Information Systems 54 (2015) 169–188

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

Real-time traffic incident detection using a probabilistic topic model

Akira Kinoshita [a,b,*], Atsuhiro Takasu [b], Jun Adachi [b]

[a] The University of Tokyo, 7-3-1 Hongo, Bunkyo, Tokyo, Japan
[b] National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda, Tokyo, Japan

ARTICLE INFO

Article history:
Received 15 December 2014
Received in revised form
18 May 2015
Accepted 1 July 2015
Available online 10 July 2015

Keywords:
Anomaly detection
Automatic incident detection
Probabilistic topic model
Probe-car data
Real-time processing
Traffic state estimation

ABSTRACT

Traffic congestion occurs frequently in urban settings, and is not always caused by traffic incidents. In this paper, we propose a simple method for detecting traffic incidents from probe-car data by identifying unusual events that distinguish incidents from spontaneous congestion. First, we introduce a traffic state model based on a probabilistic topic model to describe the traffic states for a variety of roads. Formulas for estimating the model parameters are derived, so that the model of usual traffic can be learned using an expectation–maximization algorithm. Next, we propose several divergence functions to evaluate differences between the current and usual traffic states and streaming algorithms that detect high-divergence segments in real time. We conducted an experiment with data collected for the entire Shuto Expressway system in Tokyo during 2010 and 2011. The results showed that our method discriminates successfully between anomalous car trajectories and the more usual, slowly moving traffic patterns.

1. Introduction

Automatic incident detection (AID) is a crucial technology in intelligent transport systems, particularly in terms of reducing congestion on freeways [1]. Traffic incidents often cause traffic congestion, causing great inconvenience and economic loss to society. A technology that can detect traffic incidents in real time and alert people accordingly would therefore be a desirable way of reducing these ill effects.

Against this background, there have been many studies on AID, e.g., [2,3]. Most of the approaches exploit data sent from stationary sensors and cameras installed on roads. However, the installation and maintenance of such sensors is expensive, with only the main routes likely to have them [4]. On the other hand, the use of probe-car data (PCD), on which we focus in this paper, is becoming increasingly important, as the number of probe cars and the size of the associated data archives increase. PCD includes time-stamps and vehicle locations, and may contain additional data such as the speed and direction of the probe cars. Although a PCD system cannot monitor all cars, it enables traffic administrators to watch a large area at a lower cost than by using stationary sensors. In addition, a PCD system can follow the sequence of movements for a probe car in detail, which is hard to achieve via stationary sensors, and trajectory mining can be applied to the collected data.

Using PCD for freeways, it is easy to detect any reduction in speed, which sometimes implies congestion, by analyzing the speeds of the probe cars. However, this method is less applicable to local streets, where the many crossings and traffic lights can cause cars to stop frequently under normal circumstances. Moreover, speed reduction is not always abnormal, even on freeways, and is not always caused by incidents such as accidents, which we would regard as sudden and unusual traffic events in this paper.

There are two types of congestion: spontaneous and abnormal [2]. Detecting spontaneous congestion is less

* Corresponding author.
E-mail addresses: kinoshita@nii.ac.jp (A. Kinoshita), takasu@nii.ac.jp (A. Takasu), adachi@nii.ac.jp (J. Adachi).

http://dx.doi.org/10.1016/j.is.2015.07.002
0306-4379/

important, as it originates in road design and urban planning. Any road may have potential bottlenecks such as upslopes, curves, junctions, tollgates, and narrow sections. Vehicles are likely to slow down at the bottlenecks, with vehicular gaps shortening and drivers in the following cars having to brake. Congestion will then occur even in the absence of a specific traffic incident [5]. Spontaneous congestion also occurs when the traffic demand exceeds the traffic capacity at such bottlenecks, and it is not resolved until the demand drops below the capacity [6]. Drivers may be familiar with the locations of such potential bottlenecks, and they can avoid them. On the other hand, abnormal congestion originates in traffic incidents, which need to be detected in real time to prevent or resolve any sudden heavy congestion.

In this paper, we propose an AID method for detecting traffic incidents in real time by identifying abnormal car movements and distinguishing such movements from those occurring in spontaneous congestion. Our method measures differences between current traffic states (CTS) and usual traffic states (UTS), and has two aspects, namely, traffic state estimation and anomaly detection. First, we employ a probabilistic topic model [7] to model the generation of the PCD, which is influenced by hidden traffic situations such as "smooth" and "congested." The model introduces a single set of several hidden component states that are associated with probabilistic distributions over the PCD values, and each road segment during a certain time period has its own set of mixing coefficients. Using archived PCD, maximum-likelihood parameters of the model are estimated by an expectation–maximization (EM) algorithm. The estimated model reflects the usual state over the whole observation period. Our incident detection method simply follows the intuitive meaning of "anomaly." To detect incidents, the proposed method estimates the hidden state behind an observed PCD value and compares this current state with the usual state. If the current state is significantly different from the usual state, it is recognized as an anomaly.

We conducted an experiment using PCD observed for the entire Shuto Expressway system in Tokyo during 2010 and 2011. The total length of the Shuto Expressway system is approximately 300 km, and the daily traffic is about 1,000,000 vehicles per day [8]. Although the Shuto Expressway system forms the main artery system for the Tokyo area, there are many bottlenecks, and the speed limit is 60 km/h or less over most of the system [9]. Our experiment showed that the proposed method can identify trajectories involved in an incident better than existing methods.

The main contributions of this paper are as follows.

- We propose a new method for estimating traffic states by applying a probabilistic topic model to PCD, whereby road segments are characterized in terms of their expected performance hourly.
- We propose several methods for quantitative evaluation of the divergence of the CTS from the UTS using the traffic state model. We also propose several streaming algorithms that detect traffic incidents according to this divergence, whereby the detection is conducted adaptively in terms of the road segments and time periods.
- Our experiment showed that the traffic state model could be estimated using the observed PCD to reveal bottleneck sections on routes. It also showed that our AID method performed better than existing methods at identifying anomalous behavior by vehicles encountering incidents.

This paper is an extended version of the work published in the Proceedings of the Workshops of the EDBT/ICDT 2014 Joint Conference [10]. Here, we extend our previous work by introducing new divergence functions, developing a new algorithm, and conducting a new experiment using a larger-scale dataset.

The remainder of the paper is organized as follows. In the next section, we present related work. In Section 3, we introduce the traffic state model and describe our incident detection method. We conducted an experiment to evaluate our proposed method using a real PCD, and Section 4 describes the procedure and results of this experiment. We discuss the experimental results, issues and future work in Section 5. Finally, we conclude the paper in Section 6.

2. Related work

Anomaly detection [11] has attracted increasing research interest not only for communication networks [12,13] and social networks [14], but also for urban data. Using car-parking data, for example, useful trends as well as unusual behavior can be automatically extracted by an anomaly-detection technique [15]. AID can be considered as an application of anomaly or outlier detection to vehicular traffic data. Several AID methods have been proposed that exploit temporal data for vehicular speed [16] or flow data [17], which can be extracted from roadside surveillance cameras [18]. From the viewpoint of machine learning, AID can be regarded as a classification problem. Abdulhai and Ritchie [19] used neural networks, and Yuan and Cheu [20] used support vector machines to classify the observed vectors from stationary sensors as being incident based or otherwise. AID can also be regarded as an application of the change-point detection problem in time-series analysis, with Wang et al. [3] developing a hybrid method using time-series analysis and machine learning.

PCD, on which we are focusing, are different from the data on which existing work has been based. PCD are time-ordered sequences of points in spatio-temporal spaces, or trajectories. Piciarelli et al. [21] proposed an anomaly-detection method for trajectory data. This work used feature values extracted from the entire trajectory, implying that detection is not attempted during the ongoing movement of an object. A number of other studies have been carried out on anomaly detection from trajectory data that have been extracted from surveillance-video material [22,23]. This kind of trajectory data is different from PCD in that the sphere of movement is limited. Animal-movement data are an example of trajectory data in which the objects can move around a wide area. Lee et al. [24] proposed a method to find trajectory outliers using the example of animal-movement data. Although this method

exploited spatial features of the trajectory, it did not offer a method that used temporal or spatio-temporal features.

Turning now to PCD, Zhu et al. [25] applied outlier detection methods to feature vectors carefully extracted from PCD using heuristics. If an incident occurs, cars upstream of the incident will travel slower, and downstream cars will travel faster. In addition, a car passing an incident position before the incident will travel faster than one passing just after the incident. If $v(d,t,l)$ is the vehicular speed in link l at time t on date d, Zhu et al. proposed the following four features: $v(d,t,l)$, $v(d,t,l)-v(d,t-1,l)$, $v(d,t,l-1)$, and $v(d,t,l+1)-v(d,t,l)$, where link $l-1$ is the next link upstream of l, and $l+1$ is the next link downstream. These feature vectors are filtered using the heuristics above and analyzed via distance-based outlier detection. In another AID study, Akatsuka et al. [2] proposed an alternative feature vector.

Previous work exploits the characteristics of congested traffic, such as slowdown, in which vehicular speed decreases even in the absence of a traffic incident. In this paper, we take another approach by following the intuitive meaning of "anomaly", namely, an unusual event. Intuitively, we can identify some "traffic states" as being "smooth" or "congested," although we cannot measure or observe the state directly or objectively. Despite many studies having considered the traffic-state estimation problem, there is no general agreement about a formal definition for a "traffic state." Some research estimates the traffic state in terms of vehicular speed [26,27], and this kind of estimation characterizes states, i.e., quantized speeds, as "free" or "congested" [28]. Yoon et al. [4] proposed two feature values based on vehicular speed for detecting "bad" traffic states, i.e., slow traffic. As an alternative, Kerner et al. [29] used travel time. Xia et al. [30] used a clustering method to identify congested traffic in a feature space involving traffic flow, speed, and occupancy. This approach has been well studied in traffic engineering [6].

The traffic state can be regarded as a set of values of latent variables that reflect the actual traffic in some way. Following a traffic incident, the traffic state is different from the usual traffic state, even though the set of possible traffic states is not obvious. Probabilistic models that involve latent variables, or latent-variable models, can describe the data so that the state can be determined automatically from a given dataset. Liao et al. [31] proposed a method for detecting anomalies by modeling taxi probe data using conditional random fields (CRF). This model requires a labeled dataset for model estimation. Therefore, the ground-truth data for incidents are necessary in addition to trajectory data. Conversely, in this paper, we take an unsupervised learning approach. Developing a detection technique that does not require ground-truth data should reduce the costs of preparing data and minimize human errors. Qi and Ishak [32] used a hidden Markov model (HMM) to describe the generation of temporal data about vehicular speed observed by loop detectors. Kwon and Murphy [33] modeled traffic with coupled HMMs, which assumed that the latent states have the Markov property for both temporal and spatial aspects. Herring et al. [34] applied coupled HMMs to taxi probe data on arterial roads to estimate traffic conditions. We also tried HMM to detect traffic incidents on the Shuto Expressway in our preliminary experiment, however, we found that the traffic state estimation there was poor. In this paper, we use a model that ignores spatial and temporal correlations. This simplification allows the number of model parameters to be reduced, thereby reducing both time complexity and memory usage and enabling the method to work together with real-time applications.

Probabilistic topic models, which were originally studied in the field of natural language processing [7], are also models that including latent variables. Latent Dirichlet allocation (LDA) [35] is the simplest such topic model, and several attempts have been made to model traffic data or urban data using LDA or an LDA extension. Yuan et al. [36] used a topic model to discover functional regions in a city using taxi probe data and point-of-interest information. Similarly, Farrahi and Gatica-Perez [37] used a topic model to discover human routines using mobile-phone location data. LDA can be extended to model object movement in surveillance-video images [38] and the flow of people entering or exiting a building [39]. These approaches take the temporal dependency of latent variables into account. Anomaly detection using topic models has also been investigated in previous work. Yu et al. [40] proposed a topic model for detecting an anomalous group of individuals in a social network. Several attempts have also been made to find anomalies using topic models and surveillance cameras [41,42]. Jeong et al. [43] proposed a topic model for detecting anomalous trajectories of people or vehicles in surveillance-video images.

In this paper, we exploit the idea of probabilistic topic models and aim to identify traffic incidents using PCD and an LDA-equivalent model. The proposed method first estimates a set of traffic states over an entire route, and the mixing coefficients for each road segment, with a "traffic state" corresponding to a "topic", to obtain a model for the usual traffic over the dataset. Whereas the traffic states are identical for any location and time period, the mixing coefficient represents local characteristics, enabling our model to operate despite ignorance of any spatial or temporal correlations. We then try to identify any unusual events as incidents. This approach will enable incident detection in an unsupervised way, i.e., without labeled data. In addition, because this approach avoids heuristics, it will adapt automatically to changes in traffic circumstances and be applicable to large road networks with changing characteristics over time. We also aim for incident detection in real time. We previously proposed a system architecture for real-time traffic incident detection [9]. The present paper proposes an incident detector that operates in the backend of such a system.

3. Methodology

This section describes our traffic state model and incident detection method. We first introduce a method for applying a probabilistic topic model to PCD. Our task is to estimate the model parameters using a PCD archive and to identify incidents by comparing the UTS and CTS, which are obtained from the learned model. Table 1 summarizes the notations used in this paper.

Table 1
Notation.

Notation	Definition
K	Number of traffic states
k	Index of a traffic state
S	Number of segments
s	Index of a segment
$\boldsymbol{\theta}_k$	Parameter of the k-th distribution
$\boldsymbol{\pi}_s$	Mixing coefficient vector for segment s
Λ	$(\{\boldsymbol{\pi}_s\}_{s=1,\ldots,S}, \{\boldsymbol{\theta}_k\}_{k=1,\ldots,K})$
x_{sn}	n-th data for the s-th segment
N_s	Number of observations in the s-th segment
X_s	Set of data observed in the s-th segment, i.e., $X_s = \{x_{s1}, x_{s2}, \ldots, x_{sN_s}\}$
X	Whole set of data, i.e., $X = \{X_1, \ldots, X_S\}$
$d(s,x)$	Divergence of the current traffic state from the usual state when the value x is observed for the s-th segment
D	Degree of anomaly for a trajectory or segment

3.1. Traffic state model

Intuitively, we can identify some traffic states as being "smooth" or "congested", regardless of the location. Vehicles travel fast in smooth states and behave in a stop-and-go fashion in heavily congested states. When observing the speed of a probe car, the value is likely to be small if the car is in "congested" traffic or large if the traffic is "smooth." The value will also be affected by geographical conditions such as curves and slopes. In short, the behavior of a car is affected by the surrounding traffic state, and the observed values for the probe car will change, whereas the traffic state is latent and varies according to the time and place. This relation between traffic states and PCD can be modeled using a probabilistic topic model [7].

Traffic-state information is strongly related to geographical and time-of-day conditions. We therefore introduce the *segment* as the unit for traffic observation. A segment is specified as a certain section of a route during a certain period. In this paper, we estimate at a fine level of granularity, for which we define a segment as a certain 50-m length of roadway for a certain direction on a certain expressway route for a one-hour period regardless of the day. This is described in more detail in the Experiment section. PCD includes timestamps and location data, which are obtained via GPS and are represented by longitude and latitude, enabling each probe-car observation to be assigned to a particular segment.

PCD also includes information about values such as speed and direction that can be recorded directly in the PCD or calculated from sequential observations. Here, all the observations are aggregated for each segment, and a set X_s of the observed data for the s-th segment is obtained. The symbol x_{sn}, the n-th value of X_s, might have either a scalar or a vector value. For simplicity, in this paper, we assume that x_{sn} is a scalar value, but our method could be extended to observed vector values.

Our model associates a traffic state with a probability distribution. Let K be the number of states, with the k-th traffic state corresponding to the parameter $\boldsymbol{\theta}_k$. The probability distribution for the s-th segment, given by $p(x|s)$, is described in terms of a mixture of these K distributions and can be described as follows:

$$p(x|s) = \sum_{k=1}^{K} \pi_{sk} p(x|\boldsymbol{\theta}_k), \tag{1}$$

where π_{sk} is the mixing coefficient for the k-th state and satisfies the conditions:

$$0 \le \pi_{sk} \le 1, \quad \sum_{k=1}^{K} \pi_{sk} = 1 \tag{2}$$

for each s. The state parameters $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K\}$ are identical for all segments, but the mixing coefficient vector $\boldsymbol{\pi}_s = (\pi_{s1} \cdots \pi_{sK})^{\mathsf{T}}$ is different for each segment. By using a global $\boldsymbol{\theta}_k$, we can compare and characterize segments in terms of local $\boldsymbol{\pi}_s$. For example, straight sections are dominated by "smooth" states, with sections that include tollgates being dominated by "congested" states.

Finally, for each segment, the generative process for this model is as follows.

1. Choose a hidden state $k \sim$ multinomial probability distribution Multi($\boldsymbol{\pi}_s$).
2. Generate the value $x_{sn} \sim p(x_{sn}|\boldsymbol{\theta}_k)$.

3.2. Parameter estimation

Our model is described by a mixture distribution, with its maximum-likelihood parameters estimated by an EM algorithm. It uses X, the set of observed PCD, as training data [44]. For simplicity, we introduce the symbol Λ as the set of all parameters in the model. For the entire set X of observed data, the likelihood under the model introduced above is given by the following equation:

$$L(X) = \prod_{s=1}^{S} \prod_{n=1}^{N_s} \sum_{k=1}^{K} \pi_{sk} p(x_{sn}|\boldsymbol{\theta}_k). \tag{3}$$

The update equations are derived by considering the maximization of the following Q function under constraint (2):

$$Q(X, \Lambda, \hat{\Lambda}) = \sum_{s=1}^{S} \sum_{n=1}^{N_s} \sum_{k=1}^{K} p(k|x_{sn}, \hat{\Lambda}) \log p(k, x_{sn}|\Lambda), \tag{4}$$

where

$$p\left(k \middle| x_{sn}, \hat{\Lambda}\right) = \frac{\hat{\pi}_{sk} p(x_{sn}|\hat{\boldsymbol{\theta}}_k)}{\sum_{k=1}^{K} \hat{\pi}_{sk} p(x_{sn}|\hat{\boldsymbol{\theta}}_k)} \equiv \gamma_{snk}, \tag{5}$$

$$p(k, x_{sn}|\Lambda) = \pi_{sk} p(x_{sn}|\boldsymbol{\theta}_k), \tag{6}$$

and $\hat{\Lambda}$ refers to the parameters estimated in the previous EM iteration. This Q is maximized by introducing Lagrange multipliers and setting its partial derivative to zero. The update equation for $\boldsymbol{\pi}_s$ is then derived as follows:

$$\pi_{sk} = \frac{\sum_{n=1}^{N_s} \gamma_{snk}}{N_s}. \tag{7}$$

This $\boldsymbol{\pi}_s$ does not depend on p, which means that the update equation will not be changed when the probability distribution used in the model is modified. On the other hand, the update equation for $\boldsymbol{\theta}_k$ is derived by solving the

equation:

$$\sum_{s=1}^{S}\sum_{n=1}^{N_s}\frac{\gamma_{snk}}{p(x_{sn}|\boldsymbol{\theta}_k)}\frac{\partial}{\partial\boldsymbol{\theta}_k}p(x_{sn}|\boldsymbol{\theta}_k)=0. \tag{8}$$

For the remainder of this paper, we assume that x_{sn} is the vehicular speed in km/h and is a nonnegative integer. We also assume a Poisson distribution for p:

$$p(x_{sn}|\boldsymbol{\theta}_k)\equiv p(x_{sn}|\lambda_k)=\frac{\lambda_k^{x_{sn}}e^{-\lambda_k}}{x_{sn}!}, \tag{9}$$

where λ_k is both the mean and variance, and is the only parameter of p. In this case, by solving Eq. (8), the update equation for λ_k is derived as

$$\lambda_k=\frac{\sum_{s=1}^{S}\sum_{n=1}^{N_s}\gamma_{snk}x_{sn}}{\sum_{s=1}^{S}\sum_{n=1}^{N_s}\gamma_{snk}}. \tag{10}$$

We now have an EM algorithm for estimating the parameters of our traffic state model. In this algorithm, after generating Λ at random, the EM iteration alternates between the E step, which calculates all γ_{snk} using Eq. (5), and the M step, which updates Λ according to Eqs. (7) and (8), until the log likelihood $\log L(X)$ converges. The computational complexity of the algorithm is $O(NK)$ per iteration, where N denotes the total number of observed values, i.e., $N=\sum_s N_s$.

3.3. Incident detection

Our basic approach to the traffic incident detection problem is to compare the CTS, which are estimated using real-time data, with the UTS, which are learned using archival data, and to detect divergence between them as a traffic incident, i.e., a sudden and unusual traffic event. Fig. 1 shows an example of incident detection. Assume that a probe car travels along a route and observes its speed and any other feature values for each segment that it passes through. Our AID method measures the degree of anomaly for each segment and detects incidents by finding high-divergence trajectories or segments. In this section, we propose functions for evaluating the divergence between the CTS and the UTS, and algorithms for detecting traffic incidents based on quantitative divergence values.

3.3.1. Divergence functions

We introduce several divergence functions to evaluate quantitatively the difference between the CTS and the UTS. A divergence function $d(s,x)$ returns the degree of anomaly when x is observed in the s-th segment. The value for $d(s,x)$ should be large if the observed value is anomalous and small if the observation is a usual one. We require the function to have additivity, so that the divergence of consecutive observations can be quantified as the summation of the divergences for each observation.

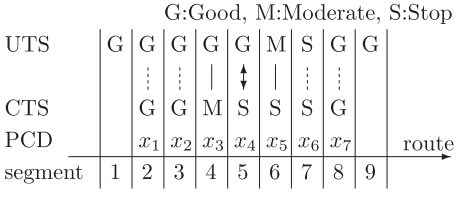

Fig. 1. Example of our incident detection method, which compares the CTS with the UTS.

A naïve definition of the function exploits the probability of observing x in the s-th segment. In general, the probability is high if the observation is as usual, while the probability is low if the observation is anomalous. Therefore, we can define $d(s,x)$ as the negative log probability of the observation, which satisfies the additivity requirement:

$$d(s,x)=-\log p(x|s)=\log\left(\sum_{k=1}^{K}\pi_{sk}p(x|\boldsymbol{\theta}_k)\right). \tag{11}$$

We can also define $d(s,x)$ by considering the traffic states. Because the parameters of the model are estimated to fit the distribution in the dataset, the learned model itself reflects the usual traffic over the whole observation period. In a segment s, we will have a distribution of traffic states $p(k|s)$ after the parameter estimation. When an observed value x is given for the segment, we can compute the posterior distribution of traffic states $p(k|s,x)$ using Bayes' theorem:

$$p(k|s,x)\propto p(k|s)p(x|s,k). \tag{12}$$

One possible definition of $d(s,x)$ compares $p(k|s,x)$ with $p(k|s)$ using a measure of the difference between two distributions, such as the Kullback–Leibler (KL) divergence:

$$d(s,x)=\mathrm{KL}(p_s\,\|\,p_{sx}), \tag{13}$$

where $\mathrm{KL}(p\,\|\,q)$ is the KL divergence of a distribution q from a distribution p, and where $p_s(k)=p(k|s)$ and $p_{sx}(k)=p(k|s,x)$.

Now, consider the intuitive idea illustrated in Fig. 1. There must be differences in travel behavior between vehicles, and the use of the probability distributions described above may be susceptible to them. Our approach considers the usual and current traffic states, which are discretized and robust toward such differences, and compares them to identify unusual events. We can therefore define the *usual traffic state* for the s-th segment, denoted by $\sigma(s)$, as the probable state:

$$\sigma(s)=\arg\max_k p(k|s)=\arg\max_k \pi_{sk}. \tag{14}$$

Meanwhile, the *current traffic state*, when x is observed in the s-th segment, is denoted by $\sigma(s,x)$ and can be defined as the probable state given x and s. It can be estimated as

$$\sigma(s,x)=\arg\max_k\{\pi_{sk}p(x|\boldsymbol{\theta}_k)\}. \tag{15}$$

For example, the usual state $\sigma(s)$ may indicate smooth traffic in a straight midnight segment, congested traffic in a rush-hour segment, or stop-and-go traffic in segments that contain tollgates for any time of day. If $\sigma(s)$ indicates congested traffic and $\sigma(s,x)$ is also congested, the current traffic remains usual and would not be considered an anomaly. If the usual state $\sigma(s)$ indicates free-flowing traffic and the current state $\sigma(s,x)$ indicates stop-and-go traffic, then it would be suspected that an incident had occurred. Because, in our model, each state is associated with a probability distribution $p(x|\boldsymbol{\theta}_k)$, we measure the difference of $\sigma(s,x)$ from $\sigma(s)$ in terms of the KL

divergence of the two distributions:

$$d(s,x) = \mathrm{KL}(p_{\sigma(s)} \| p_{\sigma(s,x)}), \tag{16}$$

where p_k denotes the probability distribution of observed data x in the k-th state, i.e., $p(x|\boldsymbol{\theta}_k)$. In this paper, as mentioned above, we assume a Poisson distribution for p. The KL divergence between two Poisson distributions p_k and p_l, whose parameters are denoted by λ_k and λ_l respectively, can be derived as

$$\mathrm{KL}(p_k \| p_l) = \lambda_l - \lambda_k + \lambda_k \log \frac{\lambda_k}{\lambda_l}. \tag{17}$$

As previously stated, the parameter λ is the mean of the distribution.

There are several alternative measures of the difference between two probability distributions. Because the KL divergence is asymmetric, we could use the inverse KL divergence instead of the KL divergence:

$$d(s,x) = \mathrm{KL}(p_{\sigma(s,x)} \| p_{\sigma(s)}). \tag{18}$$

Another alternative is the Jensen–Shannon (JS) divergence, which is a symmetric measure:

$$d(s,x) = \mathrm{JS}(p_{\sigma(s)}, p_{\sigma(s,x)}), \tag{19}$$

where $\mathrm{JS}(p_k, p_l)$ is the JS divergence:

$$\mathrm{JS}(p_k, p_l) = \tfrac{1}{2}\,\mathrm{KL}(p_k \| q) + \tfrac{1}{2}\,\mathrm{KL}(p_l \| q) \quad \text{where } q = \tfrac{1}{2}(p_k + p_l). \tag{20}$$

The JS divergence between two Poison distributions cannot be given by a closed-form expression, but the series converges rapidly, and an approximation can be calculated numerically by summing up the first 100–200 terms when both λ_k and λ_l are smaller than 100.

The Hellinger distance is another symmetric measure:

$$d(s,x) = \mathrm{Hellinger}(p_{\sigma(s)}, p_{\sigma(s,x)}), \tag{21}$$

where $\mathrm{Hellinger}(p_k, p_l)$ is the Hellinger distance. For two Poisson distributions, it can be derived as

$$\mathrm{Hellinger}(p_k, q_l) = 1 - \exp\left(-\tfrac{1}{2}\left(\sqrt{\lambda_k} - \sqrt{\lambda_l}\right)^2\right). \tag{22}$$

So far, we have defined $d(s,x)$ as the distance between the probability distributions of $\sigma(s)$ and $\sigma(s,x)$. We defined the usual traffic state of the s-th segment $\sigma(s)$ as its most probable state, under the assumption that a segment should be dominated by a single state. For example, assume that $\pi_{s1} = 0.49$ and $\pi_{s2} = 0.48$ for the s-th segment. The usual traffic state $\sigma(s)$ would be assumed to be the first state, because it is the most probable, even though the second one is almost equally probable. If the two states are very different and the estimated current state $\sigma(s,x)$ is the second one, the $d(s,x)$ defined above will be large. However, the traffic state model is a mixture model, and we can consider a superposition of traffic states to avoid this problem. Given an observed value x, the probability of the usual state is $p(k|s)$ and the probability for the current state to be l is $p(l|s,x)$. Therefore, the probability of observing the divergence between the states k and l is $p(k|s)p(l|s,x)$, and we can define the divergence $d(s,x)$ as the weighted sum of the divergences for each state pair:

$$d(s,x) = \sum_{k=1}^{K}\sum_{l=1}^{K} p(k|s)p(l|s,x)\mathrm{KL}(p_k \| p_l), \tag{23}$$

where the divergence between two states is measured in terms of KL divergence. Using $p(k|s) = \pi_{sk}$, this equation can be transformed as follows:

$$d(s,x) = \boldsymbol{\pi}_s^{\mathsf{T}} \Delta \boldsymbol{\pi}(s,x), \tag{24}$$

where

$$\Delta = \begin{pmatrix} \mathrm{KL}(p_1 \| p_1) & \cdots & \mathrm{KL}(p_1 \| p_K) \\ \vdots & \ddots & \vdots \\ \mathrm{KL}(p_K \| p_1) & \cdots & \mathrm{KL}(p_K \| p_K) \end{pmatrix}, \tag{25}$$

$$\boldsymbol{\pi}(s,x) = (p(1|s,x)\ \ p(2|s,x)\ \ \cdots\ \ p(K|s,x))^{\mathsf{T}}. \tag{26}$$

Because $\boldsymbol{\pi}_s$ and Δ are independent of x, we can obtain the transposed K-vector $\boldsymbol{\pi}_s^{\mathsf{T}}\Delta$, denoted by $\boldsymbol{\delta}_s^{\mathsf{T}}$, just after the end of the period of traffic-state-model learning. When x is observed, the divergence defined in Eq. (23) is calculated as the dot product of $\boldsymbol{\delta}_s$ and $\boldsymbol{\pi}(s,x)$. Of course, we could use inverse KL divergence, JS divergence, or Hellinger distance instead of KL divergence when $\boldsymbol{\delta}_s$ is calculated.

3.3.2. Detection algorithms

We defined $d(s,x)$ to quantify the degree of anomaly for each observation. In essence, our detection algorithm uses $d(s,x)$ to calculate a degree of anomaly, denoted by D, and gives an alert when D exceeds a given threshold. We introduce alternative definitions for D based on two different approaches, namely, a trajectory-based (TB) approach and a segment-based (SB) approach. Fig. 2 shows an example of the two approaches.

TB approach: In this approach, the detection algorithm attempts to find the trajectory of a probe car whose behavior is abnormal. A probe car travels along a route, and we measure the divergence $d(s,x)$ for each segment

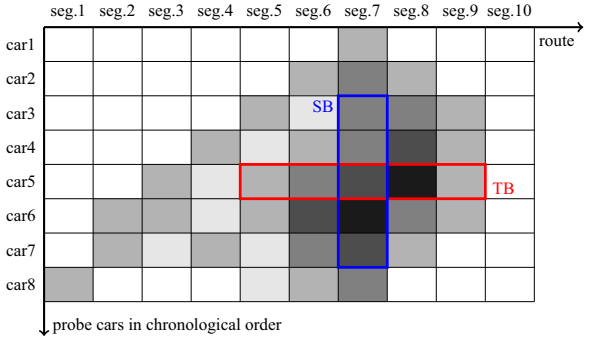

Fig. 2. Example of trajectory-based (TB) and segment-based (SB) approaches. The shades of gray indicate the value of d, the degree of anomaly, which is calculated for each vehicle for each segment. The red rectangle is a sliding window used in the TB approach, and the blue rectangle is used in the SB approach. The length of the "TB" sliding window is constant, and the sum of the d values in the sliding window is evaluated to detect incidents. The length of the "SB" sliding window can vary in this view because the sliding window is defined using a time parameter, and the average of the d values in the sliding window is evaluated.

traversed. Assume that $\boldsymbol{d}$ is a sequence of measured $d(s,x)$ values, $N_{\boldsymbol{d}}$ is the number of observations in $\boldsymbol{d}$, and d_i is the i-th value in $\boldsymbol{d}$. The oldest observation is d_1, and the latest is $d_{N_{\boldsymbol{d}}}$. Because we have defined $d(s,x)$ to have the property of additivity so that the divergence of consecutive observations can be quantified as the summation of the divergences for each observation, the divergence of the whole trajectory can be defined as follows:

$$D = \sum_{i=1}^{N_{\boldsymbol{d}}} d_i. \tag{27}$$

The more a car deviates from its usual behavior, the larger D will be. However, if D is defined as above, it will continue to increase as long as the car runs, and any trajectory would be determined eventually as being anomalous. Taking an average over the whole trajectory can avoid this problem, but it is still a problem that the value for D cannot be obtained until the probe car finishes traveling. It is also a problem that the sphere of influence of an incident cannot be specified. Therefore, we redefine D as the sum of N consecutive divergences:

$$D = \sum_{i=i_0}^{i_0+N-1} d_i. \tag{28}$$

This version of D can be calculated in real time by maintaining d_i using N-length sliding windows and using its summation [9].

SB approach: In this approach, the detection algorithm monitors the divergences over time for each segment and gives an alert when the divergence is sufficiently high. Assume that several probe cars have passed through a segment s for the past T seconds. The degree of anomaly $d(s,x)$ can be calculated for each observation, respectively. Let $\mathrm{SW}(s,T)$ be the set of d values. Then SW can be considered to be a sliding window over the time T, which is drawn as a blue rectangle in Fig. 2. Note that $|\mathrm{SW}(s,T)|$, the cardinality of the set SW, is equivalent to the number of probe cars traversing the segment s for the past T seconds, and can therefore vary according to the time. We now define D as the sum of d in SW:

$$D = \sum_{d \in \mathrm{SW}(s,T)} d. \tag{29}$$

However, this D will also increase as the traffic becomes dense, i.e., as the number of probe cars traversing the segment within T seconds increases. We therefore revise the definition of D to take an average over the probe cars in the sliding window:

$$D = \frac{1}{|\mathrm{SW}(s,T)|} \sum_{d \in \mathrm{SW}(s,T)} d. \tag{30}$$

Whereas the TB approach tries to find abnormal behavior by a probe car, the SB approach is designed to simulate roadside-sensor-based traffic monitoring via probe-car data. A sliding window in the SB approach continuously watches a certain segment so that the detector can output the time and place of an incident.

4. Experiment

4.1. Dataset and preprocessing

Our PCD was obtained from probe cars traveling on the Shuto Expressway system during 2010 and 2011. The PCD include several tens of millions of observations. We first conducted data preprocessing, which comprised five phases: segment definition, map matching, trajectory identification, interpolation, and labeling. These procedures are described below.

Segment definition: As defined in Section 3.1, a certain segment corresponds to a certain 50-m length of roadway for a certain direction on a certain expressway route for a one-hour period regardless of the day. We defined the road segments by partitioning each route on the expressway every 50 m. The direction was noted. We also divided each day into 24 h and associated each segment with one-hour periods. This experiment did not consider days of the week.

Map matching: Although the above definition of a segment is based on an expressway route, the location data in the PCD were described in terms of the two-dimensional coordinates of longitude and latitude, with the original observation not related to any particular segment. Map matching is a technology for identifying the road segment on which the vehicle is traveling and for locating the vehicle within that segment [45]. Several map matching methods have been proposed [46,47]. In this experiment, map matching was conducted in the simplest way: a probe car's observation was matched with the nearest segment to the car's location. The direction was estimated from the angular difference between the probe car's heading azimuth in the PCD and the segment's azimuth for each direction, and choosing the direction that gave the smaller angle. We used the National Land Numerical Information (NLNI)[1] as the roadmap data. This set of data contains linestrings, which represent the shape of expressways and toll roads. The shape of junctions is not represented strictly, and road-width information is not included. After the map matching, we removed noisy records, namely, those more than 100 m from the nearest road. An observation was also removed if the angular difference between its heading azimuth and the segment's azimuth was more than 45°.

Trajectory identification: To identify the continuous movement of the car, i.e., its trajectory, we grouped all observations in the dataset of records by anonymized car ID and sorted them by timestamp for each group, before concatenating them in chronological order whenever the time gap between two consecutive observations was 60 s or less.

Interpolation: We used a probe car's speed as the observed value in this experiment. However, as mentioned in Section 3.3, our detection method estimated the current state for each trajectory for each segment that the car had traversed. Our 50-m segment was too short for fast-moving probe cars to provide observations for every segment, whereas a slow-moving car could generate multiple observations in a single segment. Accordingly, we used the mean speed as the observation value for each segment for each

[1] http://nlftp.mlit.go.jp/ksj/

Table 2
Statistics for the dataset.

Route	Direction	Length (km)	# in part (a)/(b)/(c)					
			Segments	Observations	Trajectories	Incidents	Observations in incidents	Trajectories in incidents
(1) Ueno	Inbound	4.30	85/85/85	792078/454175/511037	20985/11728/13568	0/11/22	0/310/633	0/39/74
(1) Haneda	Inbound	12.58	250/250/250	6479136/3991898/3889471	74022/42323/43674	0/311/435	0/32234/44159	0/1273/1766
(1) Haneda	Outbound	12.58	250/250/250	6246959/3788658/3665425	70453/40219/41482	0/116/203	0/9246/17460	0/440/779
(2) Meguro	Inbound	5.70	112/112/112	1812270/1038831/954443	23782/13332/12260	0/71/98	0/2829/3468	0/166/185
(2) Meguro	Outbound	5.70	112/112/112	1463662/872699/736948	17354/10280/8678	0/15/14	0/503/321	0/21/18
(3) Shibuya	Inbound	11.70	232/232/232	9490711/5569544/5369891	65065/38621/38479	0/425/609	0/51908/74377	0/1640/2381
(3) Shibuya	Outbound	11.70	233/233/233	8630888/5255259/5162198	62734/38563/39184	0/332/403	0/38110/51231	0/1103/1570
(4) Shinjuku	Inbound	13.27	264/264/264	10681788/6252381/6093679	70500/40900/40431	0/394/514	0/68705/73368	0/2061/2215
(4) Shinjuku	Outbound	13.27	264/264/264	10142665/6132584/5967115	68588/40519/39194	0/436/449	0/82656/71391	0/2093/1963
(5) Ikebukuro	Inbound	21.10	420/420/420	17186588/9907871/9445068	110929/64056/62187	0/510/702	0/69114/115972	0/2354/3646
(5) Ikebukuro	Outbound	21.10	420/420/420	16416424/9335481/9086078	103907/59799/58265	0/568/816	0/58386/114376	0/2120/3579
(6) Misato	Inbound	10.23	203/203/203	6479441/3786663/3619748	44353/25326/24532	0/232/311	0/35096/62137	0/863/1369
(6) Misato	Outbound	10.23	203/203/203	7015915/4090983/4006477	44909/26295/25354	0/137/144	0/15521/16618	0/464/607
(6) Mukojima	Inbound	9.41	187/187/187	4944680/3102482/2611225	78108/46307/41934	0/226/274	0/17679/22290	0/927/1213
(6) Mukojima	Outbound	9.41	187/187/187	6304960/3691151/3489748	78388/45684/43391	0/377/441	0/32256/45488	0/1452/1811
(7) Komatsugawa	Inbound	11.19	222/222/222	4102884/2507305/2208502	39227/23071/20044	0/124/176	0/10309/12643	0/438/547
(7) Komatsugawa	Outbound	11.19	222/222/222	4084778/2609988/2259405	29087/18868/16264	0/35/35	0/2142/3433	0/81/111
(9) Fukagawa	Inbound	5.35	105/105/105	2427724/1327255/1362920	30616/16343/16780	0/93/178	0/4482/9070	0/243/436
(9) Fukagawa	Outbound	5.35	105/105/105	1991012/1043268/1111541	24652/13135/13532	0/28/10	0/456/395	0/46/21
(11) Daiba	Inbound	3.68	71/71/71	1505294/818800/837530	25453/13523/14222	0/94/158	0/3543/8010	0/220/428
(11) Daiba	Outbound	3.68	72/72/72	2338885/1274658/1394220	34977/19043/20981	0/28/36	0/1384/1406	0/76/88
(C1) Inner circular	Counterclockwise	14.00	278/278/278	10987711/6478632/6125324	228108/132148/127611	0/993/1181	0/59382/83258	0/3453/4710
(C1) Inner circular	Clockwise	14.00	278/278/278	11220562/6521415/6162208	212109/123424/117349	0/1038/1309	0/71823/86398	0/3380/3961
(C2) Central circular (west)	Counterclockwise	10.79	214/214/214	4380668/3173120/3295236	39442/25426/25621	0/276/444	0/15142/27339	0/649/1139
(C2) Central circular (west)	Clockwise	10.79	214/214/214	3340908/2735642/2973822	31195/22465/23639	0/235/350	0/13478/21081	0/597/946
(C2) Central circular (east)	Counterclockwise	25.22	501/500/500	16248358/9138931/9122772	104236/59352/58779	0/480/592	0/75190/125806	0/2093/3245
(C2) Central circular (east)	Clockwise	25.22	501/501/501	17636512/10035085/9935318	107118/61243/60511	0/577/745	0/51813/86449	0/1480/2224
(Y) Yaesu	Northbound	1.55	30/30/30	193661/107788/104587	7321/4057/3940	0/29/54	0/313/536	0/38/70
(B) Bayshore (west)	Eastbound	37.91	755/755/755	19365358/11199556/11705000	84020/48877/51501	0/121/191	0/14131/25462	0/499/792
(B) Bayshore (west)	Westbound	37.91	757/757/757	17326082/10325601/10673341	73703/41633/43200	0/121/156	0/8304/18252	0/263/533
(B) Bayshore (east)	Eastbound	24.01	479/479/479	21959324/11994826/13071082	119437/64043/70639	0/262/499	0/32253/74072	0/1066/2075
(B) Bayshore (east)	Westbound	24.01	479/479/479	23047168/12873257/13746786	121868/65826/72143	0/228/492	0/63681/142289	0/1505/3422
(S1) Omiya	Inbound	8.25	163/163/163	2893492/1616952/1679369	25837/14486/14005	0/4/17	0/101/312	0/7/25

Table 2 (*continued*)

Route	Direction	Length (km)	# in part (a)/(b)/(c)					
			Segments	Observations	Trajectories	Incidents	Observations in incidents	Trajectories in incidents
(S1) Omiya	Outbound	8.25	163/163/163	2570384/1563298/1571199	22626/13578/13384	0/3/17	0/41/555	0/3/29
(S5) Kawaguchi	Inbound	12.09	240/240/240	8215563/4703713/4738225	48346/26801/27563	0/172/237	0/27794/54694	0/657/1151
(S5) Kawaguchi	Outbound	12.09	240/240/240	8478819/4858376/5055100	45730/25762/26497	0/140/211	0/11734/20666	0/390/657
(K1) Yokohane	Inbound	19.27	383/384/383	10245910/6241855/6252839	64879/38222/39391	0/128/161	0/10197/16243	0/343/543
(K1) Yokohane	Outbound	19.27	383/383/383	10467448/6390554/6304514	66510/39667/39318	0/143/242	0/7149/18916	0/332/712
(K2) Mitsusawa	Inbound	2.53	49/49/49	1253797/749791/749555	33744/20380/20174	0/18/33	0/624/3168	0/47/162
(K2) Mitsusawa	Outbound	2.53	49/49/49	935405/552126/569965	25404/14951/15499	0/11/37	0/466/1459	0/33/93
(K3) Kariba	Inbound	8.91	177/177/177	3733713/2208095/2195942	35741/21461/21890	0/44/55	0/2352/5760	0/119/178
(K3) Kariba	Outbound	8.91	177/176/177	3218503/1943342/1934921	36676/21567/21791	0/52/118	0/2641/8995	0/130/374
(K5) Daikoku	Inbound	3.64	67/67/67	365049/215077/245619	5870/3425/3896	0/12/9	0/311/109	0/19/10
(K5) Daikoku	Outbound	3.64	69/69/69	398590/239082/243637	6068/3625/3700	0/2/6	0/22/39	0/2/6

In some cases, additional segments were recognized because the shape information of the roadmap data we used is incomplete at junctions. The nonexistent segments were rarely recognized by the map-matching process.

trajectory. This was obtained from the time required for the car to pass through the 50-m segment after we calculated the time when the car entered and left any segment by linear interpolation. Again, we removed noisy trajectories that traversed less than 10 segments, i.e., a 500-m distance.

Labeling: For the evaluation, we labeled each observation in the trajectories, using the traffic log made available by the administrator of the Shuto Expressway. This traffic log was recorded via stationary sensors on or alongside the roads every five minutes, together with manual annotations about incidents such as accidents and construction. An observation was labeled as anomalous whenever the stationary sensor nearest to the segment recorded an incident at that time.

According to NLNI, the Shuto Expressway system comprises 29 routes, including short branches, and every route has traffic in both directions. Because the regulation and traffic patterns are different for each route, we partitioned our dataset to account for each direction on each route. We removed route data involving less than two actual incident occurrences. Because we did not have ground truth data for 2010, we conducted a cross-validation as follows. We first divided the data into three parts: (a) the data during 2010, (b) the data during the first half of 2011, and (c) the data during the second half of 2011. We then estimated the model using (a) beside either (b) or (c), with the detection test conducted using the remaining part. For the remainder of this paper, "Fold 1" denotes the trial with the training dataset (a) and (b), and "Fold 2" denotes the trial with the training dataset (a) and (c). Table 2 summarizes the statistical information for our PCD after the preprocessing. In the remainder of this section, we estimated a traffic state model using a whole training dataset so that the traffic state "topic" was learned globally among routes, whereas traffic incident detection was conducted independently for each route.

4.2. Parameter estimation

After the preprocessing, we estimated the parameters of our traffic state model for each training dataset. In this experiment, the observed values represented the speed of probe cars as nonnegative integers, and we assumed a Poisson distribution for the probability distribution corresponding to each traffic state. We implemented the EM algorithm described in Section 3.2 using OpenMP for multiprocessing. The estimation was executed on our 32-core Xeon computer.

We first examined the optimal value of K, the number of traffic states, using Akaike's information criterion (AIC). The optimal parameter value is that corresponding to the minimum AIC value, which means that the estimated model will achieve high likelihoods via a simple model, i.e., a model having few parameters. Fig. 3 shows the plots of the AIC for different values of K. The effect of model complexity was substantially less than that of the likelihood for improving the AIC, with the AIC value being almost the same for large K. We therefore assumed a value for K of 8 when conducting the experiment. Fig. 4 shows the log-likelihood of the model against the number of iterations of the EM algorithm, up to 20 iterations. In this experiment, we considered the algorithm to have converged when the improvement in log-likelihood fell below 0.01%, which was achieved after seven iterations. The execution of the EM algorithm up to the seventh iteration required about 18 min for both folds.

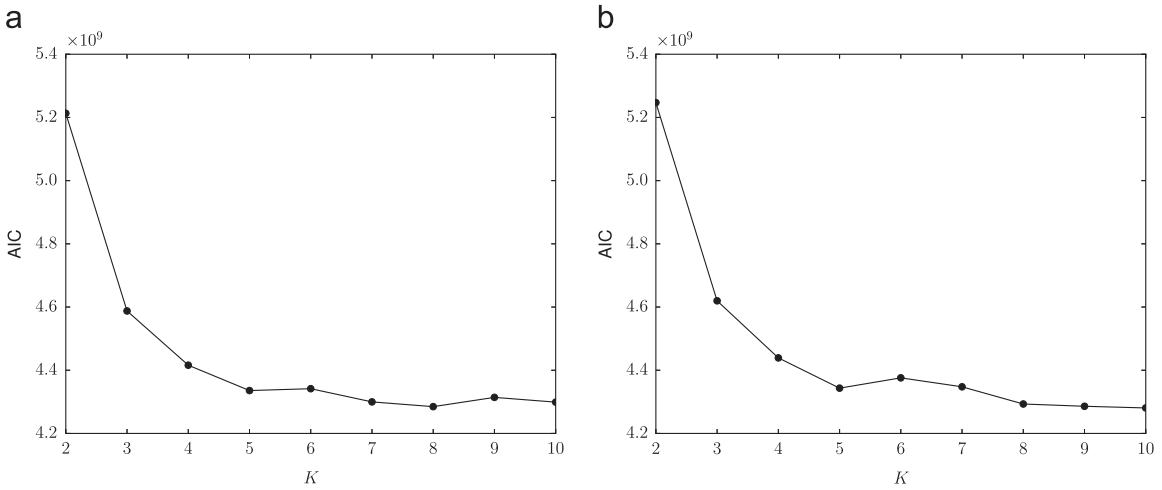

Fig. 3. Plot of AIC for different values of K, the number of traffic states: (a) Fold 1 and (b) Fold 2.

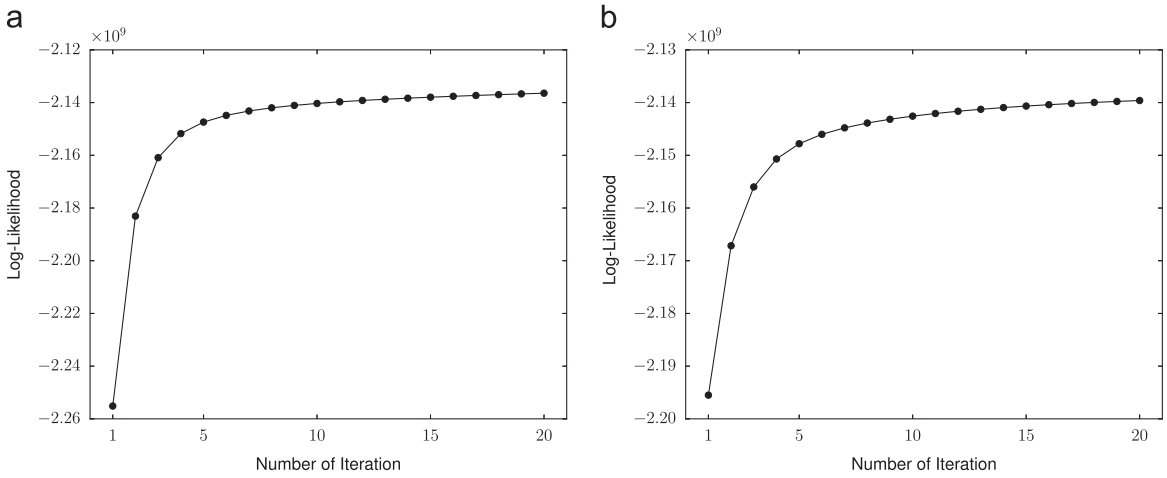

Fig. 4. Log-likelihood of the model against the number of the EM iterations, where K is 8: (a) Fold 1 and (b) Fold 2.

Fig. 5 shows the actual histogram for a segment of the inbound Shibuya route as a stepwise line chart and the estimated Poisson mixture as a solid curved line. Each of the eight Poisson distributions was multiplied by the mixing coefficients π_{sk}, which are also shown in Fig. 5 as dashed curves. We note that the estimated curve almost fits the actual histogram for the training dataset.

Fig. 6 shows the usual traffic state $\sigma(s)$ for each segment in the Shuto Expressway system, which was estimated using the learned traffic state model. The color of the segment indicates the parameter value for the Poisson distribution of $\sigma(s)$, i.e., the mean speed in the usual traffic state. Green represents high speed (100 km/h), red is moderate speed (50 km/h), and blue is "almost stopped" (0 km/h). The four figures (a)–(d) show the usual traffic over four different one-hour periods. We can see that the traffic is usually slow at several places during the rush hour, indicating that congestion usually occurs, whereas the traffic is almost smooth during the day and at midnight. Fig. 6(e) shows an enlarged view near the Iidabashi interchange on the

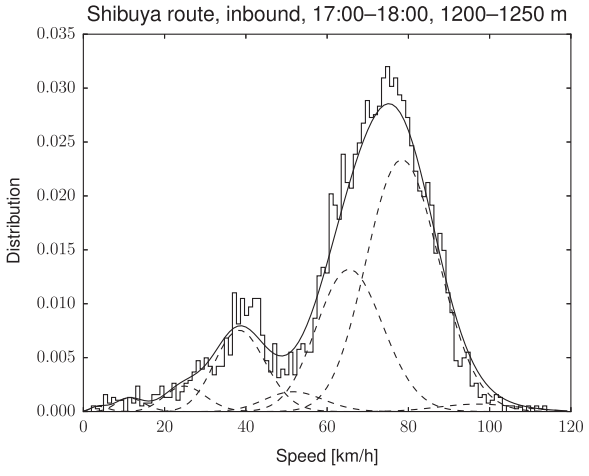

Fig. 5. Histogram of the speed of probe cars in a segment and the estimated Poisson mixture.

Fig. 6. Estimated usual traffic state $\sigma(s)$ for all 50-m segments, drawn on the OpenStreetMap. The color of a segment indicates the parameter value for the Poisson distribution of the maximum probable state, i.e., the mean speed of $\sigma(s)$. Green represents high speed (100 km/h), red is moderate speed (50 km/h), and blue is "almost stopped" (0 km/h) (map tiles © OpenStreetMap contributors, CC BY-SA 2.0). (a) 00:00–01:00, (b) 07:00–08:00, (c) 13:00–14:00, (d) 18:00–19:00 and (e) 18:00–19:00 near the Iidabashi interchange on the Ikebukuro Route.

Table 3
Tested parameter values.

Detection algorithm	Parameter values
TB	$N=1, 2, 3, 4, 5, 10, 15, 20, 25, 30$
SB	$T=60, 300, 600, 1800, 3600$

Ikebukuro Route between 18:00 and 19:00. The outbound direction is from the lower right to the upper left.[2] The figure shows that outbound traffic is usually slow near sharp curves and at the Iidabashi entrance in the center of the map, whereas the usual state of the other segments, including the inbound route, is "moderate" or "smooth." Here, it can be seen that the estimated traffic state model has enabled any road section during a certain period to be characterized using a single set of traffic states, with the usual pattern of traffic thereby being described at a fine level of time and space granularity.

4.3. Incident detection

Using the estimated traffic model, we evaluated the performance of the proposed method. Our detection method gives an alert when the divergence of an input set of observation values for the estimated traffic model exceeds a given threshold. We regarded a set that includes at least one value labeled as an anomaly to be a truly anomalous set. The two proposed algorithms require parameters N or T. Table 3 summarizes the parameter values used in this experiment.

The results shown in Section 4.2 indicated that the traffic pattern was different for each route, section, and time, and therefore the detection threshold must be changed accordingly. Although the granularity of such a threshold tuning should depend on the amount of data, we know of no method to determine the appropriate granularity. In this experiment, we conducted incident detection for each route, because the traffic characteristics were considered to be comparatively homogeneous on an individual route. We evaluate the selectivity performance of incident detectors in terms of a receiver-operating characteristic (ROC) curve. An ROC curve is drawn by plotting the true positive rate (TPR) against the false positive rate (FPR) at any threshold. Both TPR and FPR change according to the detection threshold: both values are zero when the threshold is high enough not to give an alert for any input, whereas they are equal to one when the threshold is low enough. With an ideal detector, TPR can be one with FPR being zero at a certain threshold. Therefore, the "area under the curve" (AUC) reflects the discrimination performance, with larger AUC values indicating better discrimination. For each route, we first applied the detection algorithms to our twofold dataset and obtained the values of divergence for each input. We therefore have possible values for the threshold at which either TPR or FPR changes. The ROC curve was drawn by plotting TPR against FPR in all cases.

We conducted a comprehensive experiment. The detection performance was examined by calculating AUC values

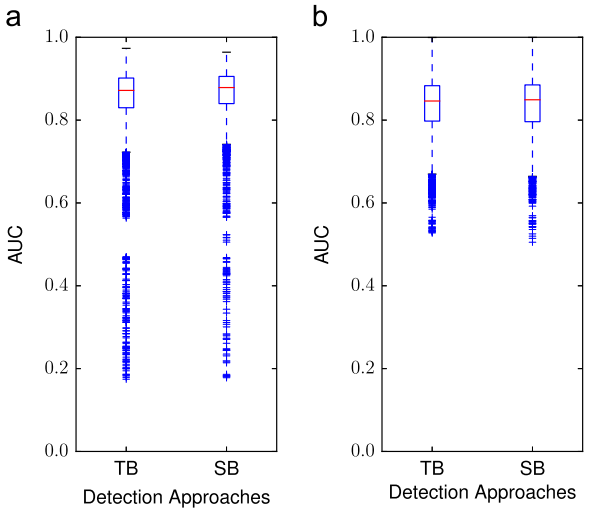

Fig. 7. Box plots of the AUC values for the proposed TB and SB methods. Each box plot describes the distribution of AUC values for all combinations of the parameters, divergence function, and route: (a) Fold 1 and (b) Fold 2.

respectively for all combinations of the algorithm (TB or SB), parameters (shown in Table 3), divergence function (proposed in Section 3.3.1), and route. At the beginning of the analysis of results, we first investigated the performance of the two algorithms proposed in Section 3.3.2. Fig. 7 shows box plots for the distribution of the AUC values, comparing the TB and SB approaches. Values less than $Q_1 - 1.5\mathrm{IQR}$ are plotted as outliers, where Q_1 is the first quartile and IQR is the interquartile range, the difference between the first and third quartiles. As seen in Fig. 7, there is no significant difference between the TB and SB algorithms' performance. Therefore, we decided to examine the performance further only in terms of the TB algorithm.

Next, we investigated the performance of the divergence functions proposed in Section 3.3.1. Fig. 8 shows the results in terms of box plots similar to those above. The results indicate that the weighted KL divergence, defined as Eq. (23), achieved the best performance among the functions because its AUC values were the most concentrated at high values and its the median was the highest. Therefore, we used this function for the remainder of the experiment.

We also examined the optimal value for the parameter N, the length of the sliding window, using weighted KL divergence as the divergence function. Fig. 9 shows the distribution of AUC values for each N. From this figure, the discrimination performance was almost unchanged as N increased. Fig. 10 shows the ROC curves for four cases when N was 10. Because we were using 50-m segments, the length of the sliding window was equivalent to a 500-m distance along a route.

As shown in Fig. 9, the AUC value was more than 0.8 in most cases. The best performance was achieved on the northbound Yaesu route, and the second best case was the outbound Kawaguchi route. The TPR reached 80%, with the FPR being less than about 3% in both cases, as shown in Fig. 10 (a) and (b). On the other hand, there were a couple of outlier

[2] Note that vehicles drive on the left side of the road in Japan.

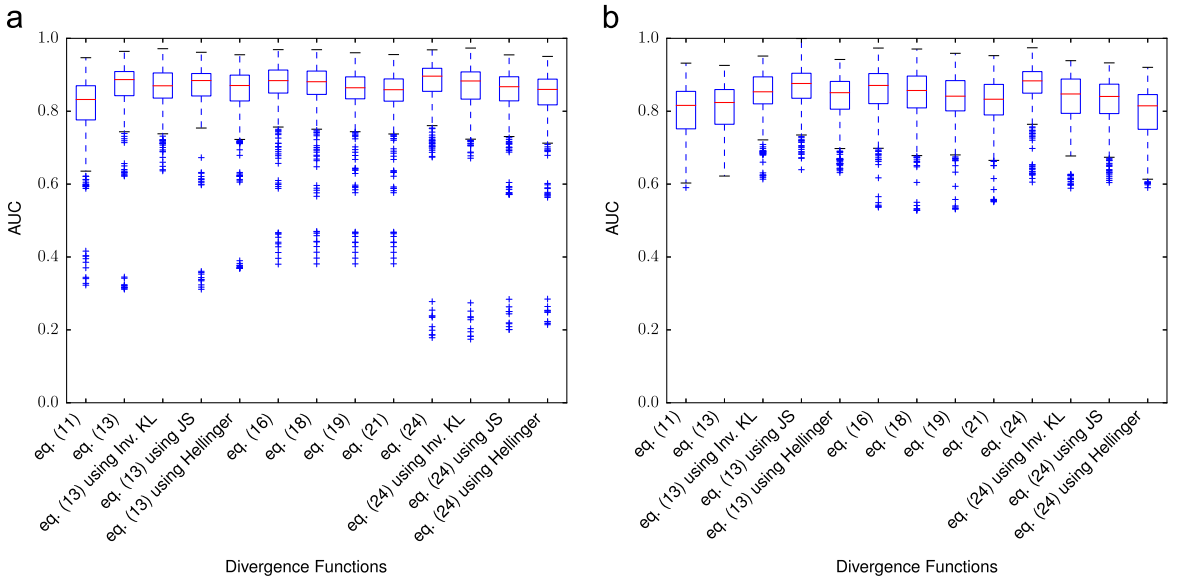

Fig. 8. Box plots of AUC values for the TB algorithm for each divergence function: (a) Fold 1 and (b) Fold 2.

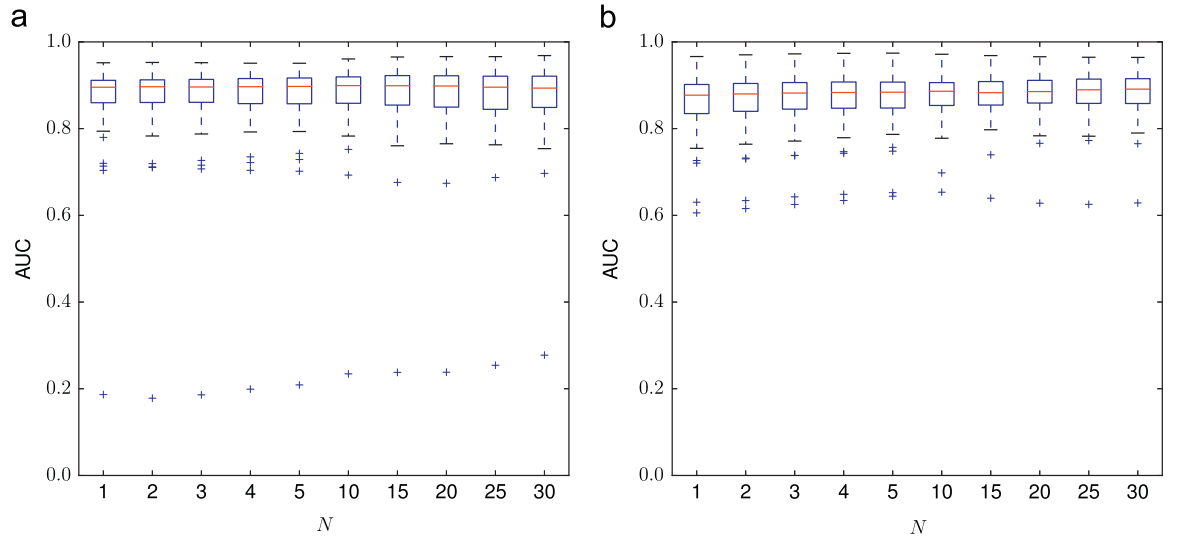

Fig. 9. Box plots of AUC values for the TB algorithm with the weighted KL divergence function for each N, namely, the number of consecutive segments in a sliding window: (a) Fold 1 and (b) Fold 2.

cases where the discrimination performance was extremely low. The worst case was the outbound Daikoku route, as shown in Fig. 10(c). The dataset for this route contains a very small number of actual incidents. Therefore, the detection performance through the dataset is greatly influenced by the characteristics of individual incidents. We investigated the dataset in detail, and we found trajectories that look as if the traveling behavior is different from the usual because of noise, whereas there was actually no incident at that time. The proposed method detected them as anomalies, and such misdetections influenced the detection performance evaluated by the ROC curves. The second worst case was the outbound Meguro route. After the TPR reached 40%, with the FPR being less than 1%, the ROC curve continued almost straight ahead to the upper right corner. From the dataset for this route, the proposed method could detect incidents with high precision when the threshold was high enough. When the threshold was decreased, the method detected other segments near incident-labeled segments as well as the abnormal segments.

4.4. Comparison with a previous method

So far, we have reported on the performance of the proposed method and obtained the best parameters and functions. We also conducted an experiment to compare our method with a baseline method, namely, the method of Zhu et al. [25], which was described in Section 2. The latter

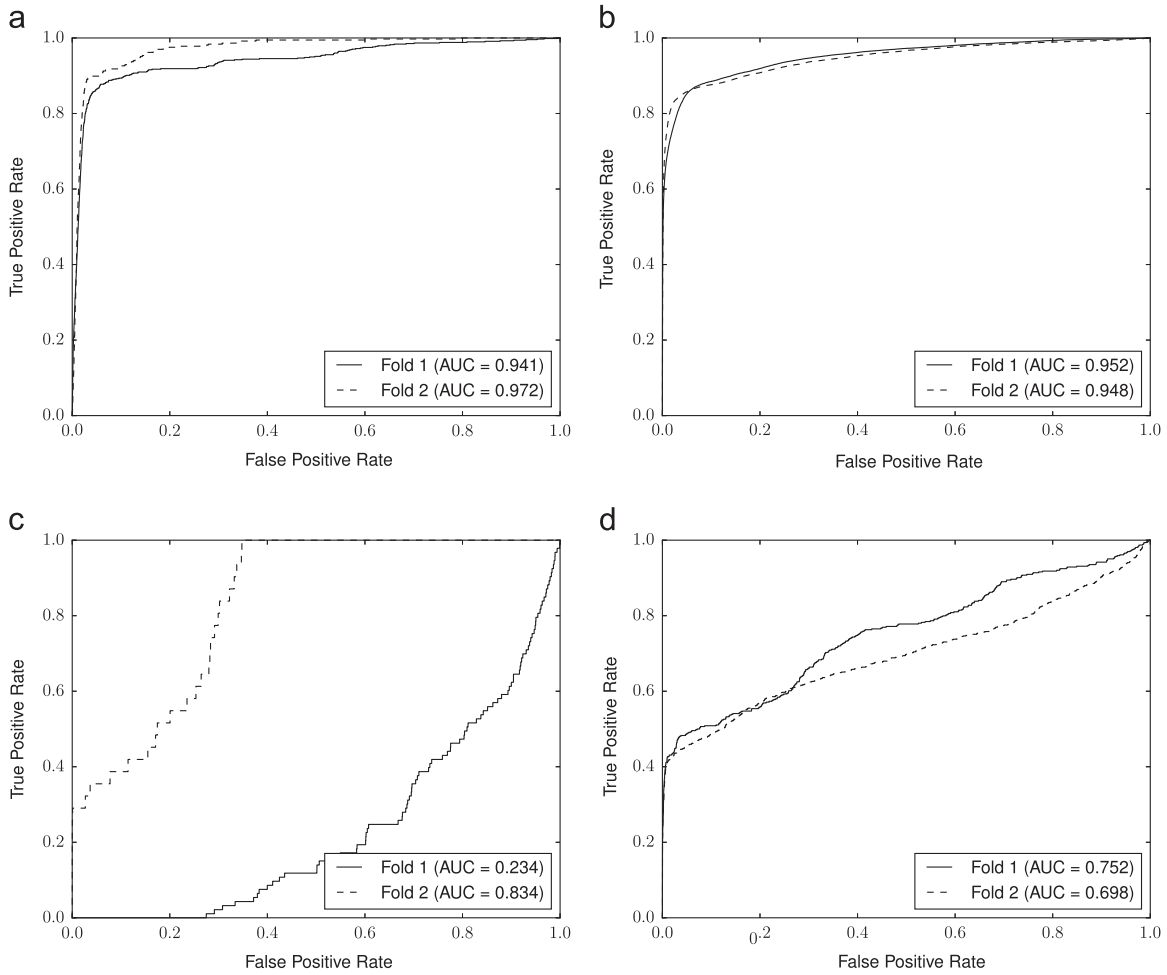

Fig. 10. Detection performances of the proposed method in terms of ROC curves. (a) The best case: the northbound Yaesu route, (b) the second best case: the outbound Kawaguchi route, (c) the worst case: the outbound Daikoku route and (d) the second worst case: the outbound Meguro route.

method finds traffic incidents by applying a distance-based outlier detection algorithm to feature vectors, which are carefully extracted using heuristics and normalized according to the mean and variance. The algorithm calculates the distance between any two feature vectors and detects outliers whenever the average distance of a vector to any other point is more than a given threshold. This algorithm runs in a batch manner, and its complexity is $O(N^2)$, where N is the number of observed feature vectors.

In this paper, we have tackled the problem of detecting traffic incidents in real time. We modified the baseline method to enable its application to our dataset in a streaming manner as follows. First, a queue was prepared to store the last n feature vectors. Variables for storing the sum and sum of squares of feature vectors for each dimension were also prepared, to enable the mean and variances to be calculated. Whenever a vehicular speed is observed, a feature vector is generated using both those data and data observed in the past, with the sum and sum of squares being updated. Any vectors in the queue in addition to the input vector are then normalized according to the mean and variance for each dimension before the distances between the input feature vector and any vectors in the queue are calculated. The algorithm outputs the average distance as the degree of anomaly for the input vector and detects whether it exceeds a given threshold. Then, if the queue is full, the oldest vector is removed, and the sum and the sum of squares are updated. Finally, the input vector is queued. We can infer that the detection performance is improved as the queue length n increases because the detector can then exploit more knowledge in determining anomalies. However, the algorithm will take longer time to execute because the computational complexity is $O(n)$ for each input, which might prevent the algorithm from working in real time.

We compared the baseline and proposed methods for three routes that were carefully selected. The chosen routes were the clockwise Inner circular route (Route C1), representing a slow-traffic ring road, the outbound Ikebukuro route (Route 5), representing a moderate-traffic radial road, and the eastbound Bayshore (east) route (Route B), representing a fast-traffic road. We implemented the baseline method using OpenMP to enable parallel execution with up to 32 threads.[3]

[3] Our proposed detector was implemented without involving parallel-processing technologies.

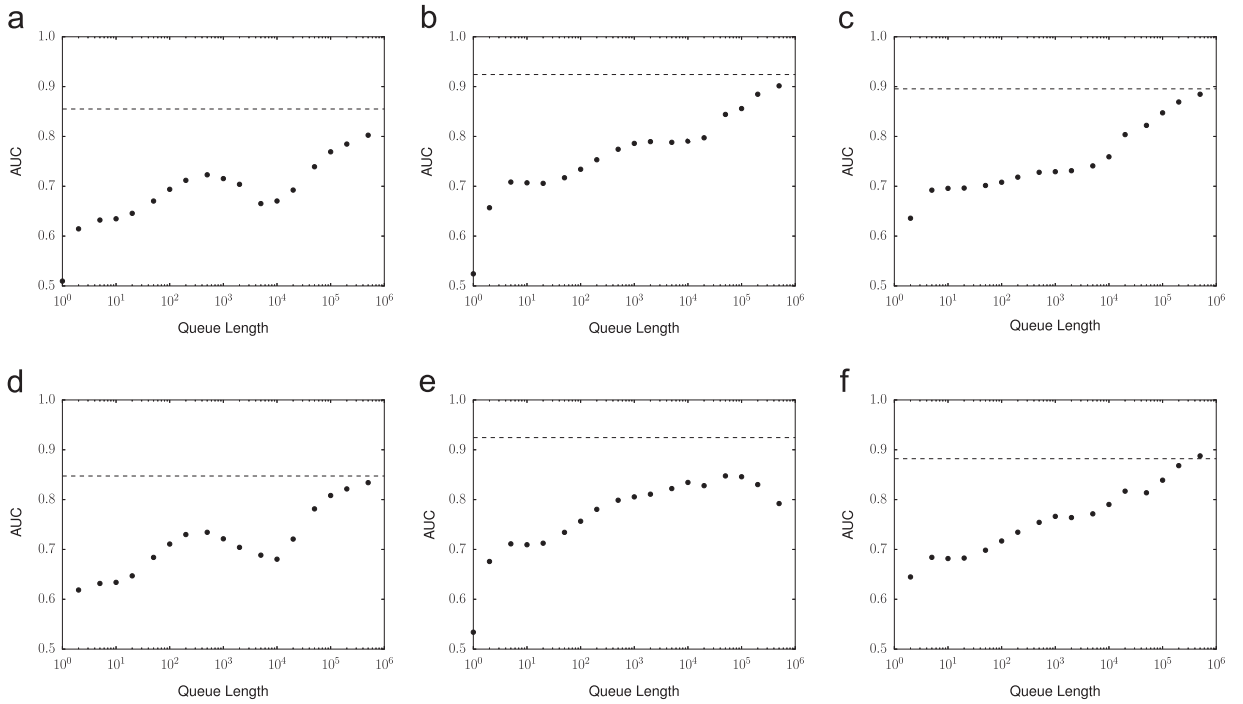

Fig. 11. AUC values for the filterless baseline method with various queue lengths. The horizontal dashed line shows the AUC values for the proposed method: (a) Route C1, Fold 1, (b) Route 5, Fold 1, (c) Route B, Fold 1, (d) Route C1, Fold 2, (e) Route 5, Fold 2, and (f) Route B, Fold 2.

Varying the queue length from 1 to 500,000, we executed the base line method with a single thread to reduce the overhead of context switching whenever the queue length was less than 10,000. Elsewhere, we used 32 threads to reduce the actual time of execution. We evaluated the two methods in terms of their AUC values and CPU times.

Although it was suggested [25] that input instances should be filtered in accordance with the heuristic conditions, we first evaluated the performance of the baseline method without any filtering to compare its discrimination performance for each input with that of the proposed method. The performance was evaluated using AUC values. Figs. 11 and 12 show the performance of the filterless baseline method with various queue lengths, in terms of the AUC values and the CPU time required to process the input dataset, with the horizontal dashed line showing the performance of the proposed method. Our method completed the detection task for a half year within 100 s for each route. As indicated in the figures, the performance of the baseline method was improved as the queue length increased because it could utilize a wider range of knowledge, but at a considerable cost in time to complete the detection. The detection performance of the proposed method was comparable to that of the baseline method when the queue length is 500,000, but the CPU time for our method was less than 0.1% of that for the baseline. Although the detection performance of the baseline method could be improved with a larger queue length to exploit more data, the algorithm would take much longer than would our method, thereby causing difficulty for real-time applications. Our method has access to data observed in the past in a compact form and can detect incidents in a short time.

Next, we evaluated the performance of the baseline method using the proposed filtering [25], to compare the detection performance for each incident with that of the proposed method. The filter discards any input vectors considered not to be from an incident, based on heuristics. Although the ROC curve should connect points (0,0) and (1,1), the baseline method with filtering broke off before the (1,1) point was reached, because the method filtered out some feature vectors of incidents, with the number of tested trajectories being less than the total number of trajectories. From the perspective of incident detection, dropping input vectors that are actually from an incident is permissible provided that at least one input vector is detected for the incident. We therefore evaluated the performance using the detection rate (DR), the ratio of the number of detected incidents to that of actual incidents. Because several trajectories may be involved in one incident, we judged that an incident was correctly detected if at least one trajectory or feature vector was detected by each method.

As with the ROC curve, the FAR–DR curve is drawn by plotting the DR against the FPR at various threshold. We used the AUC value of the FAR–DR curve instead of the ROC curve, which we call the DR-AUC value in this paper. Because the FAR–DR curve of the baseline method also broke off before (1,1) was reached, the DR-AUC value was calculated by linear interpolation between the right-hand end of the FAR–DR curve and (1,1). Fig. 13 shows the DR-AUC values for the filtered baseline method with various queue lengths, and Fig. 14 shows the CPU times required to process the entire input dataset. From these figures, the detection performance of the proposed method was comparable to that of the baseline method.

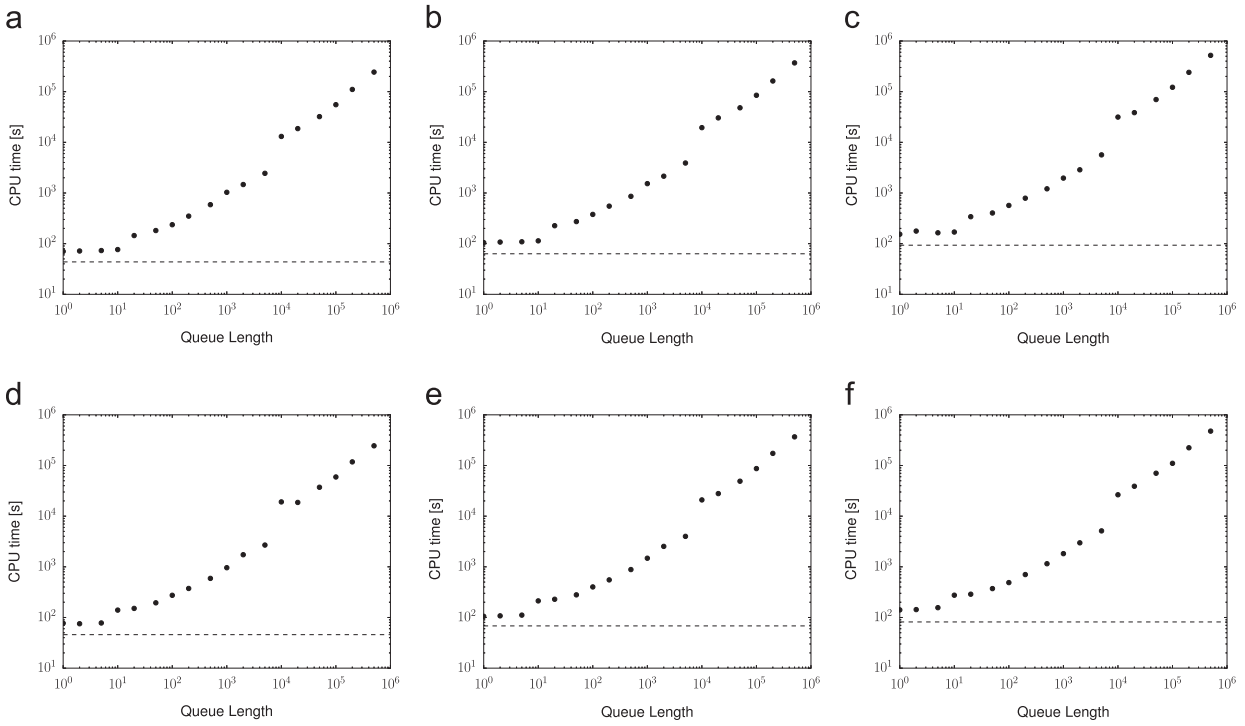

Fig. 12. CPU times for the filterless baseline method with various queue lengths. The horizontal dashed line shows the CPU time for the proposed method: (a) Route C1, Fold 1, (b) Route 5, Fold 1, (c) Route B, Fold 1, (d) Route C1, Fold 2, (e) Route 5, Fold 2, and (f) Route B, Fold 2.

5. Discussion

Previous work has exploited several heuristics to detect traffic incidents and congestion, whereas our work takes a completely statistical approach that avoids heuristics and enables real-time applications. In this study, we introduced a traffic state model based on a probabilistic topic model, proposed an incident detection method using the model, and tested its discrimination performance and incident detection performance.

We found that the proposed method could detect incidents at the same level as the existing method in a shorter time. Because our method is generic and extensible, it would be expected to outperform existing methods by including several heuristics that consider the context of individual situations. For example, it would be effective to optimize the detection threshold according to the section and time. Whereas the experiment in this paper took only the route into account, the granularity of threshold tuning remains to be determined in future studies. Although we used only the vehicular speed as the observed value in this experiment, other features, which can be extracted carefully from PCD via heuristics, should also contribute to improving the detection performance. In addition, it might be effective to extend the traffic state model by introducing other latent variables and relations among them. It is also possible to apply our method to data obtained by roadside sensors, given that a road segment in our model can correspond to a stationary sensor. Again, this might improve the detection performance. In that case, we could employ the detection algorithm for the SB approach instead of using the TB version. Although we evaluated the detection performance using ROC curves, determining the value of the threshold is necessary for practical use. It is known that the detection threshold can be optimized using ROC curves [48]. The adaptive optimization of detection thresholds was not considered here, being left to future work.

From the perspective of continuous traffic monitoring, any incident should be followed up from its occurrence to its resolution. Although it is sufficient for the initial detection of an incident that at least one input instance can be detected reliably, even if other instances of the incident are not detected, this would be insufficient for these applications. To realize an automated monitoring system, it must be able to determine correctly whether a current observation of traffic or a trajectory is anomalous.

We found that the proposed method could distinguish trajectories that were involved in an incident better than the existing method. The Shuto Expressway system has many bottlenecks, such as curves and narrow sections that require frequent changes in vehicular speed, unlike straight freeways. We speculate that this is the reason that our intuitive method found that “unusual” car behavior worked better than a heuristic method that pays attention to changes in speed. As our method detected nonabnormal segments near places of incidents as well as abnormal segments in the experiment, future work should elaborate the algorithm so that the detector can have high resolution of discrimination. Future work should also verify that this approach is also effective for

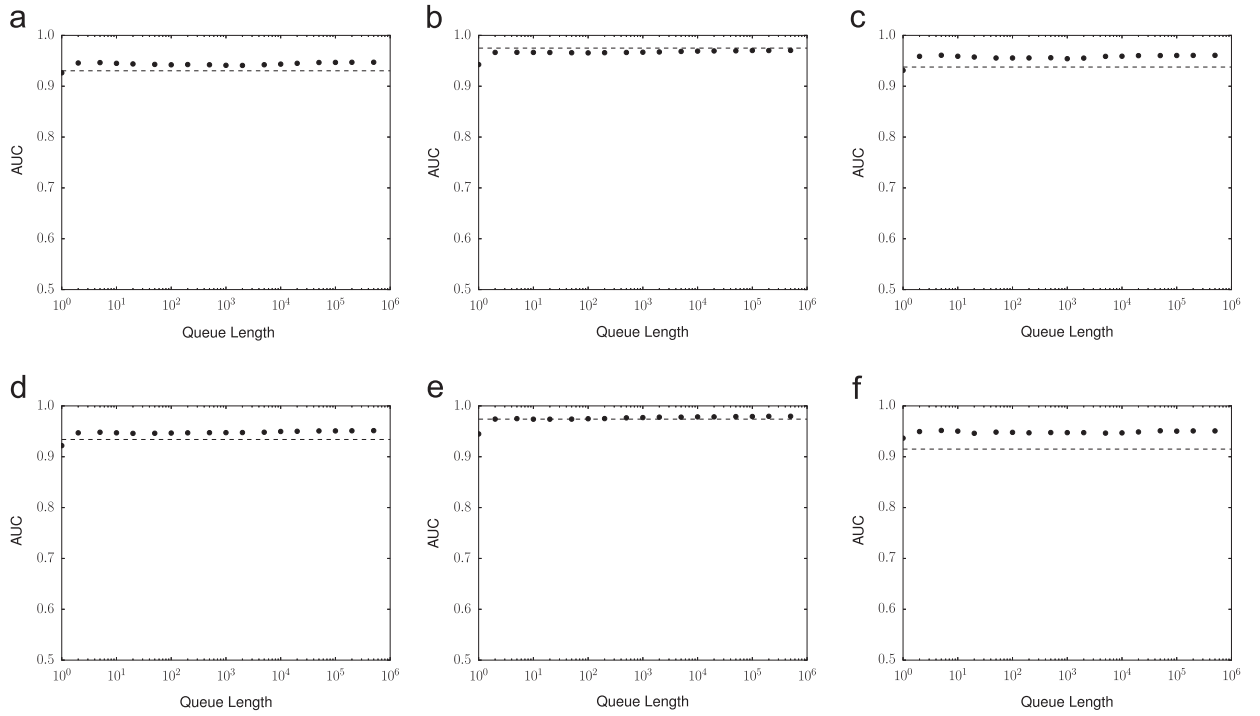

Fig. 13. DR-AUC values for the filtered baseline method with various queue lengths. The horizontal dashed line shows the DR-AUC value for the proposed method: (a) Route C1, Fold 1, (b) Route 5, Fold 1, (c) Route B, Fold 1, (d) Route C1, Fold 2, (e) Route 5, Fold 2, and (f) Route B, Fold 2.

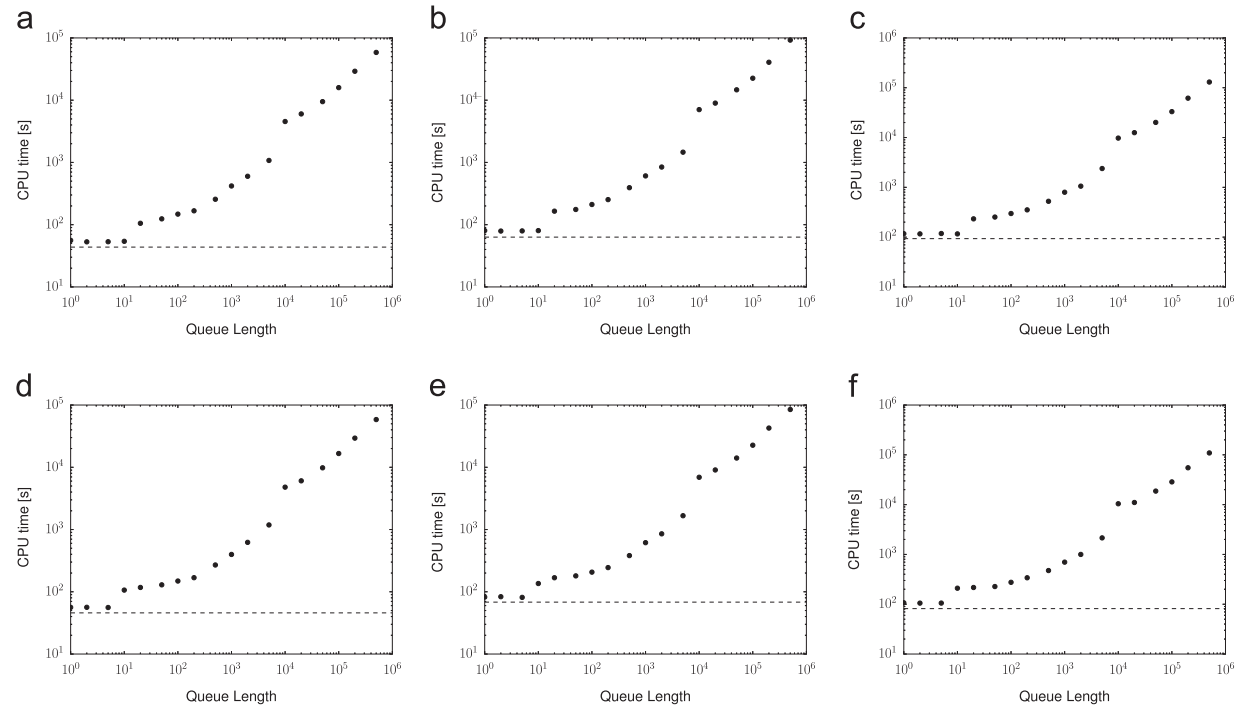

Fig. 14. CPU times for the filtered baseline method with various queue lengths. The horizontal dashed line shows the CPU time for the proposed method: (a) Route C1, Fold 1, (b) Route 5, Fold 1, (c) Route B, Fold 1, (d) Route C1, Fold 2, (e) Route 5, Fold 2 and (f) Route B, Fold 2.

local streets where normal behavior involves cars stopping frequently.

We tested the performance of the proposed method using several of the divergence functions proposed in Section 3.3.1 and found that using the negative log probability of the observation as the divergence function was the worst performer. This observation indicates that for incident detection problems, it is not satisfactory simply to

find low-probability objects. Conversely, the experiment showed that KL divergence was the best performer for the divergence function. In our model, a traffic state is associated with a Poisson distribution, which generates an observation value for vehicular speed. As shown in the Appendix to this paper, given two Poisson distributions p_1 and p_2 whose parameters are λ_1 and λ_2 respectively, $\mathrm{KL}(p_1 \| p_2) > \mathrm{KL}(p_2 \| p_1)$ holds where $\lambda_1 > \lambda_2 > 0$. Because the parameter of the Poisson distribution is equivalent to its mean, this property can be interpreted in the context of our traffic state model as follows. Assume a fast traffic state F whose mean speed is λ_F and a slow traffic state S whose mean speed is λ_S, where $\lambda_F > \lambda_S$. According to the property of KL divergence, the divergence of the current traffic state S from the usual traffic state F is greater than that of the current traffic state F from the usual traffic state S. This property is advantageous for the incident detection problem because traffic incidents often cause traffic congestion and a slowdown. However, a particular incident will be hard to detect by the proposed method if the traffic behavior in the incident is very similar to regular spontaneous congestion. It remains a challenge for future research to detect these kinds of incidents and to extract a deeper insight about each detected incident, which might involve an accident, car troubles, or road debris.

So far, we have discussed the discrimination and detection performance of the proposed method. We now consider another topic of key interest, namely, real-time processing. Our detection method itself is scalable because its complexity is $O(K)$ for each input, and the number of traffic states K is not very large in practice, as shown in the experiment. We have designed and developed a real-time traffic incident detection system, which applies our algorithm to a PCD stream and shows the high-divergence segments on a map [9]. However, our detection method is based on the traffic state model, which was acquired via batch processing. In the long term, the usual pattern of traffic can vary, and it will be necessary to update the model by some means. For our experiment, the learning process for about 18 months of data was completed in about 18 min using parallel-processing technology, which suggests that regular updating might be a simple solution to this problem. Although our experiment showed that the number of traffic states was not very large in practice, the number of segments and observations will surely become greater in the future. This requires that the algorithm should be scalable in this respect. Further work is in progress to develop a scalable method for learning the model.

6. Conclusion

In this paper, we have studied the problem of detecting traffic incidents using PCD. Although congestion can be detected by monitoring vehicular speeds, it is a chronic condition in some spots and does not necessarily indicate the occurrence of an incident. To detect traffic incidents, we took an approach that identifies any unusual events. We first introduced a probabilistic topic model to describe the state of monitored traffic so that usual traffic behavior can be learned. We then proposed several divergence functions for evaluating the difference between the current and usual traffic based on the model and streaming algorithms to detect high-divergence segments in real time. Our method was applied to real PCD collected for the entire Shuto Expressway system in Tokyo, and the discrimination and detection performance was evaluated. The results showed that our method could discriminate trajectories affected by incidents from other trajectories, using KL divergence as the divergence function, which enables monitoring of an incident from beginning to end.

Acknowledgment

This work was supported by the CPS-IIP Project in the research promotion program for national-level challenges "Research and development for the realization of next-generation IT platforms" of the Ministry of Education, Culture, Sports, Science and Technology, Japan. The National Land Numerical Information was provided by the Ministry of Land, Infrastructure, Transport and Tourism, Japan. The traffic log used in our experiment as the ground truth for incident occurrence was made available by Metropolitan Expressway Co., Ltd.

Appendix A. KL divergence between two Poisson distributions

The Poisson distribution is defined as follows:

$$p(x|\lambda) = \frac{\lambda^x e^{-\lambda}}{x!}, \tag{A.1}$$

where x is a non-negative integer and $\lambda > 0$. The mean and variance of the Poisson distribution $p(x|\lambda)$ are both equivalent to λ. Let $p_1(x) \equiv p(x|\lambda_1)$ and $p_2(x) \equiv p(x|\lambda_2)$ both be Poisson distributions. The KL divergence between them $\mathrm{KL}(p_1 \| p_2)$ is derived as follows:

$$\mathrm{KL}(p_1 \| p_2) = \lambda_2 - \lambda_1 + \lambda_1 \log \frac{\lambda_1}{\lambda_2}. \tag{A.2}$$

According to this definition, the KL divergence is clearly asymmetric, i.e., $\mathrm{KL}(p_1 \| p_2) \neq \mathrm{KL}(p_2 \| p_1)$ where $\lambda_1 \neq \lambda_2$.

Assume $\lambda_1 > \lambda_2 > 0$. We obtain the two KL divergences as follows:

$$\mathrm{KL}(p_1 \| p_2) = \lambda_2 - \lambda_1 + \lambda_1 \log \lambda_1 - \lambda_1 \log \lambda_2, \tag{A.3}$$

$$\mathrm{KL}(p_2 \| p_1) = \lambda_1 - \lambda_2 + \lambda_2 \log \lambda_2 - \lambda_2 \log \lambda_1. \tag{A.4}$$

Let $\Delta(\lambda_1, \lambda_2)$ be the difference between them:

$$\Delta(\lambda_1, \lambda_2) = \mathrm{KL}(p_1 \| p_2) - \mathrm{KL}(p_2 \| p_1) \tag{A.5}$$

$$\begin{aligned}\Delta(\lambda_1, \lambda_2) = {} & 2\lambda_2 - 2\lambda_1 + \lambda_1 \log \lambda_1 \\ & - \lambda_1 \log \lambda_2 - \lambda_2 \log \lambda_2 + \lambda_2 \log \lambda_1.\end{aligned} \tag{A.6}$$

We will now show that $\Delta(\lambda_1, \lambda_2) > 0$. First, the partial derivative of Δ with respect to λ_1 is as follows:

$$\begin{aligned}\frac{\partial \Delta}{\partial \lambda_1} &= -2 + \log \lambda_1 + 1 - \log \lambda_2 + \frac{\lambda_2}{\lambda_1} \\ &= \frac{\lambda_2}{\lambda_1} - \log \frac{\lambda_2}{\lambda_1} - 1.\end{aligned} \tag{A.7}$$

By substituting μ for λ_2/λ_1, we obtain the following function:

$$\delta(\mu) = \mu - \log \mu - 1. \tag{A.8}$$

The derivative of Eq. (A.8) is as follows:

$$\delta'(\mu) = 1 - \frac{1}{\mu} = \frac{\mu - 1}{\mu}. \tag{A.9}$$

μtakes a value between 0 and 1 because $\lambda_1 > \lambda_2 > 0$. Within this range of μ, $\delta'(\mu) < 0$ holds. Accordingly, $\delta(\mu)$ decreases monotonically. Therefore, $\delta(\mu) > \delta(1) = 0$ holds, where $0 < \mu < 1$, and the right side of Eq. (A.7) becomes positive, giving

$$\frac{\partial \Delta}{\partial \lambda_1} > 0. \tag{A.10}$$

Provided that λ_2 is constant and λ_1 is greater than λ_2, Δ will decrease as λ_1 decreases:

$$\Delta(\lambda_1, \lambda_2) > \Delta(\lambda_2, \lambda_2) = 0. \tag{A.11}$$

Therefore, $\Delta(\lambda_1, \lambda_2) > 0$ and we finally obtain the following inequality:

$$\mathrm{KL}(p_1 \| p_2) > \mathrm{KL}(p_2 \| p_1), \tag{A.12}$$

where $\lambda_1 > \lambda_2 > 0$.

References

[1] J.M. Sussman, ITS: a short history and a perspective on the future, in: Perspectives on Intelligent Transportation Systems (ITS), no. December 1996, Springer, US, 2005, pp. 3–17. http://dx.doi.org/10.1007/b101063.

[2] H. Akatsuka, A. Takasu, K. Aihara, J. Adachi, Highway incident detection based on probe car data, in: Proceedings of the International Conference on Information Systems 2013, Lisbon, Portugal, 2013, pp. 103–110.

[3] J. Wang, X. Li, S.S. Liao, Z. Hua, A hybrid approach for automatic incident detection, IEEE Trans. Intell. Transp. Syst. 14 (3) (2013) 1176–1185, http://dx.doi.org/10.1109/TITS.2013.2255594.

[4] J. Yoon, B. Noble, M. Liu, Surface street traffic estimation, in: Proceedings of the 5th International Conference on Mobile Systems, Applications and Services (MobiSys '07), ACM, San Juan, Puerto Rico, 2007, pp. 220–232. http://dx.doi.org/10.1145/1247660.1247686.

[5] East Nippon Expressway Co., Ltd., Generation mechanism of traffic congestion caused by traffic concentration, URL: ⟨http://www.e-nexco.co.jp/activity/safety/mechanism.html⟩, accessed: 2013-12-06.

[6] M. Treiber, A. Kesting, Traffic Flow Dynamics, Springer, Berlin, Heidelberghttp://dx.doi.org/10.1007/978-3-642-32460-4.

[7] D.M. Blei, Probabilistic topic models, Commun. ACM 55 (4) (2012) 77–84, http://dx.doi.org/10.1145/2133806.2133826.

[8] Metropolitan Expressway Co., Ltd., Corporate information, URL: ⟨http://www.shutoko.co.jp/company/database/⟩, accessed: 2014-11-27.

[9] A. Kinoshita, A. Takasu, J. Adachi, Real-time traffic incident detection using probe-car data on the Tokyo Metropolitan Expressway, in: Proceedings of the 2014 IEEE International Conference on Big Data, IEEE, Washington, DC, USA, 2014, pp. 43–45, http://dx.doi.org/10.1109/BigData.2014.7004488.

[10] A. Kinoshita, A. Takasu, J. Adachi, Traffic incident detection using probabilistic topic model, in: Proceedings of the Workshops of the EDBT/ICDT 2014 Joint Conference, Athens, Greece, 2014, pp. 323–330.

[11] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: a survey, ACM Comput. Surv. 41 (3) (2009) 15:1–15:58, http://dx.doi.org/10.1145/1541880.1541882.

[12] F. Iglesias, T. Zseby, Analysis of network traffic features for anomaly detection, Mach. Learn, 2014, http://dx.doi.org/10.1007/s10994-014-5473-9.

[13] C. O'Reilly, A. Gluhak, M.A. Imran, S. Rajasegarar, Anomaly detection in wireless sensor networks in a non-stationary environment, IEEE Commun. Surv. Tutor. 16 (3) (2014) 1413–1432, http://dx.doi.org/10.1109/SURV.2013.112813.00168.

[14] D. Savage, X. Zhang, X. Yu, P. Chou, Q. Wang, Anomaly detection in online social networks, Soc. Netw. 39 (2014) 62–70, http://dx.doi.org/10.1016/j.socnet.2014.05.002.

[15] Y. Zheng, S. Rajasegarar, C. Leckie, M. Palaniswami, Smart car parking: temporal clustering and anomaly detection in urban car parking, in: 2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), IEEE, Singapore, 2014, pp. 1–6. http://dx.doi.org/10.1109/ISSNIP.2014.6827618.

[16] J. La-inchua, S. Chivapreecha, S. Thajchayapong, A new system for traffic incident detection using fuzzy logic and majority voting, in: Proceedings of the 10th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON '13), IEEE, Krabi, Thailand, 2013, pp. 1–5. http://dx.doi.org/10.1109/ECTICon.2013.6559596.

[17] J. Guo, W. Huang, B.M. Williams, Real time traffic flow outlier detection using short-term traffic conditional variance prediction, Transp. Res. Part C: Emerg. Technol. 50 (2015) 160–172, http://dx.doi.org/10.1016/j.trc.2014.07.005.

[18] K. Hi-ri-o tappa, C. Likitkhajorn, A. Poolsawat, S. Thajchayapong, Traffic incident detection system using series of point detectors, in: Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems (ITCS '12), IEEE, Anchorage, Alaska, USA, 2012, pp. 182–187. http://dx.doi.org/10.1109/ITSC.2012.6338692.

[19] B. Abdulhai, S.G. Ritchie, Enhancing the universality and transferability of freeway incident detection using a Bayesian-based neural network, Transp. Res. Part C: Emerg. Technol. 7 (5) (1999) 261–280, http://dx.doi.org/10.1016/S0968-090X(99)00022-4.

[20] F. Yuan, R.L. Cheu, Incident detection using support vector machines, Transp. Res. Part C: Emerg. Technol. 11 (3–4) (2003) 309–328, http://dx.doi.org/10.1016/S0968-090X(03)00020-2.

[21] C. Piciarelli, C. Micheloni, G.L. Foresti, Trajectory-based anomalous event detection, IEEE Trans. Circuits Syst. Video Technol. 18 (11) (2008) 1544–1554, http://dx.doi.org/10.1109/TCSVT.2008.2005599.

[22] C. Xie, L. Shang, Anomaly detection in crowded scenes using genetic programming, in: 2014 IEEE Congress on Evolutionary Computation (CEC), IEEE, Beijing, China, 2014, pp. 1832–1839. http://dx.doi.org/10.1109/CEC.2014.6900396.

[23] H. Mousavi, S. Mohammadi, A. Perina, R. Chellali, V. Murino, Analyzing tracklets for the detection of abnormal crowd behavior, in: 2015 IEEE Winter Conference on Applications of Computer Vision, IEEE, Waikoloa, HI, 2015, pp. 148–155. http://dx.doi.org/10.1109/WACV.2015.27.

[24] J.-G. Lee, J. Han, X. Li, Trajectory outlier detection: a partition-and-detect framework, in: Proceedings of the 24th IEEE International Conference on Data Engineering (ICDE '08), IEEE, Cancun, 2008, pp. 140–149. http://dx.doi.org/10.1109/ICDE.2008.4497422.

[25] T. Zhu, J. Wang, W. Lv, Outlier mining based automatic incident detection on urban arterial road, in: Proceedings of the 6th International Conference on Mobile Technology, Application & Systems (Mobility '09), ACM, Nice, France, 2009, pp. 29:1–29:6. http://dx.doi.org/10.1145/1710035.1710064.

[26] S. Tao, V. Manolopoulos, S. Rodriguez, A. Rusu, Real-time urban traffic state estimation with A-GPS mobile phones as probes, J. Transp. Technol. 2 (1) (2012) 22–31, http://dx.doi.org/10.4236/jtts.2012.21003.

[27] Y. Yuan, J.W.C. van Lint, R.E. Wilson, F. van Wageningen-Kessels, S.P. Hoogendoorn, Real-time Lagrangian traffic state estimator for freeways, IEEE Trans. Intell. Transp. Syst. 13 (1) (2012) 59–70, http://dx.doi.org/10.1109/TITS.2011.2178837.

[28] C. de Fabritiis, R. Ragona, G. Valenti, Traffic estimation and prediction based on real time floating car data, in: Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems, IEEE, Beijing, China, 2008, pp. 197–203. http://dx.doi.org/10.1109/ITSC.2008.4732534.

[29] B.S. Kerner, C. Demir, R.G. Herrtwich, S.L. Klenov, H. Rehborn, M. Aleksić, A. Haug, Traffic state detection with floating car data in road networks, in: Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems, IEEE, Vienna, Austria, 2005, pp. 44–49. http://dx.doi.org/10.1109/ITSC.2005.1520133.

[30] J. Xia, W. Huang, J. Guo, A clustering approach to online freeway traffic state identification using ITS data, KSCE J. Civil Eng. 16 (3) (2012) 426–432, http://dx.doi.org/10.1007/s12205-012-1233-1.

[31] Z. Liao, Y. Yu, B. Chen, Anomaly detection in GPS data based on visual analytics, in: IEEE Symposium on Visual Analytics Science and Technology (VAST), IEEE, Salt Lake City, UT, 2010, pp. 51–58. http://dx.doi.org/10.1109/VAST.2010.5652467.

[32] Y. Qi, S. Ishak, A Hidden Markov Model for short term prediction of traffic conditions on freeways, Transp. Res. Part C: Emerg. Technol. 43 (2014) 95–111, http://dx.doi.org/10.1016/j.trc.2014.02.007.

[33] J. Kwon, K. Murphy, Modeling Freeway Traffic with Coupled HMMs, Technical Report, University of California, Berkeley, 2000.

[34] R. Herring, A. Hofleitner, P. Abbeel, A. Bayen, Estimating arterial traffic conditions using sparse probe data, in: Proceedings of the 13th International IEEE Conference on Intelligent Transportation

Systems (ITSC), IEEE, Funchal, 2010, pp. 929–936. http://dx.doi.org/10.1109/ITSC.2010.5624994.
[35] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022. URL: ⟨http://dl.acm.org/citation.cfm?id=944919.944937⟩.
[36] J. Yuan, Y. Zheng, X. Xie, Discovering regions of different functions in a city using human mobility and POIs, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '12), ACM, Beijing, China, 2012, pp. 186–194. http://dx.doi.org/10.1145/2339530.2339561.
[37] K. Farrahi, D. Gatica-Perez, Discovering routines from large-scale human locations using probabilistic topic models, ACM Trans. Intell. Syst. Technol. 2 (1) (2011) 3:1–3:27, http://dx.doi.org/10.1145/1889681.1889684.
[38] T. Hospedales, S. Gong, T. Xiang, A Markov clustering topic model for mining behaviour in video, in: IEEE 12th International Conference on Computer Vision, IEEE, Kyoto, 2009, pp. 1165–1172. http://dx.doi.org/10.1109/ICCV.2009.5459342.
[39] Z. Zhao, W. Xu, D. Cheng, EM-LDA model of user behavior detection for energy efficiency, in: IEEE International Conference on System Science and Engineering, IEEE, Shanghai, 2014, pp. 295–300. http://dx.doi.org/10.1109/ICSSE.2014.6887952.
[40] R. Yu, X. He, Y. Liu, GLAD: group anomaly detection in social media analysis, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '14), ACM, New York, New York, USA, 2014, pp. 372–381. http://dx.doi.org/10.1145/2623330.2623719.
[41] X. Hu, S. Hu, X. Zhang, H. Zhang, L. Luo, Anomaly detection based on local nearest neighbor distance descriptor in crowded scenes, Sci. World J. 2014 (2014) 632575. http://dx.doi.org/10.1155/2014/632575. URL: ⟨http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4106157/⟩.
[42] D. Pathak, A. Sharang, A. Mukerjee, Anomaly localization in topic-based analysis of surveillance videos, in: 2015 IEEE Winter Conference on Applications of Computer Vision, IEEE, Waikoloa, HI, 2015, pp. 389–395. http://dx.doi.org/10.1109/WACV.2015.58.
[43] H. Jeong, Y. Yoo, K.M. Yi, J.Y. Choi, Two-stage online inference model for traffic pattern analysis and anomaly detection, Mach. Vis. Appl. 25 (6) (2014) 1501–1517, http://dx.doi.org/10.1007/s00138-014-0629-y.
[44] C.M. Bishop, Mixture models and EM, in: Pattern Recognition and Machine Learning, Springer, New York, NY, USA, 2006, pp. 423–460 (Chapter 9).
[45] M.A. Quddus, W.Y. Ochieng, R.B. Noland, Current map-matching algorithms for transport applications: state-of-the art and future research directions, Transp. Res. Part C: Emerg. Technol. 15 (5) (2007) 312–328, http://dx.doi.org/10.1016/j.trc.2007.05.002.
[46] C.E. White, D. Bernstein, A.L. Kornhauser, Some map matching algorithms for personal navigation assistants, Transp. Res. Part C: Emerg. Technol. 8 (1–6) (2000) 91–108, http://dx.doi.org/10.1016/S0968-090X(00)00026-7.
[47] J.-s. Yang, S.-p. Kang, K.-s. Chon, The map matching algorithm of GPS data with relatively long polling time intervals, J. Eastern Asia Soc. Transp. Stud. 6 (2005) 2561–2573, http://dx.doi.org/10.11175/easts.6.2561.
[48] A.K. Akobeng, Understanding diagnostic tests 3: receiver operating characteristic curves, Acta Paediatr. 96 (5) (2007) 644–647, http://dx.doi.org/10.1111/j.1651-2227.2006.00178.x.