specification

The problem setting for predicting a travel time for passing a target road during a large-scale event is specified as follows:

Inputs: the predicted value of foot traffic series $\hat{\mathbf{y}}^* \in \mathbb{R}_+^T$
Outputs: the expected value of travel time series $\hat{\mathbf{v}}^* \in \mathbb{R}_+^T$
Training data: N pairs of the predicted foot traffic $\hat{\mathbf{y}}^{(n)} \in \mathbb{R}_+^T$, and the travel time $\mathbf{v}^{(n)} \in \mathbb{R}_+^T$ that are collected in advance,

where $U^{(n)}$ represents the number of the observed travel times for passing the target road at the n -th event. As detailed in the next section, we collected the training data of travel times by using the GPS trajectories of mobile phones.

5.2 Model Formulation

We estimate the travel time $v_t \in \mathbb{R}$ at time t from the past S frames (minutes) of the foot traffic with a log-Normal likelihood:

$$v_t | g, \tilde{\mathbf{y}}_t, \sigma^2 \sim \text{LogNormal}(g(\tilde{\mathbf{y}}_t), \sigma^2), \quad (13)$$

where $g: \mathbb{R}^{S/J} \rightarrow \mathbb{R}_+$ is a nonlinear mapping that predicts the mean parameter of the log-Normal distribution, and $\tilde{\mathbf{y}}_t \in \mathbb{R}_+^{S/J}$ is a feature vector containing the average values of the predicted traffic $\hat{y}_t$ for every J minutes as follows:

$$\tilde{\mathbf{y}}_t = \left[\frac{\sum_{i=0}^{J-1} \hat{y}_{t-S+i}}{J}, \frac{\sum_{i=J}^{2J-1} \hat{y}_{t-S+i}}{J}, \dots, \frac{\sum_{i=S-J}^{S-1} \hat{y}_{t-S+i}}{J} \right]. \quad (14)$$

We utilized the log-Normal distribution because it has been reported to be suitable for representing travel times [34], [35].

In this paper, we estimate the variance parameter σ^2 in two ways: maximum-likelihood and Bayesian manners. For the Bayesian approach, we put the following inverse gamma prior on σ^2 , which is a conjugate prior of the log-Normal likelihood:

$$\sigma^2 \sim \text{InvGamma}(a_0^{(\sigma)}, b_0^{(\sigma)}), \quad (15)$$

where $a_0^{(\sigma)}$ and $b_0^{(\sigma)}$ are hyperparameters.

5.3 Training Based on Maximum Likelihood Estimation of σ^2

In the maximum likelihood approach, we jointly find the regression model g and the parameter σ^2 that maximizes the log-likelihood $\log p(\mathbf{V} | g, \tilde{\mathbf{Y}}, \sigma^2)$. As in the proposed traffic prediction, the regression model g and the parameter σ^2 are alternately and iteratively updated. We first update the variance parameter σ^2 while g is fixed:

$$\sigma^2 \leftarrow \frac{\sum_{n=1}^N \sum_{u=1}^{U^{(n)}} \left\{ \log v_u^{(n)} - g(\tilde{\mathbf{y}}_{t_u}^{(n)}) \right\}^2}{\sum_{n=1}^N U^{(n)}} \quad (16)$$

where $t_u^{(n)}$ represents the timestamp for the u -th observation. We then update g to maximize the log-marginal likelihood while σ^2 is fixed by using a stochastic gradient descent method.

The predicted travel time $\hat{v}_t^*$ is an expected value of the log-Normal distribution $p(\hat{v}_t^* | \tilde{\mathbf{y}}_t^*, g, \sigma^2)$ as follows:

$$\hat{v}_t^* = \exp \left[g(\tilde{\mathbf{y}}_t^*) + \frac{\sigma^2}{2} \right] \quad (17)$$

where g and σ^2 are trained at the training stage.

5.4 Training Based on Bayesian Estimation of σ^2

In the Bayesian approach, we find the regression model g_t that maximizes the log-marginal likelihood $\log p(\mathbf{V} | g, \tilde{\mathbf{Y}})$ while estimating the posterior distribution of $p(\sigma^2 | \mathbf{V}, \tilde{\mathbf{Y}})$. This training utilizes the expectation-maximization algorithm that approximately optimizes the likelihood by introducing the following variational posterior distribution q :

$$p(\sigma^2 | g, \mathbf{V}, \tilde{\mathbf{Y}}) \approx q(\sigma^2). \quad (18)$$

Since the log-marginal likelihood is intractable, we maximize its

lower-bound (ELBO) as follows:

$$\mathcal{L}^{(\sigma)} = \mathbb{E}_q \left[\log p(\mathbf{V} | g, \tilde{\mathbf{Y}}, \sigma^2) \right] - \mathcal{D}_{\text{KL}} \left[q(\sigma^2) \middle| p(\sigma^2) \right]. \quad (19)$$

In the E-step, we update the variational distribution $q(\sigma^2)$ while g is fixed:

$$q(\sigma^2) \leftarrow \text{InvGamma}(a^{(\sigma)}, b^{(\sigma)}), \quad (20)$$

$$a^{(\sigma)} = \frac{1}{2} \sum_{n=1}^N U^{(n)} + a_0^{(\sigma)}, \quad (21)$$

$$b^{(\sigma)} = \frac{1}{2} \sum_{n=1}^N \sum_u \left\{ \log v_u^{(n)} - g(\tilde{\mathbf{y}}_u^{(n)}) \right\}^2 + b_0^{(\sigma)}. \quad (22)$$

In the M-step, we update g to maximize the ELBO while $q(\sigma^2)$ is fixed by using a stochastic gradient descent method.

The predicted travel time v_i^* is an expected value of the predictive distribution $p(v_i^* | \tilde{\mathbf{y}}_i^*, g, \mathbf{V}, \tilde{\mathbf{Y}})$. The expected value is calculated as follows:

$$v_i^* \triangleq \mathbb{E}_p \left[v_i^* | \tilde{\mathbf{y}}_i^*, g, \mathbf{V}, \tilde{\mathbf{Y}} \right] = \int v_i^* p(v_i^* | \tilde{\mathbf{y}}_i^*, g, \mathbf{V}, \tilde{\mathbf{Y}}) dv_i^* \quad (23)$$

$$\approx \int v_i^* \int p(v_i^* | \tilde{\mathbf{y}}_i^*, g, \sigma^2) q(\sigma^2) d\sigma^2 dv_i^* \quad (24)$$

$$= \int \mathbb{E}_p \left[v_i^* | \tilde{\mathbf{y}}_i^*, g, \sigma^2 \right] q(\sigma^2) d\sigma^2. \quad (25)$$

Since it is difficult to calculate the expectation by q in the Eq. (25) analytically, we obtain the predicted values v_i^* by a Monte-Carlo approximation as follows:

$$v_i^* = \frac{1}{L} \sum_{l=1}^L \exp \left\{ g(\tilde{\mathbf{y}}_i^*) + \frac{(\sigma^{(l)})^2}{2} \right\}, \quad (26)$$

where $(\sigma^{(l)})^2 \sim q(\sigma^2)$ is a sample of the variational posterior distribution obtained at the training stage.

6. Experimental Evaluation of Travel Time Prediction

To demonstrate an application of the proposed foot traffic method, we conducted an experiment for travel time prediction from the Kashima soccer stadium to the closest highway interchange, the Itako interchange. Note that the causal relationship between the foot traffic and the travel time is expected to be common in general event sites located in the countryside. Since there are few event sites in the countryside that attract tens of thousands of people, a large number of people leaving an event site will cause traffic congestion around the event site.

6.1 Dataset

In the experiment of travel time prediction, we used the GPS data of mobile phones that were collected at the same times and on the same dates as the foot traffic prediction. The GPS data, provided by Blogwatcher Inc., contained anonymized user information including a user ID, location (latitude and longitude), and timestamp. This information was recorded when the user received advertisements. We conducted the following pre-processing steps to reduce the effect of noise in the GPS data.

Fig. 5 Origin area and destination area. The road shown in blue is recommended in Google Maps.

We used the GPS trajectories having speed between 0.6 km/h and 150 km/h. Since the GPS data is not collected constantly, we conducted a linear interpolation of the positions and timestamps between the observed GPS locations. To determine which road the mobile device traveled, we aligned each location of the GPS samples to the nearest node of the open street map [36]. A blue line in Fig. 5 was the recommended route from the Kashima stadium to the Itako interchange in a popular commercial service of Google Maps. To compare with the travel time shown in the Google Maps, we used the GPS data traveling from the origin area through the red borders into the destination area. The travel time for each trajectory was defined by subtracting the last timestamp at the origin area from the first timestamp at the destination area. We collected travel times estimated by Google Maps at 9 games out of all the 15 games as we started collecting data from Google Maps from May 3, 2019, which is five games later than we had started collecting the GPS data. We had the data from only 9 games because we accidentally failed to collect the data at one game (Game ID: 10).

6.2 Experimental Conditions

The hyperparameters for the proposed method (two-step regression) were summarized as follows. A neural network with three fully-connected layers was used for the regression model g . The first two layers consisted of 8 units with ReLU activation function, and their dropout rates were fixed to 0.2. We set the hyperparameters $a_0^{(\sigma)}$, $b_0^{(\sigma)}$, S , and J to 2.0, 1.0×10^{-1} , 60, and 10, respectively. As the input of the regression model g , we used the predicted values by the BNMF-based foot traffic regression with the neural regression model.

We compared the prediction performance of the travel time with an indicator and a direct regression method, and a two-step regression method. The indicator was the predicted travel time on the blue line in Fig. 5 by Google Maps. The direct regression method predicted the travel time directly by taking as inputs the event information and departure time from the origin area for the regression model g . The two-step regression method predicted the travel time by using the predicted foot traffic from baseline foot traffic prediction methods.

Since we have only 9 travel time series predicted by Google

Table 2 Comparison of travel time prediction performance.

Method	Estimation of σ^2	Input	RMSE	MAPE
Google Maps	N/A	N/A	11.93	26.30
Naïve regression	Maximum likelihood	Event information & departure time	8.70	21.35
	Bayesian	Event information & departure time	8.67	21.26
Two-step regression	Maximum likelihood	Direct regression (Gaussian)	9.16	19.83
		Direct regression (Poisson)	7.40	17.87
		NMF-based regression (Gaussian)	7.78	18.79
		NMF-based regression (Poisson)	8.13	20.00
		BNMF-based regression	7.54	17.92
	Bayesian	Direct regression (Gaussian)	9.07	19.64
		Direct regression (Poisson)	7.42	17.89
		NMF-based regression (Gaussian)	7.69	18.63
		NMF-based regression (Poisson)	8.04	19.82
		BNMF-based regression	7.59	18.05

Fig. 6 Excerpts of predicted travel times. The game IDs correspond to those of Fig. 4. The input of the two-step regression methods is predicted foot traffic by the BNMF-based regression method.

Maps, the prediction performance was evaluated by a leave-one-out cross validation where one game, having the travel time predicted by Google Maps, was used for evaluation and the other $N = 14$ games were used for training in each validation. The rest of the experimental conditions followed the experimental conditions of foot traffic prediction.

6.3 Experimental Results

Table 2 shows the prediction performance of travel time. Our method, which was the two-step regression method with the predicted foot traffic, outperformed Google Maps and the naïve regression methods in both the RMSE and MAPE. For example, the best RMSE and MAPE among the naïve direct regressions were 8.67 and 21.26, while the best RMSE and MAPE among the two-step regressions were 7.40 and 17.87. These results showed that utilizing the predicted foot traffic improved the prediction performance of travel time. The performance difference between the maximum-likelihood and Bayesian estimation of the variance parameter σ^2 was not significant. This could be because the Bayesian approach had little effect on the travel time prediction methods having only one parameter, unlike the foot traffic prediction. The performances of the two-step regression meth-

ods with different foot-traffic prediction methods were comparable except that the direct regression with the Gaussian likelihood and the NMF-based regression with the Poisson likelihood. The significant performance differences were not observed because the number of the evaluated data was limited. At least, we can constantly see the effectiveness of the (B)NMF-based two-step approach compared to the naïve regression and Google Maps.

Figure 6 shows some examples of the prediction results of travel times. The prediction of the naïve regression often failed, which predicted the travel time from the event information and departure time. The travel time estimated by Google Maps had delays from the actual travel times measured by GPS. In contrast, the two-step regression methods correctly estimated the travel time following the ground truth GPS data.

7. Conclusion

This paper presented a framework for efficient traffic prediction even from a limited amount of training data. Our proposed method extracts a small number of basis traffic patterns from training data by NMF and predicts their weights by a nonlinear regression model. We formulated this method as a probabilistic generative model with a Poisson likelihood, which is the natural

representation for the count data such as foot traffic. The training procedure of the proposed method is formulated in a variational EM algorithm to prevent the model parameters from overfitting. We also proposed a travel time prediction method utilizing the foot traffic predicted by the proposed method. In the experiment with foot traffic at a soccer stadium, the proposed method outperformed a baseline method that predicted all-time series directly. The results also showed that our method outperformed the NMF-based regression methods trained by maximum likelihood estimation with a Gaussian likelihood, which has been utilized in the conventional studies [26], [27]. In addition, the proposed method predicts the foot traffic robustly regardless of the number of behavior patterns, where the method with a maximum likelihood approach deteriorates without the tuning of this parameter. The travel time prediction, on the other hand, was performed to predict travel times from the stadium to the nearest highway interchange. The experimental results showed that our method outperformed the naïve direct regression methods and the travel time prediction provided in Google Maps.

Our future work is to enable our proposed model to deal with unexpected or difficult-to-predict situations. The proposed method does not take into account changes in human behaviors after the prediction due to unexpected causes of event contents or situations. We will extend the proposed method to enable online (real-time) parameter updates.

Acknowledgments This work was partially supported by JSPS KAKENHI Grant Number JP19H04943.

References

- [1] Zhang, J., Wang, F.-Y., Wang, K., Lin, W.-H., Xu, X. and Chen, C.: Data-driven intelligent transportation systems: A survey, *IEEE Trans. Intelligent Transportation Systems*, Vol.12, No.4, pp.1624–1639 (2011).
- [2] Alsrehin, N.O., Klaib, A.F. and Magableh, A.: Intelligent transportation and control systems using data mining and machine learning techniques: A comprehensive study, *IEEE Access*, Vol.7, pp.49830–49857 (2019).
- [3] Boukerche, A. and Wang, J.: Machine Learning-based traffic prediction models for Intelligent Transportation Systems, *Computer Networks*, Vol.181, p.107530 (2020).
- [4] Jiang, R., Song, X., Huang, D., Song, X., Xia, T., Cai, Z., Wang, Z., Kim, K.-S. and Shibasaki, R.: Deepurbanevent: A system for predicting citywide crowd dynamics at big events, *Proc. 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp.2114–2122 (2019).
- [5] Shimizu, H.: Deep Reinforcement Learning for Pedestrian Guidance, *PRIMA 2020: Principles and Practice of Multi-Agent Systems: 23rd International Conference*, p.334, Springer Nature (2020).
- [6] Nihan, N.L. and Holmesland, K.O.: Use of the Box and Jenkins time series technique in traffic forecasting, *Transportation*, Vol.9, No.2, pp.125–143 (1980).
- [7] Xu, D.-W., Wang, Y.-D., Jia, L.-M., Qin, Y. and Dong, H.-H.: Real-time road traffic state prediction based on ARIMA and Kalman filter, *Frontiers of Information Technology & Electronic Engineering*, Vol.18, No.2, pp.287–302 (2017).
- [8] Kang, D., Lv, Y. and Chen, Y.-Y.: Short-term traffic flow prediction with LSTM recurrent neural network, *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp.1–6, IEEE (2017).
- [9] Ni, M., He, Q. and Gao, J.: Forecasting the subway passenger flow under event occurrences with social media, *IEEE Trans. Intelligent Transportation Systems*, Vol.18, No.6, pp.1623–1632 (2016).
- [10] Lee, D.D. and Seung, H.S.: Algorithms for Non-Negative Matrix Factorization, *Proc. 13th International Conference on Neural Information Processing Systems*, pp.535–541, MIT Press (2000).
- [11] Okukubo, T., Bando, Y., Onishi, M. and Ando, H.: Foot Traffic Prediction for Large-Scale Events Based on Pattern-Aware Neural Regression, *Proc. AAAI 21 Workshop on AI for Urban Mobility* (2021).
- [12] Wu, C.-H., Ho, J.-M. and Lee, D.-T.: Travel-time prediction with support vector regression, *IEEE Trans. Intelligent Transportation Systems*, Vol.5, No.4, pp.276–281 (2004).
- [13] Leshem, G. and Ritov, Y.: Traffic flow prediction using adaboost algorithm with random forests as a weak learner, *Proc. World Academy of Science, Engineering and Technology*, Vol.19, pp.193–198, Citeseer (2007).
- [14] Abrishami, S., Kumar, P. and Nienaber, W.: Smart stores: A scalable foot traffic collection and prediction system, *Industrial Conference on Data Mining*, pp.107–121, Springer (2017).
- [15] Duan, P., Mao, G., Kang, J. and Huang, B.: Estimation of Link Travel Time Distribution With Limited Traffic Detectors, *IEEE Trans. Intelligent Transportation Systems* (2019).
- [16] Lin, Z., Feng, J., Lu, Z., Li, Y. and Jin, D.: Deepstn+: Context-aware spatial-temporal neural network for crowd flow prediction in metropolis, *Proc. AAAI Conference on Artificial Intelligence*, Vol.33, pp.1020–1027 (2019).
- [17] Liao, B., Zhang, J., Wu, C., McIlwraith, D., Chen, T., Yang, S., Guo, Y. and Wu, F.: Deep sequence learning with auxiliary information for traffic prediction, *Proc. 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp.537–546 (2018).
- [18] Adiloğlu, K. and Vincent, E.: Variational Bayesian inference for source separation and robust feature extraction, *IEEE/ACM Trans. Audio, Speech, and Language Processing*, Vol.24, No.10, pp.1746–1758 (2016).
- [19] Guillaumet, D. and Vitria, J.: Non-negative matrix factorization for face recognition, *Catalonian Conference on Artificial Intelligence*, pp.336–344, Springer (2002).
- [20] Xu, W., Liu, X. and Gong, Y.: Document clustering based on non-negative matrix factorization, *Proc. 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pp.267–273 (2003).
- [21] Pei, Y., Chakraborty, N. and Sycara, K.: Nonnegative matrix tri-factorization with graph regularization for community detection in social networks, *24th International Joint Conference on Artificial Intelligence* (2015).
- [22] Févotte, C., Bertin, N. and Durrieu, J.-L.: Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis, *Neural Computation*, Vol.21, No.3, pp.793–830 (2009).
- [23] Cemgil, A.T.: Bayesian inference for nonnegative matrix factorisation models, *Computational Intelligence and Neuroscience* (2009).
- [24] Deng, D., Shahabi, C., Demiryurek, U., Zhu, L., Yu, R. and Liu, Y.: Latent space model for road networks to predict time-varying traffic, *Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.1525–1534 (2016).
- [25] He, P., Jiang, G., Lam, S.-K. and Sun, Y.: Learning heterogeneous traffic patterns for travel time prediction of bus journeys, *Information Sciences*, Vol.512, pp.1394–1406 (2020).
- [26] Zhu, W. and Yan, Y.: Joint linear regression and nonnegative matrix factorization based on self-organized graph for image clustering and classification, *IEEE Access*, Vol.6, pp.38820–38834 (2018).
- [27] Mei, J., De Castro, Y., Goude, Y., Azais, J.-M. and Hébrail, G.: Non-negative matrix factorization with side information for time series recovery and prediction, *IEEE Trans. Knowledge and Data Engineering*, Vol.31, No.3, pp.493–506 (2018).
- [28] Bernardo, J., Bayarri, M., Berger, J., Dawid, A., Heckerman, D., Smith, A., West, M., et al.: The variational Bayesian EM algorithm for incomplete data: With application to scoring graphical model structures, *Bayesian Statistics*, Vol.7, No.453–464, p.210 (2003).
- [29] Onishi, M.: Analysis and Visualization of Large-Scale Pedestrian Flow in Normal and Disaster Situations, *ITE Trans. Media Technology and Applications*, Vol.3, No.3, pp.170–183 (2015).
- [30] Hastie, T.J. and Tibshirani, R.J.: *Generalized Additive Models*, Vol.43, CRC Press (1990).
- [31] Breiman, L.: Random forests, *Machine Learning*, Vol.45, No.1, pp.5–32 (2001).
- [32] Servén, D. and Brummitt, C.: pyGAM: Generalized additive models in python, *Zenodo*, Vol.10 (2018).
- [33] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E.: Scikit-learn: Machine Learning in Python, *Journal of Machine Learning Research*, Vol.12, pp.2825–2830 (2011).
- [34] Shi, C., Chen, B.-Y. and Li, Q.: Estimation of travel time distributions in urban road networks using low-frequency floating car data, *ISPRS International Journal of Geo-Information*, Vol.6, No.8, p.253 (2017).
- [35] Durán-Hormazábal, E. and Tirachini, A.: Estimation of travel time variability for cars, buses, metro and door-to-door public transport trips in Santiago, Chile, *Research in Transportation Economics*, Vol.59, pp.26–39 (2016).
- [36] Boeing, G.: OSMnx: New methods for acquiring, constructing, ana-

lyzing, and visualizing complex street networks, *Computers, Environment and Urban Systems*, Vol.65, pp.126–139 (2017).

Takafumi Okukubo received his Master's degree in service science from University of Tsukuba, Ibaraki, Japan, in 2021. He is currently working at Yamato Transport Corporation, Tokyo, Japan. His research interests include human flow analysis and probabilistic modeling.

Yoshiaki Bando received his M.S. and Ph.D. degrees in informatics from Kyoto University, Kyoto, Japan, in 2015 and 2018, respectively. He is currently a Researcher with the Artificial Intelligence Research Center (AIRC), National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan. He is also a Visiting Researcher with the RIKEN Center for Advanced Intelligence Project (AIP), Tokyo. His research interests include microphone array signal processing, deep Bayesian learning, robot audition, and field robotics. He is a member of RSJ and IPSJ.

Masaki Onishi received his M.Eng. and Dr.Eng. degrees from Osaka Prefecture University, in 1999 and 2002, respectively. From 2002 to 2006, he was a Research Scientist with the Bio-Mimetic Control Research Center, RIKEN. Since 2006, he has been a Research Scientist with the National Institute of Advanced Industrial Science and Technology (AIST), where he is currently the Team Leader of the Social Intelligence Research Team, Artificial Intelligence Research Center. His research interests include video surveillance, crowd simulation, and automated machine learning.